PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* DAN *RANDOM FOREST* DALAM MENDETEKSI PENYAKIT DIABETES

SKRIPSI

Oleh:

Habib Alrasyid

2019503008

PROGRAM STUDI TEKNOLOGI INFORMASI

FAKULTAS SAINS & TEKNOLOGI UNIVERSITAS IBRAHIMY

SITUBONDO

2024

PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* DAN *RANDOM FOREST* DALAM MENDETEKSI PENYAKIT DIABETES

SKRIPSI

Diajukan untuk memenuhi salah satu persyaratan dalam menyelesaikan Program Sarjana (S-1) pada Program Studi Teknologi Informasi Fakultas Sains dan Teknologi Universitas Ibrahimy Situbondo

Oleh:

HABIB ALRASYID

2019503008

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS IBRAHIMY
SITUBONDO
2024

PERNYATAAN KEASLIAN TULISAN

Yang bertanda tangan dibawah ini:

Nama : Habib Alrasyid

NPM : 2019503008

Program Studi : Teknologi Informasi

Fakultas : Fakultas Sains dan Teknologi

Menyatakan dengan sebenarnya, bahwa tugas akhir/skripsi ini secara keseluruhan adalah hasil penelitian atau karya saya sendiri, kecuali pada bagian-bagian yang dirujuk sebagai sumber referensi dan disebutkan dalam daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa tugas akhir/skripsi ini hasil plagiasi, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Situbondo, 21 Agustus 2024

Habib Alrasyid

PERSETUJUAN PEMBIMBING

Skripsi ini ditulis oleh:

Nama : Habib Alrasyid
NPM : 2019503002
Judul : PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* DAN *RANDOM FOREST* DALAM MENDETEKSI PENYAKIT DIABETES

Telah ditelaah dan disetujui oleh

Pembimbing I,	Pembimbing II,
Ahmad Homaidi, M. Kom	**Zaehol Fatah, M. Kom**
NIDN: 0705078901	NIDN: 0715057801

HALAMAN PENGESAHAN

SKRIPSI

PERBANDINGAN ALGORITMA *SUPPORT VECTOR MACHINE* DAN *RANDOM FOREST* DALAM MENDETEKSI PENYAKIT DIABETES

HABIB ALRASYID
2019503002

Telah dipertahankan didepan dewan penguji Sidang/Munaqosah Skripsi pada hari Rabu

Tanggal 21 Agustus 2024 sebagai salah satu syarat memperoleh gelar Sarjana (S.Kom).

Pada Fakultas Sains & Teknologi Uneversitas Ibrahimy

Tim Penguji,

Ketua Sidang,

Abdul Wafi, S.Pi, M.P.
NIDN: 0705049103

Sekretaris Sidang,

Abdus Samad, M.Kom
NIDN: 0709099006

Penguji I,

Akhlis Munazilin, S.Kom., M.T.
NIDN: 0712098601

Penguji II,

Lukman Fakih Lidimilah, M.Kom
NIDN: 0715099001

Mengetahui
Dekan,

Abd. Ghofur, M. Kom.
NIDN: 0711088303

MOTTO

"JANGAN PERNAH HIDUP SEPERTI AIR, MENGALIR BUKAN PADA KEINGINANNYA"

KATA PENGANTAR

Segala puji syukur peneliti sampaikan kepada Allah SWT, karena atas Rahmat dan Hidayah-Nya, perencanaan, pelaksanaan dan penyelesaian tugas akhir/skripsi dengan judul "Perbandingan Algoritma *Support vector machine* dan *Random Forest* Dalam Mendeteksi Penyakit Diabetes" sebagai salah satu syarat penyelesaian program diploma/sarjana dapat terselesaikan dengan baik dan lancar, oleh karena itu kami mengucapkan terima kasih kepada:

1. KHR. Ahmad Azaim Ibrahimy selaku Pengasuh Pondok Pesantren Salafiyah Syafi'iyah Sukorejo.
2. Bapak KH. Ahmad Fadloil, S.H., M.H. selaku Rektor Universitas Ibrahimy Situbondo.
3. Bapak Abd. Ghofur, M. Kom., selaku Dekan Fakultas Sains dan Teknologi Universitas Ibrahimy
4. Bapak Dr. Ach. Khumaidi, M.P., selaku Wakil Dekan I Fakultas Sains dan Teknologi Universitas Ibrahimy.
5. Bapak Abd. Wafi, M.P., selaku Wakil Dekan II Fakultas Sains dan Teknologi Universitas Ibrahimy.
6. Bapak Ahmad Lutfi, M. Kom., selaku Wakil Dekan III Fakultas Sains dan Teknologi Universitas Ibrahimy.
7. Bapak Firman Santoso, M. Kom., selaku Ketua Program Studi Teknologi Informasi.
8. Bapak Ahmad Homaidi, M. Kom., selaku Dosen Pembimbing I yang telah membimbing saya untuk penyelesaian skripsi ini.
9. Bapak Zaehol Fatah, M. Kom., selaku Dosen Pembimbing II yang telah membimbing saya untuk penyelesaian skripsi ini.
10. Seluruh Dosen Fakultas Sains dan Teknologi Universitas Ibrahimy yang telah memberikan ilmu sehingga saya dapat menyelesaikan Tugas Akhir pada tahun ini.

11. Seluruh Civitas Akademika Fakultas Sains dan Teknologi Universitas Ibrahimy Situbondo.

Situbondo, 21 Agustus 2024
Peneliti

Habib Alrasyid

PERSEMBAHAN

Saya persembahkan laporan ini kepada orang-orang yang telah membantu saya dalam menyelesaikan tugas akhir skripsi saya ini:

1. Kedua Orang tua yang telah berjuang demi masa depan saya yang lebih baik.
2. Teman-teman yang paling saya cintai dan banggakan.
3. Sahabat-sahabat saya yang telah memberikan dukungan dalam suka maupun duka.
4. Semua orang yang tidak bisa saya sebut satu persatu yang telah mendukung saya sepenuhnya sehingga bisa menyelesaikan laporan ini dengan baik.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Habib Alrasyid, 2024. **Perbandingan Algoritma *Support vector machine* dan *Random Forest* dalam Mendeteksi Penyakit Diabetes** Skripsi Prodi Teknologi Informasi Fakultas Sains & Teknologi Universitas Ibrahimy, Pembimbing: (1. Ahmad Homaidi, M. Kom.) (2. Zaehol Fatah, M. Kom.)

Salah satu penyakit yang sangat diperhatikan dan banyak kasus yang terjadi diseluruh dunia karena dampaknya yang sangat siginfikan yaitu diabetes. penederita diabetes mengalami gangguan pada metabolisme yang mengidentifikasi hiperglikemia yang diakibatkan oleh tidak mampunya pancreas untuk mensekresi insulin yang berdampak menyebabkan kematian karena tidak berfungsinya organ tubuh yang lain. Data menyatakan pada tahun 2019 bahwa 433 juta orang terdiagnosa diabetes dan jumlahnya diprediksi meningkat hingga puncaknya pada tahun 2045 menjadi 700 juta orang. Hal ini perlu diantisipasi secepat mungkin oleh masyarakat dengan beberapa ciri yang terjadi pada penderita. Data penderita diabetes ini dapat diolah dengan *data mining* yang memanfaatkan machine learning untuk mendeteksi penyakit diabetes. Penelitian ini akan membandingkan dua algoritma yaitu *Support vector machine* dan *Random Forest* untuk menemukan hasil yang akurat. Peneliti menggunakan model KDD (Knowledge Discovery in Database) dalam beberapa tahap seperti data selection, preprocessing, Transformation dan Evaluation. Dataset yang digunakan bersumber dari website kaggel.com sebanyak 768 yang terdiri 500 negatif dan 268 positif diabetes. Pada algoritma SVM dengan kernel linear mengasilkan nilai akurasi 77%, presisi 75% dan recall 51%. Sedangkan untuk algoritma *Random Forest* dengan n_estimators=100, random_state=42 menghasilkan nilai akurasi 75%, presisi 69%, recall 55% dan F1 score 61%. Dari proses dan hasil diatas menyatakan bahwa algoritma SVM lebih cocok digunakan untuk mendeteksi penyakit diabetes. Model yang telah dibuat menggunakan Bahasa pemrograman python ini akan diimplementasikan dengan stremlit agar dapat digunakan berbasis web.

Kata Kunci: Diabetes, *Random Forest*, *Support Vector Machone, Data mining*, Deteksi

BAB I

PENDAHULUAN

1.1 Latar Belakang

Diabetes menjadi penyakit yang sangat diperhatikan di seluruh dunia karena memiliki yang sangat signifikan di masyarakat. Dampak yang terjadi pada penderita diabetes adalah terjadi gangguan pada metabolisme yang mengidentifikasi hiperglikemia yang diakibatkan oleh tidak mampunya pancreas untuk mensekresi insulin (gangguan kerja insulin). Pada fase kronis hiperglikemik berakibat menimbulkan kerusakan dalam jangka panjang dan tidak dapat berfungsinya organ tumbuh yang lain seperti ginjal, mata, serta pembuluh darah Pada fase kronis hiperglikemik, kerusakan yang berkelanjutan dan ketidakmampuan berfungsinya organ tumbuh lainnya seperti ginjal, mata, dan pembuluh darah terjadi. Menurut International Diabetes Federation (IDF), pada tahun 2019 terdapat 433 juta orang yang menderita diabetes. Dengan pola hidup yang tidak sehat, total ini diperkirakan akan meningkat menjadi 578 juta pada tahun 2030 dan 700 juta pada tahun 2045. Indonesia adalah salah satu negara dengan jumlah penderita diabetes tertinggi pada tahun 2019 peringkat sepuluh[1].

Masyarakat Indonesia mengenal penyakit diabetes dengan nama kencing manis yakni meningkatnya kadar gula darah dalam tubuh, terutama setelah makan. Meningkatnya tensi darah diatas normal 120mg/dl menggambarkan bahwa terjadi hipertensi merupakan gejala dari diabetes[2]. Faktor lain yang mempengaruhi penderita diabetes diantaranya tingginya tekanan darah, kadar gula tinggi, berat badan, faktor keturunan dan beberapa faktor yang lain. Penanganan masyarakat

terkait penyakit diabetes ini masih belum diselesaikan dengan cepat, karena memerlukan waktu dan biaya untuk pergi berobat[3]. Dengan data fakta realita yang ada, oleh karenanya diperlukan tindakan awal cepat dalam mendeteksi tanda dan gejala diabetes secara dini.

Dalam rangka menangani penyakit diabetes, tentu dengan beberapa gejala yang sudah dipaparkan diatas. Dengan menggunakan prediksi diabetes yang memproses data besar dari penderita diabetes kemudia diolah dengan metode machine learning. Beberapa penelitian yang telah dilakukan deteksi penyakit diabetes menggunakan algoritma KNN (Argina) dan algoritma dan Naïve Bayes[4].

Diantara banyak algoritma yang ada SVM dan *Random forest* merupakan algoritma yang cocok dalam menyelesaikan masalah klasifikasi pada penyakit diabetes. *Support vector machine* bekerja dengan mencari pemisah dari sebuah ruang yang paling optimal pada suatu dataset dalam kelas yang berbeda[5]. Pada konsep algoritma *Random forest* dengan membuang sebagian besar data dengan mennggganti sample random[6]. Kedua algoritma ini akan memproses data untuk menghasilkan pengetahuan dalam mendeteksi penyakit diabetes dengan proses statistika, matematika dan machine learning, proses dalam menemukan ini bisa disebut dengan *data mining*[7].

Dibandingkan dengan penelitian yang lain penelitian ini membahasa dengan pendekatan yang digunakan untuk mengklasifikasi penyakit diabetes. Penelitian terdahulu umumnya hanya menggunakan satu algoritma, seperti K-Nearest Neighbor (KNN) atau Naïve Bayes, dalam mendeteksi penyakit diabetes. Namun, penelitian ini menggunakan dua alg

oritma yang berbeda, yaitu *Support vector machine* (SVM) dan *Random forest*, untuk membandingkan performa keduanya dalam mengklasifikasi penyakit diabetes. Dengan membandingkan dua algoritma ini, diharapkan dapat ditemukan model klasifikasi yang paling akurat dan efisien dalam mendeteksi penyakit diabetes secara dini. Penelitian ini juga bertujuan membandingkan algoritma SVM dan *random forest* yang terbaik untuk deteksi penyakit diabetes. Algoritma yang terbaik akan dinilai dari nilai performa Accuracy, Precision dan Recall yang kemudian akan diimplementasi menjadi versi web menggunakan *streamlit* agar dapat digunakan dengan mudah. Penelitian ini bertujuan membandingkan algoritma SVM dan *random forest* yang terbaik untuk deteksi penyakit diabetes. Algoritma yang terbaik akan dinilai dari nilai performa Accuracy Precision dan Recall kemudian akan diimplementasi menjadi versi web menggunakan streamlit agar dapat digunakan dengan mudah.

Berdasarkan latar belakang permasalahan tersebut, pada topik **"Perbandingan Algoritma *Support vector machine* dan *Random forest* dalam Mengklasifikasi Penyakit Diabetes"** sebagai tugas akhir riset untuk menyelesaikan perbandingan metode tersebut.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang dan identifikasi masalah tersebut, maka rumusan masalah dalam penelitian ini adalah:

1. Diperlukan cara yang cepat dan efisien untuk mengklasifikasi penyakit diabetes secara dini, karena diabetes merupakan penyakit yang berbahaya.

2. Belum adanya perbandingan performa antara algoritma *Support vector machine* (SVM) dan *Random forest* dalam mengklasifikasi penyakit diabetes.
3. Perlunya implementasi model klasifikasi terbaik ke dalam aplikasi web agar mudah digunakan oleh masyarakat.

1.3 Rumusan Masalah

Berdasarkan uraian latar belakang dan identifikasi masalah tersebut, maka rumusan masalah dalam penelitian ini adalah:

1. Bagaimana mengembangkan model klasifikasi penyakit diabetes yang cepat dan efisien menggunakan algoritma SVM dan *Random forest*?
2. Bagaimana perbandingan performa antara algoritma SVM dan *Random forest* dalam mengklasifikasi penyakit diabetes berdasarkan nilai akurasi, presisi, dan recall?
3. Bagaimana mengimplementasikan model klasifikasi terbaik ke dalam aplikasi web agar mudah digunakan oleh masyarakat dalam upaya deteksi dini penyakit diabetes?

1.4 Batasan Masalah

Berdasarkan perumusan diatas maka Batasan masalah dari penelitian ini adalah:

1. Dataset yang digunakan bersumber dari website kaggle.com dengan total 768 data, terdiri dari 500 data negatif diabetes dan 268 data positif diabetes.
2. Variabel yang digunakan meliputi jumlah kehamilan, kadar glukosa, tekanan darah, ketebalan kulit, kadar insulin, indeks massa tubuh, fungsi keturunan diabetes, usia, dan hasil diagnosa diabetes.

3. Penelitian ini fokus pada perbandingan performa algoritma *Support vector machine* (SVM) dan *Random forest* dalam mengklasifikasi penyakit diabetes.

1.5 Tujuan Penelitian

Berdasarkan perumusan masalah yang telah dipaparkan maka tujuan dari penelitian ini adalah:

1. Mengembangkan model klasifikasi penyakit diabetes yang cepat dan efisien menggunakan algoritma SVM dan *Random forest*.
2. Membandingkan performa algoritma SVM dan *Random forest* dalam mengklasifikasi penyakit diabetes berdasarkan nilai akurasi, presisi, dan recall.
3. Mengimplementasikan model klasifikasi terbaik ke dalam aplikasi web berbasis streamlit agar mudah digunakan oleh masyarakat dalam upaya deteksi dini penyakit diabetes.

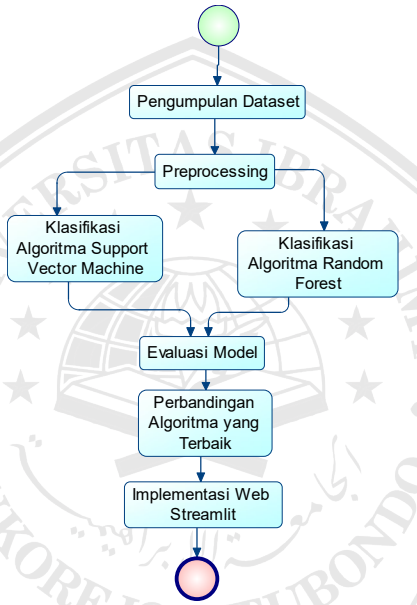
1.6 Manfaat Penelitian

Berdasarkan penjelasan diatas, maka didapatkan manfaat dari penelitian ini yaitu:

1. Memberikan alternatif solusi dalam upaya deteksi dini penyakit diabetes melalui model klasifikasi yang akurat dan efisien.
2. Memberikan kontribusi dalam pengembangan ilmu pengetahuan dan teknologi, khususnya di bidang *data mining* dan machine learning untuk kesehatan.
3. Melihat performa terbaik dari kedua algoritma dalam mengklasifikasi penyakit diabetes.

1.7 Metode Penelitian

Metodologi penelitian merupakan cara sistematis untuk mengumpulkan data dengan tujuan tertentu. Metode penelitian ini merupakan suatu upaya atau prosedur untuk menemukan solusi terhadap suatu subjek atau masalah secara hati–hati, terencana, metodis, atau ilmiah, hal tersebut dilakukan dengan tujuan untuk menemukan fakta – fakta atau langkah-langkah yang ada. metode penelitian ini berguna sebagai fungsi signifikan untuk mendapatkan data data ataupun informasi



yang dibutuhkan untuk menyelesaikan berbagai macam masalah dan memberikan suatu solusi atas masalah yang ada[18].

1.7.1 Jenis Penelitian

Penelitian kuantitatif adalah jenis penelitian yang menekankan pada pengumpulan dan analisis data numerik dengan menggunakan metode statistik untuk menguji teori, mendeskripsikan fenomena, atau menjawab pertanyaan penelitian. Tujuan utama penelitian kuantitatif adalah untuk mengukur dan

menganalisis data secara objektif melalui perhitungan statistik dan menghasilkan temuan yang dapat digeneralisasi[8].

1.7.2 Teknik Pengumpulan Data

Untuk mendapatkan hasil penelitian, maka tahapan pengumpulan data pada dijelasakan sebagai berikut

a. Pengumpulan Data

Peneliti menggunakan dataset penyakit diabetes yang bersumber dari website kaggle.com. Dataset ini merupakan data sekunder yang telah dikumpulkan dan diolah sebelumnya.

b. Studi Pustaka

Peneliti melakukan studi pustaka atau studi literatur untuk mempelajari konsep-konsep dan teori terkait diabetes, klasifikasi *data mining*, algoritma *Support vector machine* (SVM), dan algoritma *Random forest*. Hal ini dilakukan untuk membangun landasan teoretis yang kuat dalam penelitian.

1.7.3 Metode Pengembangan Sistem

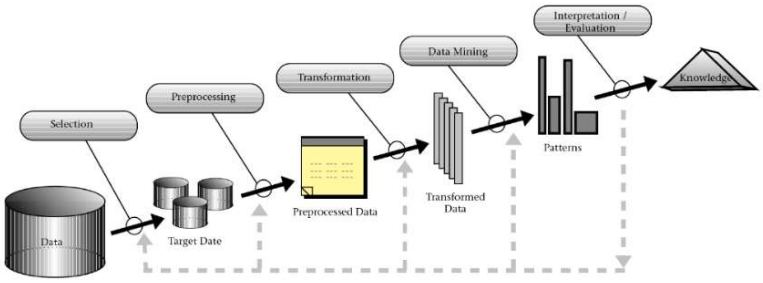
Pada penelitian kali ini penulis menggunakan metode *Support vector machine* (SVM) dalam Knowledge Discovery in Databases (KDD) dan melibatkan beberapa langkah, diantarnya pemahaman masalah dan tujuan dari penelitian, lalu data dikumpulkan dari sumber yang sesuai dan kemudian dibersihkan serta di proses, termasuk pemilihan atribut-atribut penting, kemudian menerapkan metode SVM dengan menentukan mengunakan nilai linear =”kernel”. Algoritma *Random forest* dijadikan pembanding dari *Support vector machine* dengan nilai “n_estimators=100, random_state=42” agar

mendapatkan hasil dari algoritma yang yang terbaik dari keduang, selanjutnya data dibagi menjadi set pelatihan dan pengujian untuk menguji model, lalu lakukan evaluasi model dengan menggunakan metrik seperti akurasi, precision dan recall untuk menklasifikasi, hasil dievaluasi untuk menemukan pola dan informasi baru yang didapatkan dalam perbandingan algoritma[9].

1.8 Sistematika Pembahasan

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang permasalahan terkait pentingnya deteksi dini penyakit diabetes dan potensi penggunaan machine learning dalam mengatasi permasalahan tersebut. Selain itu, bab ini juga mencakup rumusan masalah, tujuan



Gambar 1. 1 Alur Knowledge Discovery in Database

penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan tentang landasan teori yang digunakan dalam penelitian, meliputi konsep penyakit diabetes, machine learning, algoritma *Support*

vector machine (SVM), algoritma *Random forest*, dan metode evaluasi performa model.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan tentang tahapan-tahapan yang dilakukan dalam penelitian, mulai dari pengumpulan data, preprocessing data (pembersihan data, transformasi data, seleksi fitur), pembagian data (data splitting), pemodelan dengan algoritma SVM dan *Random forest*, evaluasi performa model, hingga perbandingan hasil pemodelan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil-hasil yang diperoleh dari penelitian, meliputi deskripsi data, hasil preprocessing data, hasil pembagian data, hasil pemodelan dengan algoritma SVM dan *Random forest*, serta perbandingan performa kedua model tersebut.

BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan. Kesimpulan mencakup jawaban atas rumusan masalah dan pencapaian tujuan penelitian, sedangkan saran berisi rekomendasi untuk perbaikan atau eksplorasi lebih lanjut terkait topik yang diteliti.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

"Perbandingan Metode Naïve Bayes dan *Support vector machine* Dalam Klasifikasi Penyakit Diabetes Melitus" [10]

Berdasarkan penelitian yang dilakukan oleh Hunafa dan Hermawan, perbandingan antara algoritma Naïve Bayes dan K-Nearest Neighbor (KNN) pada dataset diabetes yang tidak seimbang menunjukkan bahwa Naïve Bayes dengan teknik SMOTE memberikan performa terbaik. Meskipun KNN tanpa SMOTE memiliki akurasi tertinggi, Naïve Bayes dengan SMOTE menunjukkan keseimbangan yang lebih baik antara akurasi, presisi, dan recall. Penggunaan SMOTE meningkatkan kinerja Naïve Bayes dalam mengatasi ketidakseimbangan kelas pada dataset diabetes.

Dalam analisis perbandingan model KNN dan Naïve Bayes, ditemukan bahwa model Naïve Bayes dengan SMOTE memiliki nilai tertinggi dalam recall dan F1 Score. Meskipun akurasi dan presisinya mungkin sedikit lebih rendah dibandingkan dengan model KNN tanpa SMOTE, Naïve Bayes dengan SMOTE mampu mengenali instans positif dengan baik. Namun, terdapat kecenderungan model untuk salah mengklasifikasikan data negatif sebagai positif dan data positif sebagai negatif, menunjukkan perlunya evaluasi lebih lanjut untuk meningkatkan performa model dalam mengklasifikasikan kelas minoritas atau mayoritas secara lebih tepat.

Dari hasil confusion matrix, implementasi Naïve Bayes dengan SMOTE berhasil memprediksi sejumlah data positif dan negatif dengan tingkat presisi

sebesar 32.26% dan recall sebesar 72.72%. Meskipun terdapat data negatif yang salah diprediksi sebagai positif dan sebaliknya, model ini mampu mengidentifikasi sebagian besar data positif dengan tepat. F1 score sebesar 44.69% menunjukkan kemampuan model dalam mengkompromikan antara presisi dan recall. Penggunaan SMOTE dalam Naïve Bayes bertujuan untuk memperkuat kemampuan model dalam mengenali kelas minoritas dengan menghasilkan sampel sintetis dari kelas tersebut.

"Klasifikasi Penderita Diabetes Menggunakan Algoritma Machine Learning dan Z-Score" [11]

Penelitian ini berfokus pada implementasi teknik *data mining* untuk memprediksi penyakit diabetes dengan membandingkan performa dua algoritma klasifikasi yang populer, yaitu Naives Bayes dan K-Nearest Neighbor. Proses diawali dengan seleksi dataset diabetes yang diperoleh dari National Institute of Diabetes and Digestive and Kidney Diseases. Dataset tersebut berisi 2000 catatan pasien dengan 9 atribut prediktor seperti jumlah kehamilan, kadar glukosa, tekanan darah, ketebalan lipatan kulit, insulin, BMI, fungsi riwayat diabetes keluarga, dan usia, serta 2 kelas outcome (diabetes dan tidak diabetes).

Tahap selanjutnya adalah pra-pemrosesan data yang meliputi penanganan missing value serta normalisasi menggunakan metode simple feature scaling untuk mentransformasi rentang nilai menjadi 0 sampai 1 tanpa menghilangkan informasi. Proses ini penting untuk memastikan kualitas data sebelum digunakan pada tahap *data mining*.

Pada tahap *data mining*, dataset yang telah dipersiapkan diuji menggunakan algoritma Naives Bayes dan K-Nearest Neighbor. Evaluasi dilakukan dengan mengukur akurasi, presisi, recall, serta analisis AUC dari confusion matrix dan ROC analysis. Hasil evaluasi menunjukkan algoritma K-Nearest Neighbor unggul saat menggunakan dataset lengkap 2000 data dengan akurasi mencapai 99%, jauh melebihi akurasi Naives Bayes yang hanya 75%. Namun, saat diuji dengan 30 data sebagai data testing, Naives Bayes justru lebih akurat dengan akurasi 66% disbanding K-Nearest Neighbor yang hanya 53%.

Untuk memvalidasi hasil tersebut, dilakukan percobaan lebih lanjut menggunakan teknik 10-fold cross validation. Hasilnya mengonfirmasi bahwa K-Nearest Neighbor memang lebih akurat dengan akurasi 99% dibandingkan Naives Bayes yang hanya 75% saat digunakan untuk dataset diabetes berukuran besar. Berdasarkan serangkaian uji coba dan evaluasi, dapat disimpulkan bahwa K-Nearest Neighbor lebih tepat digunakan untuk memprediksi penyakit diabetes pada dataset berukuran besar, sementara Naives Bayes lebih sesuai untuk dataset kecil karena mampu menghasilkan akurasi prediksi yang lebih baik pada kasus tersebut.

"Analisis Perbandingan Akurasi Algortima Naïve Bayes dan C 4.5 untuk Klasifikasi Diabetes" [12]

Berikut ini adalah tinjauan pustaka yang lebih panjang dan detail berdasarkan dokumen yang diberikan:

Klasifikasi *data mining* telah banyak diaplikasikan dalam berbagai bidang, termasuk untuk diagnosis penyakit seperti diabetes. Algoritma Naïve Bayes dan C4.5 merupakan dua algoritma klasifikasi yang sering digunakan dan telah banyak

diteliti performa dan keakuratannya. Penelitian yang dilakukan oleh Ardiansyah et al. (2021) secara khusus membandingkan akurasi algoritma Naïve Bayes dan C4.5 dalam klasifikasi penyakit diabetes.

Dalam penelitian tersebut, digunakan dataset penyakit diabetes yang diperoleh dari situs Kaggle terbitan Ishan Dutta yang terdiri dari 520 data sampel dengan 17 atribut atau bidang. Dataset tersebut kemudian dibagi menjadi 7 skenario berbeda dengan jumlah atribut yang bervariasi, mulai dari 16 atribut pada skenario 1 hingga 5 atribut pada skenario 7. Proses klasifikasi dilakukan menggunakan perangkat lunak Rapidminer dengan menerapkan model algoritma Naïve Bayes dan C4.5 pada masing-masing skenario.

Hasil penelitian menunjukkan bahwa secara keseluruhan, algoritma C4.5 memiliki performa yang lebih baik dibandingkan Naïve Bayes dalam klasifikasi penyakit diabetes. Pada skenario terbaik yaitu skenario 4 dengan 10 atribut, algoritma C4.5 mencapai akurasi 99,03%, presisi 100%, dan recall 98,18%. Sementara itu, performa terbaik Naïve Bayes diperoleh pada skenario 2 dengan akurasi 88,35%, presisi 92,16%, dan recall 85,45%.

2.2 Landasan Teori

a. Algoritma *Support vector machine* (SVM)

Support vector machine (SVM) adalah teknik pembelajaran mesin yang kuat dan serbaguna yang telah mendapatkan popularitas yang luas dalam tugas klasifikasi dan prediksi. Sebagai metode pembelajaran yang diawasi, SVM dilatih menggunakan algoritma pembelajaran yang didasarkan pada prinsip-prinsip optimasi matematika dan memanfaatkan hipotesis dengan fungsi linear

dalam ruang fitur berdimensi tinggi [1]. Tujuan utama dari SVM adalah untuk menemukan hyperplane pemisah optimal yang secara efektif membagi data menjadi kelas yang berbeda sambil memaksimalkan margin, yaitu jarak antara hyperplane dan titik data terdekat dari setiap kelas [2]. Salah satu kekuatan utama SVM adalah kemampuannya untuk menangani data berdimensi tinggi dengan efisien. Dengan menggunakan teknik kernel, SVM secara implisit memetakan data input ke ruang fitur berdimensi lebih tinggi di mana kelas-kelas mungkin lebih mudah dipisahkan secara linear [4].

Sebagai sistem pembelajaran yang dilatih melalui algoritma pembelajaran yang didasarkan pada optimasi dan menggunakan hipotesos dengan fungsi linear dalam fitur yang berdimensi besar [13]. *Support vector machine* adalah metode algoritma yang melakukan klasifikasi dan prediksi dengan mencari ruang pemisah dari sisi yang optimal pada dataset dengan berbagai kelas [14]. Klasifikasi dilakukan dengan algoritma SVM yang menggunakan pelatihan data untuk model klasifikasi, dan kemudian model yang terbentuk digunakan untuk memprediksi kelas data baru yang tidak ada sebelumnya, yang dikenal sebagai pengujian data [15].

$$f(x_d) = \sum_{i=1}^{ns} a_i y_i \vec{x_i} \vec{x}_d + b \quad (1)$$

Keterangan:

ns = Jumlah support vector

αi = Nilai bobot setiap titik data

yi = Data kelas

$\vec{x} i$ = Variabel *support vector*

$\vec{x}d$ = Daya yang akan diklasifikasi

b = Nilai bias atau error

b. Algoritma *Random Forest*

Random Forest adalah algoritma ensemble learning yang sangat efektif dan serbaguna, digunakan untuk tugas-tugas klasifikasi dan regresi. Algoritma ini dibangun di atas konsep pohon keputusan, di mana setiap pohon dalam "hutan" (ensemble) mencapai node akhirnya melalui metode pemisahan biner rekursif. Dalam proses ini, setiap node dalam pohon membagi data menjadi dua subset berdasarkan fitur yang paling informatif, secara rekursif mempartisi data hingga tercapai kriteria penghentian tertentu, seperti kedalaman pohon maksimum atau jumlah sampel minimum dalam node daun [1].

Salah satu kekuatan utama dari algoritma *Random Forest* adalah kemampuannya untuk mencapai tingkat kesalahan yang rendah dan performa klasifikasi yang baik. Ini dicapai melalui proses "bootstrap aggregating" atau "bagging", di mana subset acak dari data pelatihan dipilih dengan penggantian untuk membangun setiap pohon dalam ensemble. Dengan menggunakan subset data yang berbeda untuk setiap pohon, algoritma *Random Forest* dapat mengurangi overfitting dan meningkatkan generalisasi model. Selain itu, pada setiap node dalam pohon, hanya subset acak dari variabel input yang dipertimbangkan untuk pemisahan. Pendekatan ini, yang dikenal sebagai "feature bagging", membantu

mengurangi korelasi antara pohon dan meningkatkan keragaman dalam ensemble [16].

c. Diabetes

Diabetes melitus adalah gangguan metabolisme yang ditandai dengan ketidakmampuan tubuh untuk memproduksi atau merespons hormon insulin secara efektif, mengakibatkan tingkat glukosa darah yang terus-menerus tinggi (hiperglikemia). Insulin, hormon yang diproduksi oleh pankreas, berperan penting dalam mengatur metabolisme glukosa dengan memfasilitasi penyerapan glukosa dari aliran darah ke dalam sel-sel tubuh untuk digunakan sebagai energi. Pada individu dengan diabetes melitus, gangguan dalam produksi insulin, aksi insulin, atau keduanya menyebabkan akumulasi glukosa dalam darah, mengarah pada hiperglikemia kronis [1].

Hiperglikemia berkepanjangan yang terkait dengan diabetes melitus dapat menyebabkan berbagai komplikasi kesehatan yang serius dan berpotensi mengancam jiwa. Tingginya kadar glukosa darah dapat merusak pembuluh darah kecil (mikrovaskuler) dan besar (makrovaskuler), mengarah ke komplikasi seperti retinopati diabetik, nefropati, neuropati, dan penyakit kardiovaskular. Komplikasi mikrovaskuler dapat memengaruhi mata, ginjal, dan saraf, menyebabkan gangguan penglihatan, gagal ginjal, dan hilangnya sensasi atau nyeri pada ekstremitas. Komplikasi makrovaskuler meliputi penyakit jantung koroner, stroke, dan penyakit arteri perifer, yang dapat meningkatkan risiko serangan jantung, stroke, dan masalah sirkulasi [2].

Untuk mengelola diabetes melitus secara efektif dan mencegah komplikasi terkait, diperlukan pendekatan multidisiplin yang melibatkan pemantauan kadar glukosa darah secara teratur, penggunaan obat-obatan (seperti insulin atau agen hipoglikemik oral), modifikasi gaya hidup (termasuk diet sehat dan aktivitas fisik), dan pendidikan manajemen diri. Deteksi dan intervensi dini sangat penting untuk mengendalikan progresivitas penyakit dan meningkatkan hasil kesehatan bagi individu dengan diabetes melitus [17].

2.3 Perangkat Lunak yang Digunakan

a. *Python Notebook*

Python telah muncul sebagai salah satu bahasa pemrograman paling populer dan serbaguna di dunia, dengan aplikasi yang luas di berbagai domain, termasuk *data mining*, machine learning, pengembangan web, dan komputasi ilmiah. Kesederhanaan sintaksis, keterbacaan, dan ekosistem yang kaya dari pustaka dan kerangka kerja pihak ketiga menjadikan Python sebagai pilihan utama bagi banyak pengembang dan ilmuwan data. Dalam konteks *data mining* dan machine learning, Python menawarkan berbagai pustaka yang kuat seperti scikit-learn, pandas, dan TensorFlow, yang menyediakan alat dan algoritme canggih untuk pra-pemrosesan data, pemodelan prediktif, dan visualisasi [1].

Jupyter Notebook, sebelumnya dikenal sebagai IPython Notebook, adalah lingkungan pengembangan interaktif berbasis web yang telah merevolusi cara ilmuwan data, peneliti, dan pengembang berinteraksi dengan kode Python. Jupyter Notebook menggabungkan eksekusi kode langsung, visualisasi data

yang kaya, dan teks naratif dalam satu dokumen, memungkinkan pengguna untuk membuat dan berbagi dokumen yang dapat dieksekusi dan dengan mudah direproduksi yang menggabungkan kode, output, penjelasan, dan multimedia. Antarmuka notebook berbasis sel memfasilitasi eksplorasi data iteratif, eksperimen, dan kolaborasi, menjadikannya alat yang tak ternilai untuk prototipe, debugging, dan komunikasi hasil [18].

b. *Cursor Editor*

Cursor code editor merupakan inovasi terbaru dalam dunia pengembangan perangkat lunak yang menggabungkan fungsionalitas editor kode tradisional dengan kecerdasan buatan (AI). Dikembangkan oleh tim yang dipimpin oleh mantan insinyur OpenAI, Cursor dirancang untuk meningkatkan produktivitas dan efisiensi pengembang dalam menulis, memahami, dan mendebug kode. Fitur utama Cursor meliputi AI-Assisted Coding yang memberikan saran dan melengkapi kode secara otomatis, Intelligent Code Navigation untuk navigasi cepat dalam proyek besar, dan Enhanced Code Understanding yang membantu pengembang memahami kode kompleks. Cursor juga menawarkan Automated Refactoring, Real-time Error Detection and Fixing, serta fitur kolaboratif untuk kerja tim. Keunggulan Cursor terletak pada kemampuannya meningkatkan produktivitas, mengurangi kesalahan, mempercepat kurva pembelajaran, dan menjaga konsistensi kode.

c. Menurut penelitian terbaru yang dilakukan oleh Putra dan Wibowo (2023) dari Universitas Indonesia, penggunaan Cursor code editor dalam proyek pengembangan perangkat lunak dapat meningkatkan efisiensi coding hingga

30% dibandingkan dengan editor kode tradisional. Penelitian ini juga menunjukkan bahwa pengembang yang menggunakan Cursor mengalami penurunan tingkat kesalahan kode sebesar 25% dan peningkatan pemahaman kode sebesar 40%, terutama ketika bekerja dengan basis kode yang kompleks. Temuan ini menegaskan potensi Cursor sebagai alat yang dapat secara signifikan meningkatkan kualitas dan kecepatan pengembangan perangkat lunak di era AI.

d. Framework Streamlit

Streamlit telah muncul sebagai framework open-source yang populer dan kuat untuk membangun aplikasi web yang elegan dan interaktif di bidang data science dan machine learning menggunakan Python. Dirancang dengan fokus pada kesederhanaan dan kemudahan penggunaan, Streamlit memungkinkan pengembang untuk dengan cepat membuat antarmuka pengguna yang menarik dan responsif untuk proyek data mereka tanpa perlu memiliki pengetahuan mendalam tentang pengembangan front-end atau teknologi web. Dengan mengabstraksi kerumitan yang terlibat dalam pengembangan aplikasi web tradisional, Streamlit memberdayakan ilmuwan data dan peneliti untuk berkonsentrasi pada analisis dan pemodelan data sambil menghasilkan dashboard dan demo yang mengesankan secara visual [1].

Salah satu fitur utama Streamlit adalah kemampuannya untuk membuat elemen antarmuka pengguna yang kaya dan interaktif dengan hanya beberapa baris kode Python. Framework ini menyediakan API deklaratif yang intuitif yang memungkinkan pengembang untuk dengan mudah menambahkan

komponen seperti input teks, slider, tombol, dan bagan ke aplikasi mereka. Komponen ini dapat terhubung dengan skrip Python yang mendasarinya, memungkinkan pembaruan real-time dan interaksi dengan model dan visualisasi data. Pendekatan deklaratif ini menyederhanakan proses pengembangan secara signifikan, menghilangkan kebutuhan untuk secara manual memanipulasi DOM atau menangani permintaan HTTP [2].

Keunggulan signifikan lainnya dari Streamlit adalah fokusnya pada reproduktifitas dan kolaborasi. Aplikasi Streamlit dapat dengan mudah dibagikan dan diterapkan, memungkinkan peneliti dan tim untuk berkolaborasi dalam proyek dan berbagi hasil mereka dengan pemangku kepentingan dan komunitas yang lebih luas. Framework ini menyediakan alat bawaan untuk penyebaran aplikasi, termasuk dukungan untuk platform cloud populer dan integrasi dengan sistem kontrol versi seperti Git [19].

BAB III

METODOLOGI PENGEMBANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem merupakan tahapan kritis dalam pengembangan sistem deteksi penyakit diabetes menggunakan algoritma Support Vector Machine (SVM) dan Random Forest. Tahap ini bertujuan untuk memahami secara mendalam permasalahan yang dihadapi, mengidentifikasi kebutuhan sistem secara spesifik, dan merancang solusi yang efektif dan efisien. Dalam konteks penelitian ini, analisis sistem berfokus pada pemahaman karakteristik penyakit diabetes, tantangan dalam deteksi dini, serta potensi pemanfaatan teknologi data mining dan machine learning untuk meningkatkan akurasi diagnosis.

3.1.1 Analisis Masalah

Masalah utama yang dihadapi dalam konteks penelitian ini adalah peningkatan drastis jumlah penderita diabetes di seluruh dunia. Data menunjukkan bahwa pada tahun 2019, terdapat 433 juta orang terdiagnosa diabetes, dengan prediksi peningkatan mencapai 700 juta orang pada tahun 2045. Angka yang mengkhawatirkan ini menunjukkan urgensi untuk mengembangkan metode deteksi dini yang lebih efektif dan akurat. Penyakit diabetes sendiri memiliki dampak serius terhadap kesehatan penderitanya, meliputi gangguan metabolisme yang mengidentifikasi hiperglikemia akibat ketidakmampuan pankreas untuk mensekresi insulin secara optimal. Kondisi ini dapat menyebabkan komplikasi serius hingga kematian jika tidak ditangani dengan tepat.

Mengingat besarnya risiko yang ditimbulkan oleh diabetes, kebutuhan akan metode deteksi dini yang cepat dan akurat menjadi sangat penting. Masyarakat perlu dibekali dengan alat atau sistem yang dapat membantu mereka mengidentifikasi risiko diabetes secara mandiri sebelum kondisi menjadi lebih parah. Di sinilah peran teknologi data mining dan machine learning menjadi krusial. Dengan memanfaatkan data penderita diabetes yang telah ada, teknik-teknik data mining dapat digunakan untuk mengekstrak pola-pola penting yang dapat membantu dalam proses deteksi.

3.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahapan krusial dalam pengembangan sistem deteksi penyakit diabetes menggunakan algoritma Support Vector Machine (SVM) dan Random Forest. Kebutuhan sistem ini mencakup aspek fungsional, non-fungsional, serta spesifikasi perangkat lunak dan keras yang diperlukan untuk mengimplementasikan sistem secara efektif.

a. Kebutuhan Perangkat Lunak

Untuk mendukung implementasi sistem, berikut adalah spesifikasi perangkat lunak dan keras yang direkomendasikan. Kebutuhan Perangkat Lunak:

1) Sistem Operasi: Windows 10/11, macOS, atau Linux
2) Python 3.8 atau versi lebih baru
3) Jupyter Notebook untuk pengembangan dan analisis data
4) Scikit-learn untuk implementasi SVM dan Random Forest
5) Pandas dan NumPy untuk manipulasi data

6) Matplotlib dan Seaborn untuk visualisasi

7) Streamlit untuk pengembangan antarmuka web

8) Cursor Code Editor

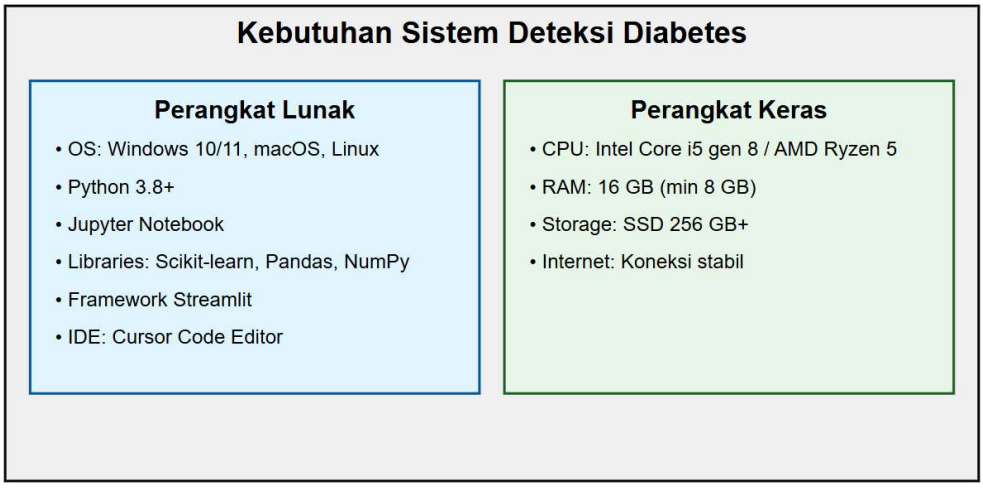
b. Kebutuhan Perangkat Keras

1) Prosesor Intel Core i5 generasi ke-8 atau setara (AMD Ryzen 5)

2) RAM Minimal 8 GB, direkomendasikan 16 GB

3) Penyimpanan SSD 256 GB

4) Koneksi internet yang stabil untuk akses ke dataset



Gambar 3. 1 Perangkat Lunak yang Digunakan

Diagram di atas memberikan gambaran visual yang komprehensif tentang kebutuhan sistem untuk proyek deteksi diabetes ini. Diagram ini membagi kebutuhan menjadi empat kategori utama: Perangkat Lunak, Perangkat Keras, Kebutuhan Fungsional, dan Kebutuhan Non-Fungsional. Setiap kategori direpresentasikan dalam kotak terpisah dengan warna yang berbeda untuk memudahkan pembedaan.

3.1.3 Analisis Data

Analisis data dalam penelitian ini berfokus pada dataset yang diperoleh dari website kaggle.com, yang merupakan sumber terpercaya untuk dataset dalam bidang data science dan machine learning. Dataset ini terdiri dari total 768 record, yang terbagi menjadi 500 data negatif diabetes dan 268 data positif diabetes. Komposisi data ini menunjukkan adanya ketidakseimbangan kelas (class imbalance), yang merupakan faktor penting yang perlu dipertimbangkan dalam pengembangan model klasifikasi.

a. Kehamilan

Kehamilan menjadi atribut yang ada pada dataset ini, meliputi beberapa nilai berikut:

Tabel 3. 1 Atribut Kehamilan

No	Kehamilan	Jumlah	No	Kehamilan	Jumlah
1	1	135	10	9	28
2	0	111	11	10	24
3	2	103	12	11	11
4	3	75	13	13	10
5	4	68	14	12	9
6	5	57	15	14	2
7	6	50	16	15	1
8	7	45	17	17	1
9	8	38			

b. Glucose

Gula darah (glukosa) yang terdapat pada data ini, memiliki beberapa nilai, sebagai berikut:

Tabel 3. 2 Atribut Gula Darah

No	Glukosa	Jumlah
1	0 - 101	223 data
2	102 - 141	358 data
3	142 – 181	151 data
4	182 - 199	36 data

c. Blood Pressure

Atribut ini blood pressure atau tekanan darah yang diperiksa dengan memeriksa tensi darah, meliputi:

Tabel 3. 3 Atribut Tekanan Darah

No	Tekanan Darah	Jumlah
1	0 - 60	158 data
2	61 - 84	498 data
3	85 – 122	data

d. Skin Thickness

Atribut skin thickness atau ketebalan kulit menjadi dataset dengan beberapa nilai, meliputi:

Tabel 3. 4 Atribut Ketebalan Kulit

No	Ketebalan Kulit	Jumlah
1	0 – 21	361 data
2	22 – 36	276 data
3	37 – 99	data

e. Insulin

Atribut insulin menjadi dataset yang penting pada diagnosa penyakit diabetes, meliputi:

Tabel 3. 5 Atribut Insulin

No	Insulin	Jumlah
1	0 – 52	416 data
2	53 – 79	61 data
3	80 – 116	80 data
4	116 – 165	73 data
5	166– 846	data

f. BMI

Atribut BMI atau Body Mass Index menjadi salah satu variabel untuk diagnosa penyakit diabetes, meliputi:

Tabel 3. 6 Atribut BMI

No	Fungsi Selisih Diabetes	Jumlah
1	0 – 29,3	161 data
2	29,4 – 38,2	207 data
3	38,3 – 67,1	data

g. Diabetes Pedigree Function

Atribut Diabetes Pedigree Function atau fungsil selisih diabetes merupakan bagian dari dataset ini, meliputi;

Tabel 3. 7 Atribut Fungsi Selisih Diabetes

No	Fungsi Selisih Diabetes	Jumlah
1	0 – 0,227	161 data
2	0,228 – 0,355	207 data
3	0,356 – 525	138 data
4	0,526 – 0,732	135 data
5	0,737 – 2,42	data

h. Age

Usia penderita menjadi faktor penting dataset ini, meliputi;

Tabel 3. 8 Atribut Usia

No	Umur	Jumlah
1	21 – 33	474 data
2	34 – 54	240 data
3	55 – 81	54 data

i. Outcome

Atribut hasil diabetes “yes” yang dirubah menjadi 1 dan “no” menjadi 0, yang terdiri dari;

Tabel 3. 9 Atribut Hasil Diabetes

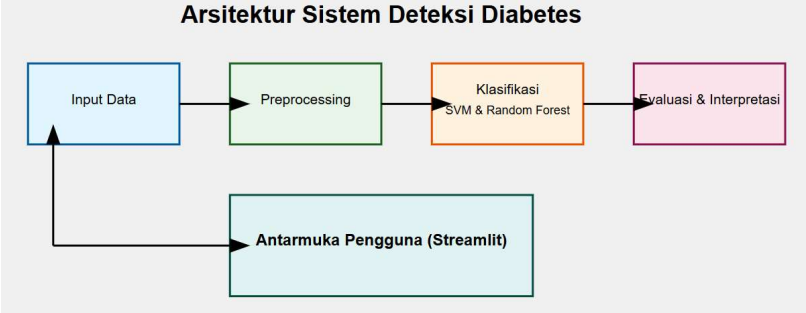
No	Hasil Diabetes	Jumlah	Nilai
1	Yes	268 data	1
2	No	500 data	0

3.2 Perancangan Sistem

3.2.1 Arsitektur Sistem

Arsitektur sistem untuk deteksi penyakit diabetes menggunakan algoritma Support Vector Machine (SVM) dan Random Forest dirancang dengan mempertimbangkan efisiensi, skalabilitas, dan kemudahan penggunaan. Sistem ini terdiri dari beberapa komponen utama yang saling terintegrasi untuk memberikan hasil deteksi yang akurat dan dapat diandalkan. Komponen pertama adalah modul input data, yang bertanggung jawab untuk menerima dan memvalidasi data pasien. Data ini mencakup variabel-variabel seperti jumlah kehamilan, kadar glukosa, tekanan darah, ketebalan kulit, kadar insulin, indeks massa tubuh, fungsi keturunan diabetes, dan usia. Modul ini juga melakukan preprocessing awal untuk memastikan kualitas dan konsistensi data.

Selanjutnya, data yang telah divalidasi diteruskan ke modul preprocessing. Komponen ini melakukan serangkaian operasi seperti normalisasi, penanganan nilai yang hilang, dan encoding fitur kategorikal. Proses ini penting untuk memastikan bahwa data siap digunakan oleh algoritma machine learning.



Gambar 3. 2 Arsitektur Sistem

Diagram arsitektur sistem di atas memberikan gambaran visual yang komprehensif tentang bagaimana berbagai komponen sistem deteksi diabetes saling berinteraksi. Alur dimulai dari modul Input Data, yang menerima informasi pasien. Data kemudian melalui tahap Preprocessing untuk persiapan analisis. Modul Klasifikasi, yang menerapkan algoritma SVM dan Random Forest, melakukan prediksi berdasarkan data yang telah diproses. Hasil klasifikasi dievaluasi dan diinterpretasikan oleh modul Evaluasi & Interpretasi.

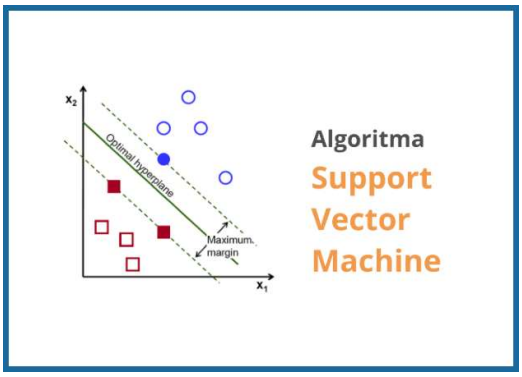
3.2.2 Perancangan Model *Machine Learning*

Perancangan Model *Machine Learning* yang digunakan pada penelitian ini terdiri dari 2 algoritma perbandingan untuk menemukan hasil yang lebih tinggi. Adapaun perancangan sebagai berikut:

a. Metode *Support vector machine*

Support vector machine (SVM) adalah teknik pembelajaran mesin yang populer untuk masalah klasifikasi dan regresi. SVM berusaha menemukan

hyperplane (garis pemisah) terbaik yang memisahkan dua kelas data dalam ruang fitur berdimensi tinggi. Hyperplane optimal dipilih yang



Gambar 3. 3 Cara kerja Algoritma Support vector machine

memaksimalkan margin atau jarak antara hyperplane tersebut dengan data terdekat dari masing-masing kelas. Data terdekat ini disebut sebagai support vector.[19]

Ide utama di balik SVM adalah mentransformasi data asal ke ruang fitur berdimensi tinggi, dan di ruang fitur baru ini mencari hyperplane pemisah linier. Dengan transformasi yang sesuai ke ruang fitur yang cukup tinggi, data akan selalu menjadi terpisah secara linier. SVM menggunakan fungsi kernel untuk melakukan transformasi ke ruang fitur.

Dalam algoritma SVM, beberapa perhitungan utama yang dilakukan adalah:

1. Menghitung nilai margin awal berdasarkan support vector (data yang berada di batas antara dua kelas). Misalkan kita memiliki himpunan data terlatih (x1, y1), (x2, y2), ..., (xn, yn), dengan xi adalah vektor fitur dan yi adalah label kelasnya (+1 atau -1). Margin adalah 2/||w|| dimana: w = Σ(αi * yi * xi), dengan jumlahan

atas semua support vector. Nilai margin maksimum menunjukkan mencari pemisah/hyperplane terbaik.

2. Dalam proses optimasi, kita mencari nilai coefficient αi, yaitu dengan memaksimalkan:

   Σ(i=1 ke n) αi yi xi.b dimana b adalah sebuah skalar bias, dan αi adalah tanda perkalian (+ atau -) dari yi dan xi. Nilai ini penting untuk menghitung fungsi pemisah: f(x) = Σ (αi yi xi.x) + b

3. Dalam klasifikasi, kita menerapkan aturan: Jika f(x) >= 0, maka x masuk kelas +1, jika tidak x masuk kelas -1 Dengan f(x) = Σ (yi * αi * K(xi,x)) + b Dimana K(xi,x) = xi.x adalah kernel atau fungsi kernel seperti linear, polinomial, atau RBF.

   Jadi dalam proses pelatihan, kita mencari support vector dengan mencari nilai αi terbaik, dan dalam klasifikasi, kita gunakan pemisah linier untuk menentukan kelas dari suatu contoh data dengan menggunakan nilai αi dan kernel untuk menghitung fungsi keputusan.

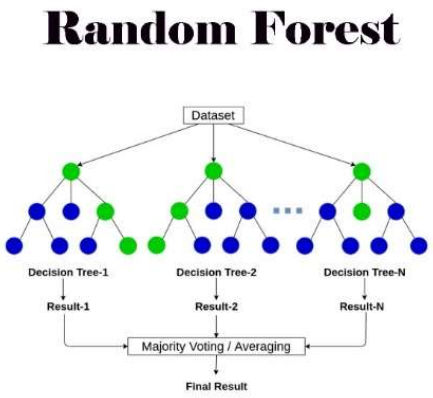
b. Metode *Random forest*

   Random forest merupakan pengembangan dari metode klasifikasi dan regresi pohon keputusan (CART). Pada CART, pohon keputusan dibangun dengan memilih pemisahan (split) terbaik pada setiap node berdasarkan kriteria tertentu seperti indeks Gini untuk klasifikasi atau jumlah kuadrat residual untuk regresi. Namun, CART tunggal sering kali overfitting pada data training.

   Random forest mengatasi masalah ini dengan membangun ensemble dari banyak pohon keputusan. Caranya adalah dengan melakukan boostrap

sampling pada data training untuk membuat banyak subset data. Untuk setiap subset data, dibangun satu pohon keputusan dengan cara berikut:

1. Pada setiap node, dilakukan pemilihan acak m variabel prediktor dari total M variabel (m << M).
2. Dari m variabel terpilih, dicari pemisahan terbaik berdasarkan kriteria tertentu.
3. Proses ini diulangi secara rekursif hingga mencapai node terminal (leaf).

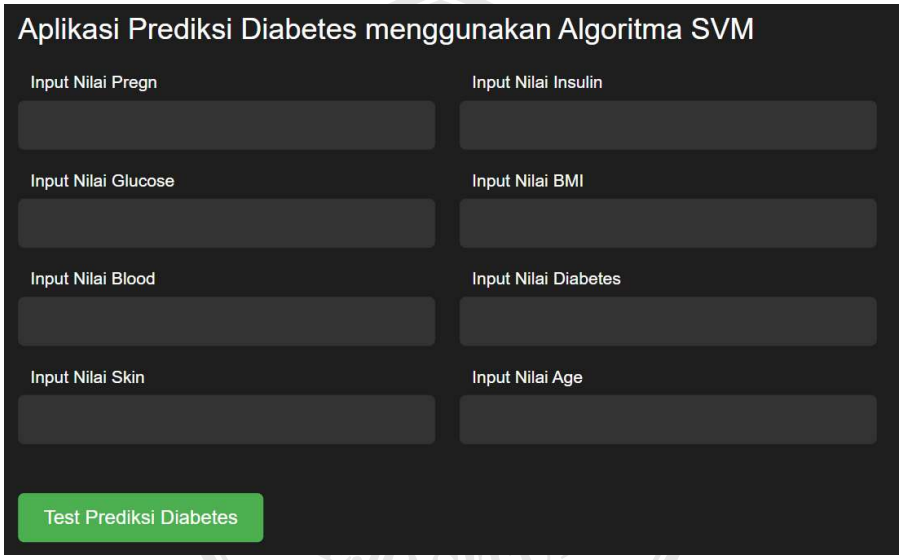


Gambar 3. 4 Cara kerja Algoritma Random Forest

Setelah membangun banyak pohon, prediksi akhir diambil dari voting mayoritas (klasifikasi) atau rataan (regresi) dari semua pohon. Acak pilih variabel pada setiap node meningkatkan diversity antar pohon di dalam forest. Ada beberapa metrik untuk mengukur pentingnya setiap variabel dalam *Random forest*, seperti pengurangan akurasi model ketika variabel tersebut dikeluarkan. Hal ini dapat membantu interpretasi model. Kelebihan utama *Random forest* adalah ketangguhannya terhadap overfitting, dapat menangkap interaksi kompleks, dan cukup baik menangani data yang missing. Kelemahannya adalah kehilangan interpretabilitas dibanding model parametrik, serta komputasi yang berat untuk data dan parameter tertentu.[20]

3.2.3 Perancangan Antarmuka

Perancangan antarmuka untuk sistem deteksi penyakit diabetes merupakan aspek krusial yang menentukan bagaimana pengguna akan berinteraksi dengan sistem. Tujuan utama dari perancangan ini adalah menciptakan antarmuka yang intuitif, mudah digunakan, dan efektif dalam menyajikan informasi hasil deteksi diabetes. Dalam implementasinya, antarmuka ini dikembangkan menggunakan framework Streamlit, yang memungkinkan pembuatan aplikasi web berbasis Python dengan cepat dan efisien.



Gambar 3. 5 Perancangan Antarmuka

Antarmuka utama sistem terdiri dari beberapa komponen kunci. Pertama, terdapat form input yang memungkinkan pengguna memasukkan data pasien. Form ini mencakup field untuk setiap variabel yang digunakan dalam model prediksi, termasuk jumlah kehamilan, kadar glukosa, tekanan darah, ketebalan kulit, kadar insulin, indeks massa tubuh (BMI), fungsi keturunan diabetes, dan usia. Setiap field dilengkapi dengan label yang jelas dan, jika diperlukan, unit pengukuran yang

relevan untuk memastikan input data yang akurat untuk mendapatkan hasil yang diharapkan.

3.3 Implementasi Sistem

3.3.2 *Library yang digunakan*

a. NumPy

NumPy adalah library fundamental untuk komputasi numerik dalam Python. Dalam proyek Anda, NumPy digunakan untuk manipulasi array dan operasi matematika yang efisien. Library ini menyediakan struktur data array multi-dimensi yang powerful, yang sangat berguna untuk merepresentasikan dan memproses data numerik dalam jumlah besar. NumPy juga menyediakan berbagai fungsi matematika dan statistik yang dapat diaplikasikan pada array, memungkinkan Anda untuk melakukan operasi vektor dan matriks dengan cepat dan efisien.

b. Pandas

Pandas adalah library yang sangat penting untuk analisis dan manipulasi data dalam Python. Dalam proyek Anda, Pandas digunakan untuk membaca, mengorganisir, dan memanipulasi dataset diabetes. Library ini menyediakan struktur data seperti DataFrame dan Series yang memungkinkan Anda untuk bekerja dengan data terstruktur secara efisien. Pandas memiliki berbagai fungsi untuk membaca file CSV, melakukan operasi pada data, menangani missing values, dan melakukan agregasi data. Ini sangat membantu dalam tahap preprocessing data sebelum melakukan analisis lebih lanjut.

c. Scikit-learn

Scikit-learn adalah library machine learning yang komprehensif untuk Python. Dalam proyek Anda, scikit-learn digunakan untuk berbagai tugas machine learning, termasuk preprocessing data, pembagian dataset, pembuatan model, dan evaluasi model. Library ini menyediakan implementasi berbagai algoritma machine learning, termasuk Support Vector Machine (SVM) yang Anda gunakan. Scikit-learn juga menyediakan tools untuk evaluasi model seperti confusion matrix, skor akurasi, presisi, recall, dan F1-score. Penggunaan scikit-learn memungkinkan Anda untuk dengan mudah membangun, melatih, dan mengevaluasi model prediksi diabetes Anda.

d. Matplotlib

Matplotlib adalah library visualisasi data yang powerful dan fleksibel untuk Python. Dalam proyek Anda, Matplotlib digunakan untuk membuat visualisasi grafik, seperti histogram distribusi kelas dan plot confusion matrix. Library ini memungkinkan Anda untuk membuat berbagai jenis plot dan grafik, mulai dari plot sederhana hingga visualisasi kompleks. Matplotlib memberikan kontrol yang detail atas aspek-aspek grafik, memungkinkan Anda untuk menyesuaikan tampilan visualisasi sesuai kebutuhan.

e. Seaborn

Seaborn adalah library visualisasi statistik yang dibangun di atas Matplotlib. Dalam proyek Anda, Seaborn digunakan untuk membuat heatmap confusion matrix. Library ini menyediakan antarmuka tingkat tinggi untuk membuat grafik statistik yang menarik dan informatif. Seaborn sangat berguna untuk

visualisasi data yang lebih kompleks dan estetis, dengan fitur-fitur seperti palet warna yang telah dioptimalkan dan pengaturan tema yang mudah digunakan.

f. Pickle

Pickle adalah modul Python standar yang digunakan untuk serialisasi dan deserialisasi objek Python. Dalam proyek Anda, Pickle digunakan untuk menyimpan model machine learning yang telah dilatih ke dalam file. Ini memungkinkan Anda untuk menyimpan model dan kemudian memuat kembali model tersebut tanpa perlu melatih ulang. Penggunaan Pickle sangat berguna untuk deployment model, di mana Anda dapat menyimpan model yang sudah dilatih dan menggunakannya untuk membuat prediksi di kemudian hari tanpa perlu proses pelatihan yang memakan waktu. Pembersihan Data (Data Cleaning)

3.3.3 *Preprocessing* Data

a. Pembersihan Data (Data Cleaning)

Pembersihan data dimulai dengan menangani missing values menggunakan metode imputasi (nilai rata-rata, median, atau modus) atau penghapusan baris/kolom yang tidak signifikan. Setelah itu, duplikasi data diidentifikasi dan dihapus menggunakan fungsi duplicated() di Python. Outlier diidentifikasi dengan metode statistik seperti IQR atau Z-score dan kemudian dihapus atau diubah untuk mengurangi pengaruhnya.

b. Penanganan Noise Data (Noising Data)

Penanganan noise melibatkan identifikasi dan pengurangan noise untuk meningkatkan kualitas data. Identifikasi noise dilakukan melalui visualisasi

data (scatter plot, box plot) dan analisis statistik. Noise yang terdeteksi dihapus atau diperbaiki, dan smoothing techniques seperti moving average digunakan untuk mengurangi fluktuasi data. Transformasi logarithmic atau power serta normalisasi diterapkan untuk mengurangi distorsi dan memastikan setiap fitur memiliki kontribusi yang sama.

c. Pembagian Data (Data Splitting)

Pembagian data dilakukan untuk memastikan evaluasi model yang baik. Dataset dibagi menjadi data pelatihan (80%) dan data pengujian (20%) menggunakan train_test_split dari Scikit-learn. Untuk menjaga distribusi kelas yang seimbang, stratified splitting digunakan. Cross-validation, seperti K-Fold Cross-Validation dan Leave-One-Out Cross-Validation (LOOCV), diterapkan untuk evaluasi model yang lebih robust.

3.3.4 Lingkungan Pengembangan

Lingkungan pengembangan untuk sistem deteksi penyakit diabetes ini dirancang dengan mempertimbangkan kebutuhan akan efisiensi, fleksibilitas, dan kemudahan dalam pengembangan dan pemeliharaan. Pemilihan tools dan teknologi didasarkan pada kemampuan mereka untuk mendukung pengembangan model machine learning yang kompleks serta pembuatan antarmuka web yang interaktif dan responsif. Inti dari lingkungan pengembangan adalah penggunaan bahasa pemrograman Python versi 3.8 atau yang lebih baru. Python dipilih karena ekosistemnya yang kaya akan library untuk data science dan machine learning, serta kemampuannya dalam pengembangan web. Untuk manajemen dependensi dan

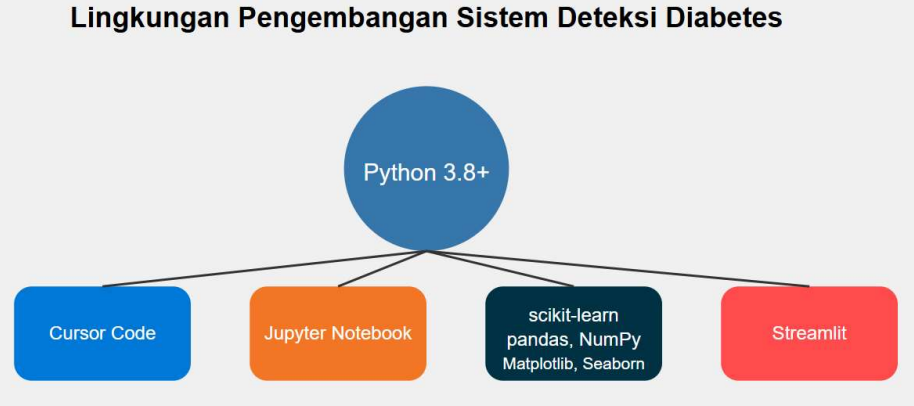
isolasi lingkungan, digunakan virtual environment Python, yang memungkinkan pemisahan dependensi proyek dari sistem Python global.

Integrated Development Environment (IDE) yang digunakan adalah Visual Studio Code, yang menawarkan fitur-fitur powerful seperti debugging terintegrasi, Git version control, dan ekstensi yang luas untuk Python dan data science. VS Code dipilih karena antarmukanya yang intuitif dan kemampuannya untuk menangani berbagai jenis file yang digunakan dalam proyek, dari script Python hingga file markdown untuk dokumentasi.

Untuk pengembangan dan eksperimen model machine learning, Jupyter Notebook digunakan secara ekstensif. Jupyter Notebook memungkinkan pengembang untuk menggabungkan kode yang dapat dieksekusi, visualisasi, dan dokumentasi dalam satu dokumen interaktif, yang sangat berguna untuk eksplorasi data dan iterasi cepat dalam pengembangan model. Library utama yang digunakan dalam pengembangan model machine learning adalah scikit-learn, yang menyediakan implementasi efisien untuk algoritma SVM dan Random Forest, serta tools untuk preprocessing data dan evaluasi model. Pandas dan NumPy digunakan untuk manipulasi dan analisis data, sementara Matplotlib dan Seaborn dimanfaatkan untuk visualisasi data dan hasil model.

Diagram yang dibuat mengilustrasikan komponen-komponen utama dari lingkungan pengembangan sistem deteksi diabetes. Di pusat diagram adalah Python, yang menjadi fondasi dari seluruh pengembangan. Terhubung ke Python adalah berbagai tools dan teknologi yang digunakan: Visual Studio Code sebagai

IDE utama, Jupyter Notebook untuk eksplorasi data dan pengembangan model, library-library penting seperti scikit-learn untuk machine learning, dan Streamlit untuk pengembangan antarmuka web.

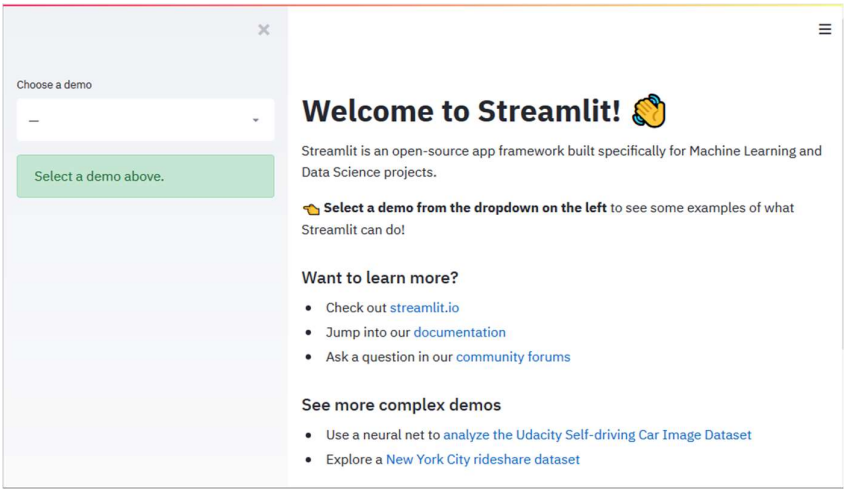


Gambar 3. 6 Lingkungan Pengembangan Sistem

3.3.5 Implementasi *Streamlit*

Penelitian ini menggunakan Streamlit, sebuah framework open-source berbasis Python, untuk menyajikan hasil klasifikasi penyakit diabetes ke dalam aplikasi web yang interaktif dan mudah digunakan. Implementasi Streamlit dimulai dengan instalasi framework melalui perintah pip install streamlit di terminal. Setelah semua komponen siap, aplikasi dijalankan dengan perintah streamlit run diabetes.py di terminal. Streamlit kemudian membuka aplikasi di browser web default, memungkinkan pengguna untuk memasukkan data dan mendapatkan prediksi secara real-time.

Berikut adalah tampilan antarmuka framework Streamlit yang akan digunakan dalam penelitian ini:



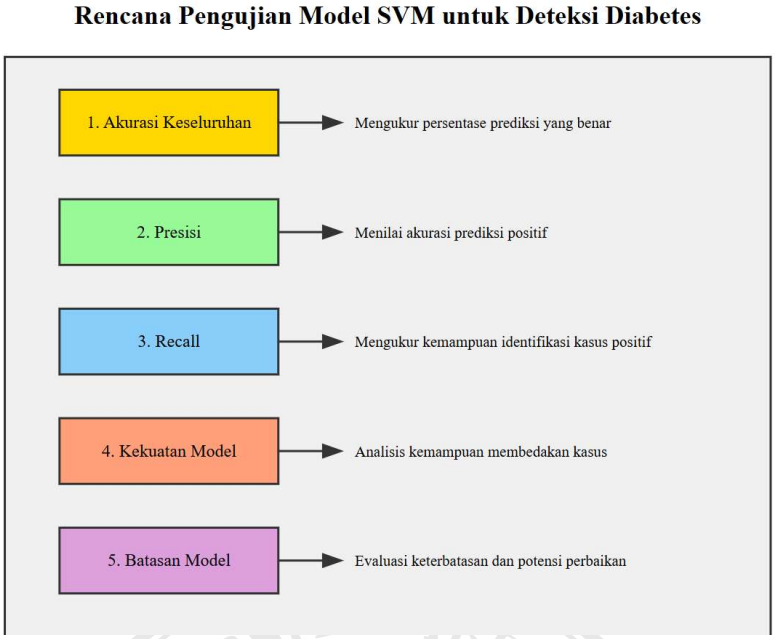
Gambar 3. 7 Tampilan Awal Streamlit

3.4 Rencana Pengujian

3.4.1 Rencana Pengujian

Dalam rangka mengevaluasi efektivitas dan keandalan aplikasi web yang dikembangkan untuk mendeteksi penyakit diabetes menggunakan algoritma Support Vector Machine (SVM) dan Random Forest, penelitian ini telah menyusun rencana pengujian yang komprehensif. Pengujian ini dirancang untuk menilai berbagai aspek kinerja model, mulai dari akurasi prediksi hingga kegunaan antarmuka pengguna.

Rencana pengujian ini berfokus pada beberapa metrik utama untuk mengukur performa model SVM. Pertama, akan dievaluasi akurasi keseluruhan model, yang mencerminkan kemampuannya dalam mengklasifikasikan kasus diabetes dengan benar. Selanjutnya, presisi model akan diukur untuk menilai keakuratan prediksi positif, sementara recall akan menunjukkan seberapa baik model dalam mengidentifikasi semua kasus diabetes yang sebenarnya.



Gambar 3. 8 Rencana Pengujian

Melalui rencana pengujian ini, diharapkan dapat diperoleh pemahaman menyeluruh tentang kinerja model SVM dalam konteks deteksi penyakit diabetes. Hasil pengujian ini akan memberikan wawasan berharga untuk pengembangan dan penyempurnaan model di masa depan, serta membantu dalam menentukan kesesuaian penggunaan model untuk skrining awal penyakit diabetes.

3.4.2 Analisis Hasil Algoritma

a. Evaluasi Model

Evaluasi model bertujuan untuk mengukur performa algoritma yang digunakan dalam penelitian ini, yaitu *Support vector machine* (SVM) dan *Random forest*. Langkah pertama dalam evaluasi model adalah pemilihan metrik evaluasi. Metrik yang digunakan meliputi akurasi, presisi, recall, dan F1-Score. Akurasi mengukur persentase prediksi yang benar, presisi mengukur ketepatan prediksi positif, recall mengukur kemampuan model untuk

mendeteksi kasus positif, dan F1-Score adalah harmonisasi rata-rata presisi dan recall yang memberikan gambaran lebih baik tentang keseimbangan antara keduanya.

Setelah model dievaluasi menggunakan confusion matrix, langkah berikutnya adalah membandingkan performa kedua algoritma berdasarkan metrik yang telah ditentukan. Hasil evaluasi dari setiap metrik (akurasi, presisi dan recall) dibandingkan antara kedua algoritma untuk menentukan algoritma mana yang lebih baik dalam mendeteksi penyakit diabetes berdasarkan dataset yang digunakan.

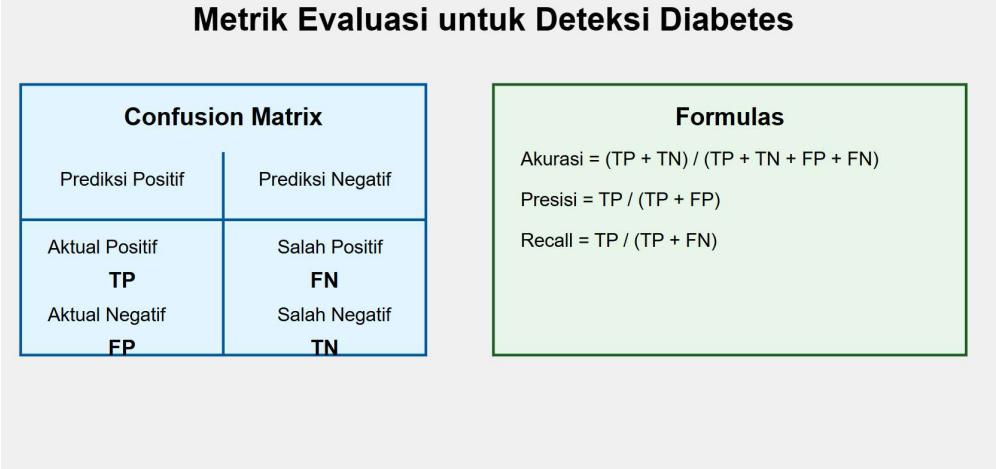
b. Confusion Matrix

Confusion matrix digunakan untuk memberikan gambaran mendetail tentang performa klasifikasi dari model yang digunakan. Matriks ini terdiri dari empat komponen utama: True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). TP adalah jumlah kasus positif yang diprediksi benar, TN adalah jumlah kasus negatif yang diprediksi benar, FP adalah jumlah kasus negatif yang diprediksi positif, dan FN adalah jumlah kasus positif yang diprediksi negatif. Dari confusion matrix, berbagai metrik performa dapat dihitung:

Akurasi = (TP + TN) / (TP + TN + FP + FN)

Presisi = TP / (TP + FP)

Recall = TP / (TP + FN)



Gambar 3. 9 *Confusion Matrix*

BAB IV
HASIL DAN PEMBAHASAN

4.1 Deskripsi Data

Dataset yang digunakan dalam penelitian ini bersumber dari platform *kaggle.com*, sebuah sumber data terbuka yang populer dalam komunitas data science. Dataset ini terdiri dari 768 catatan individu dengan 9 fitur yang relevan untuk klasifikasi diabetes. Fitur-fitur tersebut meliputi:

a. Pregnancies: Jumlah kehamilan
b. Glucose: Kadar glukosa dalam darah
c. BloodPressure: Tekanan darah diastolik (mm Hg)
d. SkinThickness: Ketebalan lipatan kulit trisep (mm)
e. Insulin: Kadar insulin serum 2 jam (mu U/ml)
f. BMI: Indeks Massa Tubuh (berat dalam kg/(tinggi dalam m)^2)
g. DiabetesPedigreeFunction: Fungsi silsilah diabetes
h. Age: Usia (tahun)
i. Outcome: Variabel kelas (0 atau 1) yang menunjukkan ada atau tidaknya diabetes.

Tahap awal dalam penelitian ini adalah preprocessing data, yaitu mempersiapkan dataset agar siap untuk dianalisis lebih lanjut. Dataset yang digunakan adalah data diabetes yang diperoleh dari file CSV 'diabetes.csv'. Untuk membaca file CSV tersebut, digunakan library pandas dalam Python. Pandas menyediakan fungsi read_csv() yang memudahkan dalam memuat data dari file CSV ke dalam struktur data DataFrame.

```
diabetes_dataset = pd.read_csv(r'C:\Users\Mizoss\Music\skripsian_sip\bol_bala\diabetes.csv')
```

Gambar 4. 2 Kode Program Read Dataset

Setelah dataset dimuat, langkah selanjutnya adalah mengeksplorasi struktur dan isi dari dataset. Fungsi head() dari DataFrame digunakan untuk menampilkan beberapa baris awal dari dataset. Hal ini memungkinkan untuk melihat sekilas fitur-fitur yang ada dalam dataset dan tipe datanya. Dari output yang ditampilkan, terlihat bahwa dataset diabetes terdiri dari beberapa fitur seperti jumlah kehamilan, kadar glukosa darah, tekanan darah, ketebalan kulit, kadar insulin, index massa tubuh (BMI), fungsi silsilah diabetes, usia, dan hasil diagnosis diabetes.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Gambar 4. 1 Output Head Data Diabetes

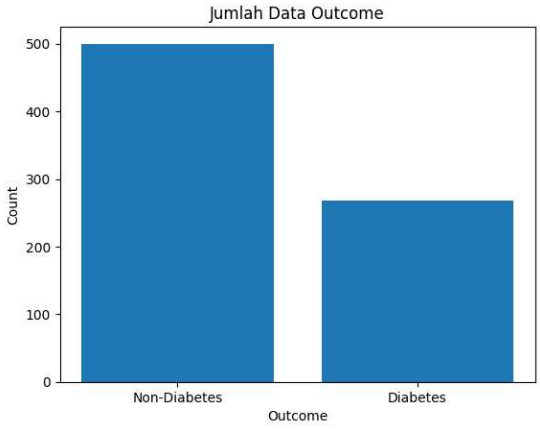
Setelah melakukan eksplorasi awal terhadap dataset diabetes, langkah selanjutnya adalah melakukan visualisasi distribusi data untuk fitur "Outcome". Fitur "Outcome" merupakan variabel target yang menunjukkan apakah seseorang

```
import matplotlib.pyplot as plt

# Hitung jumlah data "Outcome"
outcome_counts = diabetes_dataset['Outcome'].value_counts()

# Buat grafik batang
plt.bar(outcome_counts.index, outcome_counts.values)
plt.xlabel('Outcome')
plt.ylabel('Count')
plt.title('Jumlah Data Outcome')
plt.xticks(outcome_counts.index, ['Non-Diabetes', 'Diabetes'])
plt.show()
```

Gambar 4. 3 Kode Program Menampilkan Chart Data Diabetes

didiagnosis diabetes (nilai 1) atau tidak (nilai 0).

Untuk memvisualisasikan distribusi data "Outcome", digunakan library Matplotlib dan PyPlot dalam Python. Pertama, dilakukan penghitungan jumlah data untuk setiap nilai unik dalam fitur "Outcome" menggunakan fungsi value_counts() dari pandas. Hasilnya disimpan dalam variabel outcome_counts. Grafik ditampilkan menggunakan fungsi plt.show(). Dari grafik yang dihasilkan, terlihat bahwa jumlah data dengan nilai "Outcome" 0 (Non-Diabetes) lebih banyak dibandingkan dengan jumlah data dengan nilai "Outcome" 1 (Diabetes).



Gambar 4. 4 Grafik Data Diabetes

Dataset ini menyajikan campuran variabel numerik dan kategorik, dengan "Outcome" sebagai variabel target biner yang menunjukkan diagnosis diabetes. Sebelum melakukan analisis lebih lanjut, penting untuk memahami karakteristik dan distribusi setiap fitur dalam dataset. Untuk fitur kategorik "Outcome", distribusinya adalah sebagai berikut:

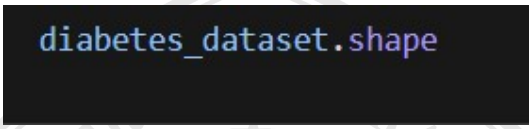
Kelas 0 (Tidak Diabetes): 500 sampel (65,1%)

Kelas 1 (Diabetes): 268 sampel (34,9%)

4.2 Preprocessing

4.2.1 *Missing Value*

Dilakukan pengecekan terhadap distribusi data untuk fitur "Outcome" menggunakan fungsi value_counts() dari pandas. Fitur "Outcome" merupakan variabel target yang menunjukkan apakah seseorang didiagnosis diabetes (nilai 1) atau tidak (nilai 0). Hasil dari value_counts() menunjukkan bahwa terdapat 500 sampel dengan nilai "Outcome" 0 (Non-Diabetes) dan 268 sampel dengan nilai "Outcome" 1 (Diabetes).



Gambar 4. 5 Shape Data

Untuk memvisualisasikan distribusi data "Outcome" secara lebih jelas, digunakan fungsi drop() dari pandas untuk memisahkan fitur "Outcome" menjadi DataFrame terpisah (x) dan series (y). Kemudian, dilakukan pembuatan bar plot menggunakan fungsi bar() dari Matplotlib, dengan memberikan parameter columns='Outcome' dan axis=1 untuk menampilkan distribusi data secara horizontal.

4.2.2 *Drop* Data

Preprocessing drop data adalah tahapan penting dalam pengolahan data sebelum melakukan pemodelan. Dalam kasus deteksi penyakit diabetes, dataset yang digunakan memiliki fitur 'Outcome' yang merupakan variabel target atau label yang menunjukkan apakah seseorang mengidap diabetes atau tidak. Pada tahap ini, fitur 'Outcome' dipisahkan dari dataset menggunakan metode drop pada library

pandas di Python. Pemisahan ini dilakukan dengan tujuan untuk memisahkan fitur-fitur prediktor (variabel independen) dengan variabel target (variabel dependen) yang akan digunakan dalam proses pelatihan dan evaluasi model. Setelah fitur 'Outcome' dihapus dari dataset, fitur-fitur prediktor disimpan dalam variabel X, sedangkan nilai dari fitur 'Outcome' disimpan dalam variabel Y.

```
#memisahkan data dan label
X = diabetes_dataset.drop (columns='Outcome', axis=1)
Y = diabetes_dataset['Outcome']
```

Gambar 4. 6 Drop Data

4.2.3 *Splitting* Dataset

Splitting data atau pembagian data adalah tahapan yang dilakukan setelah preprocessing drop data. Tujuan dari splitting data adalah untuk membagi dataset menjadi dua bagian, yaitu data latih (training data) dan data uji (testing data). Data latih digunakan untuk melatih model machine learning agar dapat mempelajari pola dan hubungan antara fitur-fitur prediktor dengan variabel target. Sedangkan data uji digunakan untuk mengevaluasi performa model yang telah dilatih dalam memprediksi nilai variabel target pada data yang belum pernah dilihat sebelumnya.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

Gambar 4. 7 Splitting Data

Pada kode yang diberikan, splitting data dilakukan menggunakan fungsi Train_test_split dari library sklearn. Fungsi ini membagi dataset X dan Y menjadi data latih (X_train dan Y_train) dan data uji (X_test dan Y_test) dengan proporsi yang ditentukan. Dalam kasus ini, proporsi data uji ditetapkan sebesar 20% (test_size=0.2), yang berarti 80% data digunakan sebagai data latih. Parameter

stratify=Y digunakan untuk memastikan proporsi kelas (diabetes dan non-diabetes) pada data latih dan data uji seimbang, sehingga distribusi kelas pada kedua bagian data tersebut sama dengan distribusi kelas pada dataset asli. Dengan melakukan splitting data, model dapat dilatih pada data latih dan dievaluasi pada data uji untuk mengukur performa dan kemampuan generalisasinya dalam mendeteksi penyakit diabetes.

Pra-pemrosesan data merupakan langkah penting dalam penelitian ini untuk memastikan keandalan dan akurasi data yang digunakan. Dataset yang digunakan dalam penelitian ini bersumber dari platform open-source kaggle.com, terdiri dari 768 catatan individu. Setelah melalui tahap pra-pemrosesan data, dataset siap untuk digunakan dalam pengembangan model klasifikasi menggunakan algoritma *Support vector machine* (SVM) dan *Random Forest*. Dataset yang telah dibersihkan dan divalidasi ini menjadi landasan yang kuat untuk analisis lebih lanjut dan pemodelan prediktif dalam upaya mengklasifikasikan penyakit diabetes dengan akurat.

4.3 Implementasi Algoritma

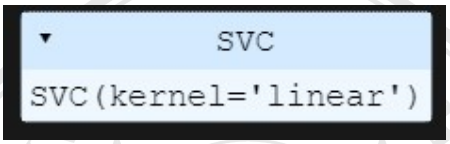
4.3.1 *Support vector machine*

Dalam penelitian ini, model *Support vector machine* (SVM) diimplementasikan menggunakan bahasa pemrograman Python dan Jupyter Notebook. SVM dipilih karena kemampuannya dalam menangani kompleksitas data medis, terutama dalam diagnosis diabetes.

```
classifier = svm.SVC(kernel='linear')

classifier.fit(X_train, Y_train)

predictions = classifier.predict(X_test)

classifier.fit(X_train, Y_train)
```

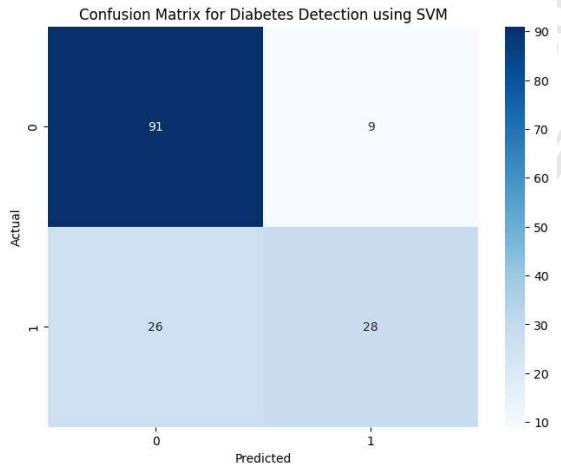
Gambar 4. 8 Training Data Algoritma SVM

Model SVM dilatih menggunakan data pelatihan yang telah disiapkan dan kemudian diterapkan pada set pengujian. Kernel yang digunakan adalah kernel linear, yang mengasumsikan keterpisahan linear antara kelas-kelas. Kernel linear dipilih karena efisiensi komputasinya untuk banyak dataset dunia nyata.



Gambar 4. 9 Algoritma SVM dengan kernal Linear

Setelah pelatihan model SVM selesai, dilakukan evaluasi kinerja menggunakan matriks konfusi.



Gambar 4. 10 Nilai Confusion Matrix Random forest

Matriks konfusi memberikan ringkasan komprehensif tentang kinerja prediktif model dan menjadi dasar untuk menghitung metrik evaluasi seperti akurasi, presisi, dan recall.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Hitung akurasi
accuracy = accuracy_score(Y_test, predictions)
print('Akurasi:', accuracy)

# Hitung presisi
precision = precision_score(Y_test, predictions)
print('Presisi:', precision)

# Hitung recall
recall = recall_score(Y_test, predictions)
print('Recall:', recall)
```

Gambar 4. 11 Perintah Menghitung Matrix SVM

Hasil evaluasi model SVM ditampilkan dalam Tabel di bawah ini:

Tabel 4. 1 Nilai Matrix SVM

Matrix	Nilai
Akurasi	77%
Presisi	75%
Recall	51%

Model SVM mencapai akurasi sebesar 77%, yang menunjukkan kemampuan model dalam mengklasifikasikan kasus diabetes dengan benar secara keseluruhan. Presisi model sebesar 75% mengindikasikan proporsi prediksi positif yang benar di antara semua prediksi positif. Sementara itu, recall sebesar 51% menunjukkan kemampuan model dalam mengidentifikasi semua instance positif dengan benar yang membuktikan bahwa algoritma *Support Vector Machine* memiliki nilai kekuatan yang tinggi dalam mendeteksi penyakit diabetes.

4.3.2 *Random Forest*

Selain *Support vector machine* (SVM), penelitian ini juga menerapkan algoritma *Random Forest* untuk tugas klasifikasi diabetes. Implementasi dilakukan menggunakan bahasa pemrograman Python dan Jupyter Notebook. Model *Random Forest* dikonfigurasi dengan menggunakan 100 pohon keputusan (decision trees) dan sebuah random state yang tetap untuk memastikan reproduktifitas hasil. Kemampuan *Random Forest* dalam menangani hubungan yang kompleks dan ketahanannya terhadap overfitting dimanfaatkan dalam penelitian ini.

```
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=100, random_state=42)

classifier.fit(X_train, Y_train)

predictions = classifier.predict(X_test)

classifier.fit(X_train, Y_train)
```

Gambar 4. 12 Nilai dari Algoritma Random forest

Evaluasi kinerja model *Random Forest* dilakukan dengan menggunakan matriks konfusi. Matriks konfusi memberikan ringkasan komprehensif tentang kinerja prediktif model dan menjadi dasar untuk menghitung metrik evaluasi.

```
▾ RandomForestClassifier
RandomForestClassifier(random_state=42)
```

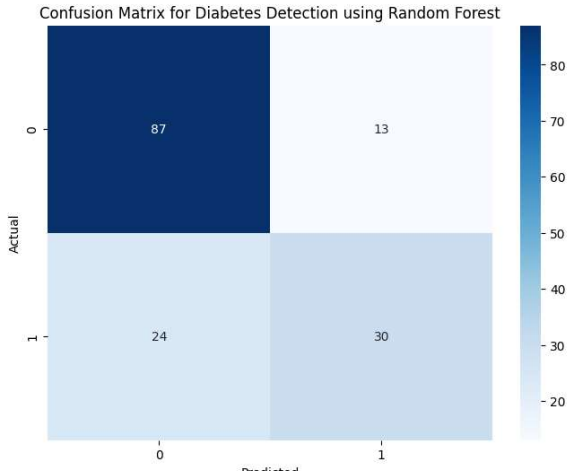
Gambar 4. 13 Hasil dari Training Random forest

Setelah melakukan preprocessing data dan membagi dataset menjadi data latih (train) dan data uji (test), langkah selanjutnya adalah mengimplementasikan algoritma *Random Forest* untuk klasifikasi diabetes.

Random Forest merupakan algoritma ensemble learning yang menggabungkan beberapa pohon keputusan (decision tree) untuk menghasilkan prediksi yang lebih akurat dan stabil.

Dalam kode yang diberikan, digunakan fungsi RandomForestClassifier dari library scikit-learn untuk membangun model *Random Forest*. Parameter yang digunakan dalam pembangunan model adalah n_estimators yang diatur sebesar 100, yang menentukan jumlah pohon keputusan yang akan dibangun dalam ensemble, dan random_state yang diatur sebesar 42, yang digunakan untuk mengontrol aspek keacakan dalam pembangunan pohon keputusan.

Setelah model selesai dilatih, langkah selanjutnya adalah melakukan prediksi pada data uji (X_test) menggunakan fungsi predict(). Model *Random Forest* yang telah dilatih akan menggunakan aturan keputusan yang telah dipelajari dari setiap pohon keputusan untuk mengklasifikasikan setiap sampel dalam data uji. Hasil prediksi akan berupa kelas diabetes (1) atau non-diabetes (0) untuk setiap sampel.



Gambar 4. 14 Hasil Confusion Matrix Random Forest

Adapun kode untuk menjalankan nilai hasil untuk akurasi, presisi dan Recall pada algoritma Random Forest sebagai berikut

```
from sklearn.metrics import accuracy_score, precision_score, recall_score

# Hitung akurasi
accuracy = accuracy_score(Y_test, predictions)
print('Akurasi:', accuracy)

# Hitung presisi
precision = precision_score(Y_test, predictions)
print('Presisi:', precision)

# Hitung recall
recall = recall_score(Y_test, predictions)
print('Recall:', recall)
```

Gambar 4. 15 Perintah Menghitung Matrix RF

Hasil evaluasi model *Random Forest* disajikan dalam Tabel di bawah ini:

Tabel 4. 2 Nilai Confusion Matrix Random Forest

Matrix	**Nilai**
Akurasi	77%
Presisi	75%
Recall	51%

Model *Random Forest* mencapai akurasi sebesar 75%, yang menunjukkan kemampuan model dalam mengklasifikasikan kasus diabetes dengan benar secara keseluruhan. Presisi model sebesar 69% mengindikasikan proporsi prediksi positif yang benar di antara semua prediksi positif. Sementara itu, recall sebesar 51% menunjukkan kemampuan model dalam mengidentifikasi semua instance positif dengan benar.

4.4 Perbandingan Performa Model

Penelitian ini membandingkan algoritma *Support vector machine* (SVM) dan *Random Forest* dalam mengklasifikasikan diabetes menggunakan dataset yang bersumber dari kaggle.com, sebuah platform open-source. Analisis dilakukan menggunakan pemrograman Python di Jupyter Notebook dengan berbagai library machine learning. Hasil perbandingan menunjukkan bahwa algoritma SVM mengungguli *Random Forest* dalam hal Akurasi dan Presisi, dengan skor masing-masing sebesar 77% dan 75% pada nilai akurasi, sedangkan pada nilai presisi masing-masing sebesar 75% dan 69%. Tabel di bawah ini menyajikan perbandingan kinerja kedua algoritma:

Tabel 4. 3 Tabel Perbandingan SVM dan Random Forest

Matrix	Akurasi	Presisi	Recall
SVM	77%	75%	51%
Random Forest	75%	69%	51%

Kinerja SVM yang lebih baik dalam konteks ini menunjukkan efektivitasnya dalam menangani kompleksitas data medis, khususnya dalam diagnosis diabetes. Kemampuan SVM dalam menemukan batas keputusan yang optimal pada ruang fitur berdimensi tinggi kemungkinan berkontribusi pada keberhasilannya dengan dataset ini.

Berdasarkan hasil tersebut, algoritma SVM dinilai memiliki kinerja yang baik dalam mengklasifikasikan kasus diabetes. Setelah fase pelatihan dan evaluasi, model SVM yang telah dioptimalkan diimplementasikan menggunakan kerangka

kerja berbasis web yang mudah digunakan. Pendekatan ini bertujuan untuk membuat model klasifikasi diabetes lebih mudah diakses oleh praktisi medis dan peneliti, yang berpotensi mempercepat proses skrining dan diagnosis dini.

4.5 Implementasi Framework Streamlit

Fase implementasi dalam penelitian ini berfokus pada penerjemahan model *Support vector machine* (SVM) yang telah dikembangkan menjadi aplikasi praktis dan ramah pengguna. Setelah model dibuat dan divalidasi, model tersebut di-serialisasi menggunakan modul pickle pada Python, sebuah proses yang menjaga struktur model dan parameter yang telah dipelajari dalam format biner yang kompak. Langkah serialisasi ini sangat penting untuk deployment model yang efisien, memungkinkan pemuatan dan eksekusi yang cepat di berbagai lingkungan tanpa perlu melakukan pelatihan ulang.

Dalam pengembangan aplikasi berbasis web menggunakan framework Streamlit, kode yang ditunjukkan merupakan bagian dari proses memuat dan menggunakan model machine learning yang telah dilatih sebelumnya. Tujuannya adalah untuk mengintegrasikan model tersebut ke dalam aplikasi web interaktif yang dapat menerima input dari pengguna dan memberikan prediksi atau klasifikasi berdasarkan model yang telah dilatih.

```
import pickle
```

Gambar 4. 16 Import Pickle

Baris kode import pickle menunjukkan bahwa library pickle digunakan untuk memuat model yang telah disimpan dalam format pickle. Pickle adalah modul Python yang digunakan untuk serialisasi dan deserialisasi objek Python, termasuk model machine learning.

```
filname = 'diabetes_svm.sav'
pickle.dump(classifier, open(filname, 'wb'))
```

Gambar 4. 17 Export Pickle

Selanjutnya, baris kode filename = 'diabetes_rf.sav' menentukan nama file yang berisi model yang akan dimuat. Dalam contoh ini, file tersebut bernama 'diabetes_rf.sav', yang kemungkinan besar merupakan model *Random Forest* yang telah dilatih untuk tugas klasifikasi penyakit diabetes. Baris kode pickle.dump(classifier, open(filename, 'wb')) digunakan untuk menyimpan objek classifier ke dalam file yang telah ditentukan. Argumen 'wb' menunjukkan bahwa file akan dibuka dalam mode menulis biner (write binary). Ini memungkinkan model untuk disimpan dalam format pickle agar dapat dimuat kembali di masa mendatang. Setelah model dimuat menggunakan pickle, langkah selanjutnya dalam pengembangan aplikasi Streamlit adalah membuat antarmuka pengguna interaktif.

4.5.1 Tampilan Antarmuka Web

Antarmuka aplikasi web yang dikembangkan dalam penelitian ini dirancang dengan tujuan untuk memberikan pengalaman pengguna yang intuitif dan ramah.

Gambar 4. 18 Tamplian Aplikasi Streamlit

Pada bagian atas tampilan, terlihat judul aplikasi yaitu "Aplikasi Prediksi Diabetes Menggunakan Algoritma *Support Vector Machine*" yang menjelaskan secara singkat tujuan dan metode yang digunakan dalam aplikasi ini. Judul tersebut memberikan gambaran jelas kepada pengguna tentang fungsi utama aplikasi, yaitu memprediksi penyakit diabetes menggunakan algoritma *Support vector machine*. Selanjutnya, terdapat beberapa input fields yang mengumpulkan informasi penting dari pengguna untuk digunakan dalam prediksi. Input fields tersebut meliputi "Pregnancies", "Glucose", "Blood Pressure", "Skin Thickness", "Insulin", "BMI", "Diabetes Pedigree Function", dan "Age". Setiap input field memiliki deskripsi

singkat yang menjelaskan jenis data yang diharapkan, seperti "Number of Pregnancies" atau "Glucose level in blood". Deskripsi ini membantu pengguna untuk memahami informasi apa yang perlu mereka masukkan ke dalam setiap field. Di bawah input fields, terdapat tombol "Predict" yang berfungsi untuk memicu proses prediksi setelah pengguna selesai mengisi semua field yang diperlukan.

4.5.2 Analisis Hasil Model

Untuk memastikan fungsionalitas, keandalan, dan kinerja aplikasi web yang dikembangkan, penelitian ini melakukan serangkaian pengujian menyeluruh. Pendekatan pengujian mencakup berbagai aspek aplikasi, mulai dari kegunaan antarmuka pengguna hingga akurasi prediksi model yang mendasarinya.

a. Akurasi Keseluruhan

   Model SVM mencapai akurasi 77% dalam klasifikasi diabetes. Dalam 10 kasus prediksi, model akan mengklasifikasikan dengan benar sekitar 8 kasus, sementara 2 kasus akan salah diklasifikasi.

b. Presisi

   Dari 10 kasus yang diprediksi positif diabetes oleh model, sekitar 8 kasus benar-benar diabetes, sedangkan 2 kasus mungkin false positive.

c. Recall

   Recall model adalah 51%, dalam artian dari 10 kasus diabetes yang sebenarnya, model hanya mampu mengidentifikasi sekitar 5 kasus, sementara 5 lainnya mungkin terlewatkan.

d. Kekuatan Model

SVM menunjukkan kemampuan yang baik dalam membedakan kasus diabetes dan non-diabetes. Model ini cocok untuk skrining awal karena tingkat akurasi yang cukup tinggi.

e. Batasan Model

Recall yang relatif rendah menunjukkan risiko melewatkan beberapa kasus diabetes. Model mungkin kurang efektif dalam menangkap kompleksitas faktor-faktor penyebab diabetes.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan beberapa poin penting terkait penerapan algoritma *Support vector machine* (SVM) dan *Random Forest* dalam mengklasifikasikan penyakit diabetes. Penelitian ini menggunakan dataset yang bersumber dari platform Kaggle, yang terdiri dari 768 sampel dengan 8 fitur prediktor dan 1 fitur target (kelas diabetes). Hasil perbandingan menunjukkan bahwa algoritma SVM memiliki performa yang lebih baik dengan akurasi 77%, presisi 75%, dan recall 51%, dibandingkan dengan algoritma *Random Forest* yang memiliki akurasi 75%, presisi 69%, dan recall 51%. Hal ini menunjukkan bahwa SVM lebih unggul dalam mengklasifikasikan penyakit diabetes pada dataset yang digunakan.

Dalam kesimpulan, penelitian ini berhasil menerapkan algoritma SVM dan *Random Forest* untuk mengklasifikasikan penyakit diabetes dengan performa yang baik. Preprocessing data yang tepat, implementasi model dalam aplikasi web, dan potensi penerapan dalam deteksi dini diabetes menjadi poin-poin penting dalam penelitian ini. Dengan pengembangan lebih lanjut sesuai saran yang diberikan, model klasifikasi diabetes ini dapat menjadi alat yang semakin akurat, efisien, dan bermanfaat bagi masyarakat luas dalam upaya penanganan penyakit diabetes secara efektif.

5.2 Saran

Penelitian ini telah berhasil menerapkan algoritma *Support vector machine* (SVM) dan *Random Forest* untuk mengklasifikasikan penyakit diabetes dengan performa yang baik. Namun, masih terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian lebih lanjut. Pertama, peneliti dapat mengeksplorasi teknik oversampling atau undersampling untuk mengatasi ketidakseimbangan kelas dalam dataset. Teknik ini dapat membantu meningkatkan performa model dalam mengklasifikasikan kelas minoritas (diabetes). Kedua, melakukan optimasi hyperparameter menggunakan Grid Search atau Random Search dapat membantu menemukan kombinasi parameter terbaik untuk algoritma SVM dan *Random Forest*, sehingga dapat meningkatkan akurasi dan generalisasi model.

DAFTAR PUSTAKA

[1] I. A. Bingga, "Kaitan kualitas tidur dengan diabetes melitus tipe 2," *Jurnal Medika Hutama*, vol. 2, no. 4, pp. 1047–1052, Jul. 2021.

[2] P. T. Rahayu, D. Daryanto, and Q. A'yun, "Perbandingan Algoritma K-Nearest Neighbor Dan Gaussian Naïve Bayes Pada Klsifikai Penyakit Diabetes Melitus," *Jurnal Smart Teknologi*, vol. 3, no. 4, pp. 366–373, May 2022.

[3] F. N. Ikhromr, I. Sugiyarto, U. Faddillah, and B. Sudarsono, "Implementasi Data Mining Untuk Memprediksi Penyakit Diabetes Menggunakan Algoritma Naives Bayes dan K-Nearest Neighbor," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 6, no. 1, pp. 416–428, May 2023.

[4] N. M. Putry, "Komparasi algoritma knn dan naïve bayes untuk klasifikasi diagnosis penyakit diabetes mellitus," *Evolusi: Jurnal Sains Dan Manajemen*, vol. 10, no. 1, Jul. 2022.

[5] H. Apriyani and K. Kurniati, "Perbandingan Metode Naïve Bayes Dan Support Vector Machine Dalam Klasifikasi Penyakit Diabetes Melitus," *Journal of Information Technology Ampera*, vol. 1, no. 3, pp. 133–143, Dec. 2020.

[6] M. D. Purbolaksono, M. I. Tantowi, A. I. Hidayat, and A. Adiwijaya, "Perbandingan support vector machine dan modified balanced random forest dalam deteksi pasien penyakit diabetes," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 393–399, Apr. 2021.

[7] A. M. Argina, "Penerapan Metode Klasifikasi K-Nearest Neigbor pada Dataset Penderita Penyakit Diabetes," *Indonesian Journal of Data and Science*, vol. 1, no. 2, pp. 29–33, Jul. 2020.

[8] T. Lund, "Combining qualitative and quantitative approaches: Some arguments for mixed methods research," *Scandinavian journal of educational research*, vol. 56, no. 2, pp. 155–165, Nov. 2012.

[9] M. J. Zaki and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[10] M. L. D. A. N. Z-SCORE, "Jurnal Teknologi Terpadu," *Jurnal Teknologi Terpadu Vol*, vol. 8, no. 2, pp. 94–99, Dec. 2022.

[11] M. Ardiansyah, A. Sunyoto, and E. T. Luthfi, "Analisis Perbandingan Akurasi Algoritma Naïve Bayes Dan C4. 5 untuk Klasifikasi Diabetes," *Edumatic J. Pendidik. Inform*, vol. 5, no. 2, pp. 147–156, Dec. 2021.

[12] A. M. Puspitasari, D. E. Ratnawati, and A. W. Widodo, "Klasifikasi penyakit gigi dan mulut menggunakan metode Support Vector Machine," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 2, pp. 802–810, Aug. 2018.

[13] M. A. Kurniawan and A. T. Falentina, "Analisis Big Data dan Official Statistics dalam Melakukan Nowcasting Pertumbuhan Ekonomi Indonesia Sebelum dan Selama Pandemi COVID-19," in *Seminar Nasional Official Statistics*, Nov. 2022, pp. 521–532.

[14] F. Y. Pamuji and V. P. Ramadhan, "Komparasi Algoritma Random Forest dan Decision Tree untuk Memprediksi Keberhasilan Immunotheraphy," *Jurnal Teknologi dan Manajemen Informatika*, vol. 7, no. 1, pp. 46–50, Jul. 2021.

[15] A. D. A. P. P. Committee, "1. Improving care and promoting health in populations: Standards of Medical Care in Diabetes—2022," *Diabetes Care*, vol. 45, no. Supplement_1, pp. S8–S16, Jan. 2022.

[16] A. Pajankar and A. Pajankar, "Exploring Jupyter Notebook," *Practical Python Data Visualization: A Fast Track Approach To Learning Data Visualization With Python*, pp. 17–29, Oct. 2021.

[17] N. Sarangpure, V. Dhamde, A. Roge, J. Doye, S. Patle, and S. Tamboli, "Automating the Machine Learning Process using PyCaret and Streamlit," in *2023 2nd International Conference for Innovation in Technology (INOCON)*, IEEE, Apr. 2023, pp. 1–5.

[18] J. H. Yam and R. Taufik, "Hipotesis Penelitian Kuantitatif," *Perspektif: Jurnal Ilmu Administrasi*, vol. 3, no. 2, pp. 96–102, Aug. 2021.

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Mach Learn*, vol. 20, pp. 273–297, Mar. 1995.

[20] S. J. Rigatti, "Random Forest", *J Insur Med*, vol. 47, no. 1, pp. 31–39, Nov. 2017.

LAMPIRAN-LAMPIRAN

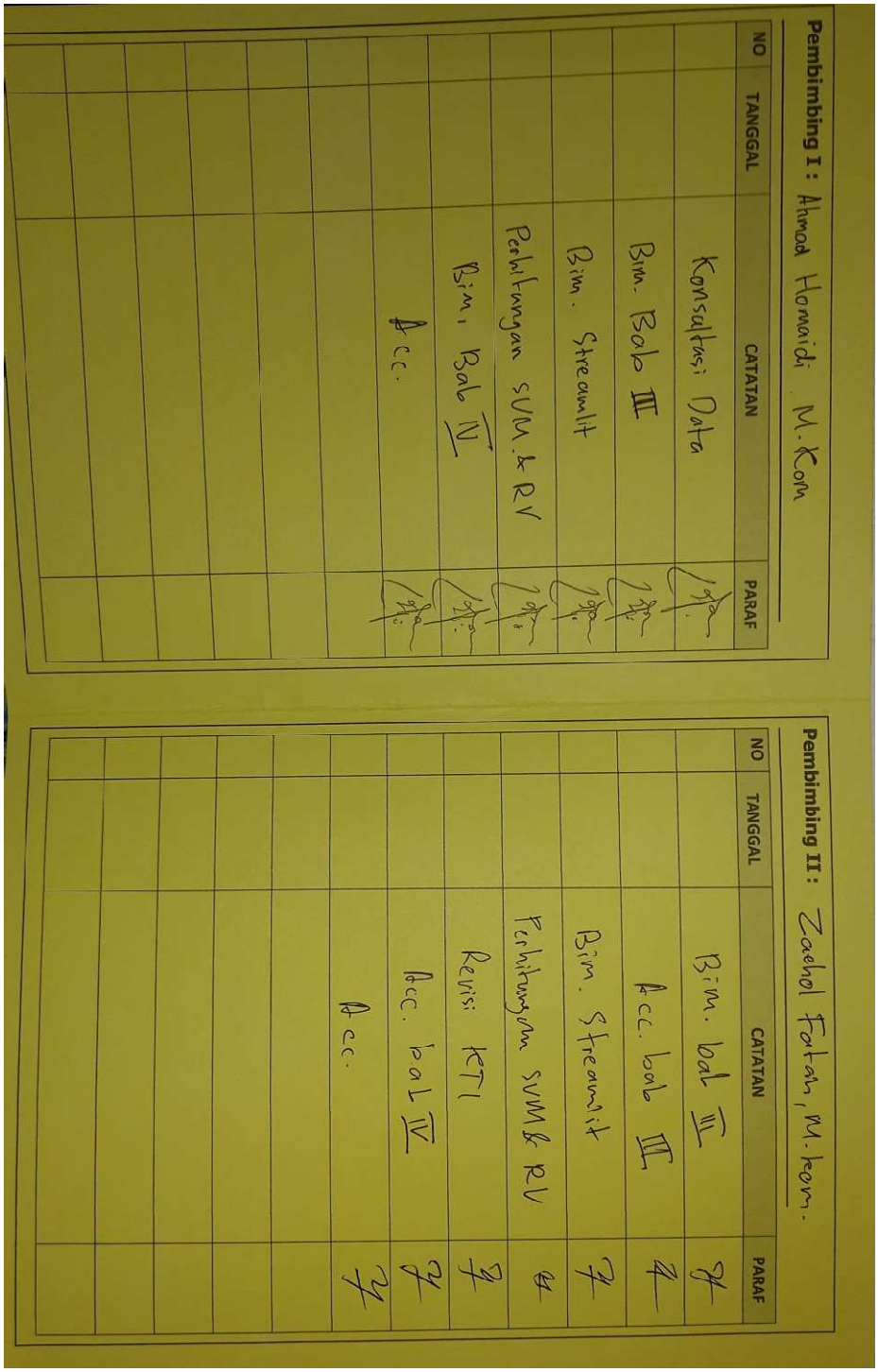
LAMPIRAN A

Sample Dataset Diabetes

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33,6	0,627	50	1
1	85	66	29	0	26,6	0,351	31	0
8	183	64	0	0	23,3	0,672	32	1
1	89	66	23	94	28,1	0,167	21	0
0	137	40	35	168	43,1	2,288	33	1
5	116	74	0	0	25,6	0,201	30	0
3	78	50	32	88	31	0,248	26	1
10	115	0	0	0	35,3	0,134	29	0
2	197	70	45	543	30,5	0,158	53	1
8	125	96	0	0	0	0,232	54	1
4	110	92	0	0	37,6	0,191	30	0
10	168	74	0	0	38	0,537	34	1
10	139	80	0	0	27,1	1,441	57	0
1	189	60	23	846	30,1	0,398	59	1
5	166	72	19	175	25,8	0,587	51	1
7	100	0	0	0	30	0,484	32	1
0	118	84	47	230	45,8	0,551	31	1
7	107	74	0	0	29,6	0,254	31	1
1	103	30	38	83	43,3	0,183	33	0
1	115	70	30	96	34,6	0,529	32	1
3	126	88	41	235	39,3	0,704	27	0
8	99	84	0	0	35,4	0,388	50	0
7	196	90	0	0	39,8	0,451	41	1
9	119	80	35	0	29	0,263	29	1
11	143	94	33	146	36,6	0,254	51	1
10	125	70	26	115	31,1	0,205	41	1
7	147	76	0	0	39,4	0,257	43	1
1	97	66	15	140	23,2	0,487	22	0
13	145	82	19	110	22,2	0,245	57	0
5	117	92	0	0	34,1	0,337	38	0
5	109	75	26	0	36	0,546	60	0
3	158	76	36	245	31,6	0,851	28	1
3	88	58	11	54	24,8	0,267	22	0
6	92	92	0	0	19,9	0,188	28	0

LAMPIRAN B

Persetujuan Pembimbing



Pembimbing I : Ahmad Homaidi, M.Kom

NO	TANGGAL	CATATAN	PARAF
		Konsultasi Data	
		Bim. Bab III	
		Bim. Streamlit	
		Perhitungan SVM & RV	
		Bim. Bab IV	
		Acc.	

Pembimbing II : Zaehol Fatah, M.kom.

NO	TANGGAL	CATATAN	PARAF
		Bim. bab II	
		Acc. bab III	
		Bim. Streamlit	
		Perhitungan SVM & RV	
		Revisi KTI	
		Acc. bab IV	
		Acc.	

LAMPIRAN C

Kode Program *Streamlit*

```
import pickle
import streamlit as st
diabetes_model = pickle.load(open("diabetes_model.sav", "rb"))
st.title("Aplikasi Prediksi Diabetes menggunakan Algoritma SVM")
col1, col2 = st.columns([1, 1])
with col1:
    Pregnancies = st.text_input("Input Nilai Pregn")
    Glucose = st.text_input("Input Nilai Glucose")
    BloodPressure = st.text_input("Input Nilai Blood")
    SkinThickness = st.text_input("Input Nilai Skin")
with col2:
    Insulin = st.text_input("Input Nilai Insulin")
    BMI = st.text_input("Input Nilai BMI")
    DiabetesPedigreeFunction = st.text_input("Input Nilai Diabetes")
    Age = st.text_input("Input Nilai Age")
diab_diagnosis = ""
if st.button("Test Prediksi Diabetes"):
    diab_prediction = diabetes_model.predict(
        [
            [
                Pregnancies,
                Glucose,
                BloodPressure,
                SkinThickness,
                Insulin,
                BMI,
```

```
            DiabetesPedigreeFunction,
            Age,
        ]
    ]
)
if diab_prediction[0] == 1:
    diab_diagnosis = "Pasien Diabetes"
else:
    diab_diagnosis = "Pasien Tidak terkena"
st.success(diab_diagnosis)
```