DOI: 10.22146/jnteti.v14i1.15198

Perbandingan Kinerja Algoritma KNN dan SVM Menggunakan SMOTE untuk Klasifikasi Penyakit Diabetes

Asri Mulyani[1], Sarah Khoerunisa[1], Dede Kurniadi[1]

[1] Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Institut Teknologi Garut, Garut, Jawa Barat, 44151, Indonesia.

[Diserahkan: 27 Juli 2024, Direvisi: 30 Oktober 2024, Diterima: 22 Januari 2025]
Penulis Korespondensi: Asri Mulyani (email: asrimulyani@itg.ac.id)

INTISARI — Diabetes seringkali tidak terdeteksi atau didiagnosis terlambat. Hal ini dapat menyebabkan berbagai komplikasi serius, seperti kerusakan organ, *stroke*, dan penyakit jantung. International Diabetes Federation (IDF) menyebutkan bahwa 10,5% dari populasi orang dewasa berusia 20 hingga 79 tahun didiagnosis menderita diabetes dan hampir setengahnya tidak menyadari kondisi tersebut, sehingga angka penderita diabetes terus meningkat secara signifikan, mencapai empat kali lipat dibandingkan dengan periode sebelumnya. Deteksi diabetes secara dini merupakan langkah penting bagi penderita untuk mencegah munculnya komplikasi, salah satunya dengan memanfaatkan teknologi kecerdasan buatan, yaitu penambangan data. Oleh sebab itu, diperlukan pengetahuan mengenai algoritma yang efektif digunakan untuk melakukan deteksi penyakit diabetes. Penelitian ini bertujuan untuk membandingkan dua algoritma, yakni *k-nearest neighbor* (KNN) dan *support vector machine* (SVM), dalam klasifikasi penyakit diabetes menggunakan *synthetic minority oversampling technique* (SMOTE). Penelitian ini menerapkan metode *machine learning life cycle* untuk mengukur kinerja kedua algoritma. Hasil penelitian menunjukkan bahwa kedua algoritma memiliki kinerja yang baik dalam mendeteksi diabetes, tetapi terdapat perbedaan kinerja yang signifikan antara keduanya. Algoritma SVM dengan kernel *radial basis function* (RBF) mencapai akurasi sebesar 81,67%, presisi 85,91%, *recall* 79,01%, dan *F*1-*score* 82,32%. Di sisi lain, algoritma KNN dengan nilai $k = 3$ yang ditemukan melalui *cross-validation* mencapai akurasi sebesar 83,33%, presisi 85,00%, *recall* 83,95%, dan *F*1-*score* 84,47%. Berdasarkan evaluasi *confusion matrix*, KNN menunjukkan kinerja yang lebih unggul dibandingkan SVM dalam hal akurasi dan metrik evaluasi lainnya. Hasil ini menunjukkan bahwa KNN lebih efektif dalam mendeteksi diabetes pada *dataset* yang digunakan dalam penelitian ini.

KATA KUNCI — Algoritma, Diabetes, *K-Nearest Neighbor*, SMOTE, *Support Vector Machine*.

I. PENDAHULUAN

Diabetes merupakan penyakit metabolik kronis yang terjadi karena tubuh tidak mampu menghasilkan insulin dalam jumlah yang cukup, sehingga menyebabkan tingginya kadar gula darah [1]. Penyakit ini memiliki dampak global yang signifikan. Menurut International Diabetes Federation (IDF), pada tahun 2021, sekitar 10,5% populasi orang dewasa (usia 20–79 tahun) menderita diabetes dan hampir setengahnya tidak menyadari kondisi tersebut. IDF memproyeksikan peningkatan drastis hingga 783 juta penderita pada tahun 2045, atau sekitar 46% dari populasi saat ini [2]. World Health Organization (WHO) melaporkan bahwa angka kematian akibat diabetes mencapai satu juta jiwa per tahun [3].

Tantangan utama dalam penanganan diabetes adalah sulitnya deteksi dini. Tanpa deteksi pada tahap awal, diabetes dapat mengakibatkan komplikasi serius, seperti kerusakan organ, *stroke*, dan penyakit jantung [4]. Meskipun memiliki gejala klinis yang khas, yang dikenal sebagai "triaspoli" (polidipsi, polifagi, poliuri) [5], gejala-gejala ini sering diabaikan atau tidak dikenali. Gejala lain, seperti kesemutan pada jari tangan, kelelahan berlebihan, penurunan berat badan signifikan, dan gangguan penglihatan, juga sering luput dari perhatian. Pengendalian diabetes yang baik sangat bergantung pada deteksi dini dan pengobatan yang konsisten [6].

Teknologi kecerdasan buatan (*artificial intelligence*, AI) menawarkan solusi potensial untuk meningkatkan akurasi deteksi dini diabetes. Melalui algoritma penambangan data, AI dapat menganalisis data historis pasien untuk memprediksi risiko diabetes. Dua algoritma yang sering digunakan adalah *k-nearest neighbor* (KNN) dan *support vector machine* (SVM). Namun, efektivitas kedua algoritma ini sering terhambat oleh ketidakseimbangan data antara jumlah pasien diabetes dan nondiabetes yang tidak proporsional.

Penelitian ini membandingkan kinerja algoritma KNN dan SVM dalam klasifikasi diabetes dengan menerapkan teknik *synthetic minority oversampling technique* (SMOTE) untuk mengatasi ketidakseimbangan kelas. Kedua algoritma ini dipilih karena memiliki keunggulan masing-masing dalam klasifikasi medis. KNN efektif dalam mengelompokkan data berdasarkan tetangga terdekatnya [7], sedangkan SVM mampu memisahkan kelas dengan margin optimal [8], sehingga cocok untuk *dataset* yang bervariasi. Dengan SMOTE, penelitian ini mengeksplorasi efektivitas KNN dan SVM dalam mengatasi ketidakseimbangan kelas, yang menjadi tantangan utama dalam klasifikasi medis. Selain itu, KNN dan SVM memiliki kinerja yang stabil pada *dataset* kecil hingga menengah [9], sehingga lebih efisien dibandingkan algoritma lain yang membutuhkan sumber daya lebih besar. Hasil penelitian ini diharapkan dapat memberikan pemahaman yang lebih baik tentang efektivitas kedua algoritma dalam mendeteksi diabetes, khususnya ketika diterapkan pada *dataset* yang tidak seimbang.

Bagian selanjutnya dari makalah ini memaparkan penelitian terkait serta definisi dan konsep yang relevan, sedangkan Bagian III membahas metodologi penelitian yang telah dilakukan, mulai dari pengumpulan *dataset* hingga evaluasi hasil. Bagian IV membahas hasil dari pengujian yang telah

dilakukan, kemudian Bagian V menyajikan kesimpulan berdasarkan hasil pengujian.

II. PENELITIAN TERKAIT

Beberapa penelitian untuk klasifikasi penyakit diabetes, dengan berbagai pendekatan dan algoritma pembelajaran mesin yang digunakan, telah dilakukan. Metode KNN diterapkan pada *dataset* individu dengan diabetes [10]. *Dataset* yang terdiri atas 77 data digunakan dalam penelitian ini. Sebanyak 10% dari *dataset* ini digunakan untuk pelatihan data dan 10% lagi untuk pengujian data. Pada penelitian ini dilakukan teknik prapemrosesan, termasuk membagi, membersihkan, dan mempersiapkan data. Dalam algoritma KNN, evaluasi dilakukan dengan mengukur akurasi, presisi, *recall*, dan *F-measure* untuk beberapa nilai *K*. Hasilnya menunjukkan bahwa akurasi mencapai 0,39 pada $K = 3$, presisi mencapai 0,65 pada $K = 3$ dan $K = 5$, *recall* mencapai 0,36 pada $K = 3$, dan *F-measure* tertinggi mencapai 0,46 pada $K = 3$.

Klasifikasi penyakit diabetes dengan SVM dilakukan pada *dataset* Pima Indians Diabetes [11]. Penelitian ini menggunakan *dataset Diabetes Database Pima Indians*, yang terdiri atas 768 sampel, dengan 268 sampel mewakili pasien diabetes dan 500 sampel mewakili pasien diabetes yang sehat. Penelitian ini menggunakan dua teknik prapemrosesan, yaitu penanganan ketidaksesuaian melalui identifikasi dan pembersihan *outlier* serta manajemen nilai nol pada komponen tertentu melalui imputasi nilai median. Selanjutnya, fitur diskalakan menggunakan metode normalisasi *min-max scaler*. Beberapa metrik, seperti akurasi, presisi, sensitivitas, dan spesifisitas, digunakan untuk menilai model. Akurasi rata-rata sebesar 0,87, presisi rata-rata 0,82, sensitivitas rata-rata 0,78, dan spesifisitas rata-rata 0,87 dicapai oleh model SVM yang dioptimasi. Model SVM awal, atau *scratch*, mencapai nilai rata-rata 0,78, presisi rata-rata 0,69, sensitivitas rata-rata 0,59, dan spesifisitas rata-rata 0,87.

Model SVM *radial basis function* (RBF) dengan bantuan *forward selection* diterapkan untuk memprediksi penyakit diabetes [12]. Sampel data yang digunakan terdiri atas 1.017 catatan pasien. Untuk memastikan bahwa semua fitur memiliki rentang nilai yang sama, prapemrosesan dimulai dengan normalisasi data. Data kemudian dibagi menjadi dua, yaitu 80% data latih dan 20% data uji. Untuk memastikan keandalan dan generalisasi model, evaluasi dilakukan dengan menggunakan 10-*fold cross validation*. Selain itu, *confusion matrix* digunakan untuk menilai kinerja model. Penelitian ini mencapai tingkat akurasi sebesar 91,2%. Model SVM RBF mampu memberikan tingkat akurasi yang lebih tinggi dengan menggunakan metode *forward selection*.

Algoritma KNN diterapkan untuk mengidentifikasi penyakit diabetes berdasarkan delapan gejala pokok [13]. *Dataset* yang digunakan terdiri atas 135 *record data* pasien, dengan 81 untuk data latih dan 54 untuk data uji, melalui proses pembersihan. Setelah dilakukan normalisasi, jarak tetangga terdekat dihitung dengan jarak Euclidean dengan nilai $K = 9$. Evaluasi model dilakukan menggunakan metrik seperti akurasi dan *confusion matrix*. Hasilnya menunjukkan bahwa 4 orang positif dan 50 orang negatif untuk diabetes. Evaluasi algoritma KNN dengan *matrix confusion* menunjukkan nilai akurasi sebesar 93%. Dengan demikian, penelitian ini berhasil menerapkan metode KNN dalam klasifikasi penyakit diabetes.

Teknik SMOTE diterapkan untuk klasifikasi Masyarakat Penerima BLT DD menggunakan algoritma *naïve Bayes* [14]. *Dataset* terdiri atas 375 *record* data, dengan 205 *record* terklasifikasi sebagai layak dan 170 *record* sebagai tidak layak. Hasil terbaik diperoleh dari pemodelan *naïve Bayes* dengan SMOTE, yang mencapai nilai akurasi sebesar 97,80% dan nilai AUC 0,99.

Dari rujukan penelitian yang telah dipaparkan, diketahui bahwa banyak penelitian sebelumnya telah menguji kinerja KNN dan SVM secara terpisah dalam mengklasifikasikan penyakit diabetes dengan hasil yang baik, tetapi sering kali tanpa menangani masalah ketidakseimbangan kelas dalam *dataset*. Selain itu, metode SMOTE telah terbukti efektif dalam mengatasi ketidakseimbangan kelas [14]. Dengan demikian, penelitian ini menekankan kebaruan dengan membandingkan kinerja algoritma KNN dan SVM melalui penerapan SMOTE untuk klasifikasi penyakit diabetes.

A. DIABETES

Kondisi diabetes dapat merusak organ-organ seperti jantung, pembuluh darah, mata, ginjal, dan saraf. Jenis diabetes yang umum adalah tipe satu dan tipe dua. Diabetes tipe satu biasanya terjadi pada orang dewasa dan ditandai oleh ketidakmampuan tubuh menggunakan insulin secara efektif atau kurangnya produksi insulin. Sementara itu, diabetes tipe satu adalah kondisi jangka panjang dengan kondisi pankreas hampir tidak sama sekali menghasilkan insulin [15].

Diabetes dapat menimbulkan gejala khas, seperti rasa haus berlebihan, sering buang air kecil, gangguan penglihatan, dan penurunan berat badan. Gejala yang paling parah dapat menyebabkan ketoasidosis, atau keadaan *hiperosmolar nonketotik*, yang dapat menyebabkan dehidrasi, koma, dan jika tidak diobati akan menyebabkan kematian. Namun, pada diabetes tipe dua, gejalanya sering kali tidak begitu parah atau bahkan tidak terlihat sama sekali karena perkembangan hiperglikemianya yang lambat. Oleh karena itu, tanpa pemeriksaan biokimia, tingkat gula darah yang mencukupi untuk menyebabkan perubahan patologis dan fungsional mungkin sudah ada dalam tubuh dengan jangka waktu yang lama sebelum diagnosis dapat dikonfirmasi, yang pada akhirnya dapat menyebabkan terjadinya komplikasi saat diagnosis akhirnya ditentukan.

B. K-NEAREST NEIGHBOR

KNN adalah algoritma pembelajaran berbasis *instance* yang masuk pada kategori *lazy learning*. Algoritma ini berfungsi dengan mencari sejumlah objek *k* dalam data latih yang hampir sama dengan objek pada data baru atau data uji. Oleh karena itu, jika ada sampel yang tidak teridentifikasi, sampel tersebut dapat diprediksi dengan melihat sampel yang tidak teridentifikasi di dekatnya [16]. Tahapan implementasi KNN meliputi menentukan nilai parameter *K*, menghitung jarak, mengurutkan jarak, menentukan *K* jarak terdekat, memetakan kelas yang sesuai, dan memilih kelas data untuk dievaluasi [17].

C. SUPPORT VECTOR MACHINE

SVM adalah metode yang digunakan untuk melakukan klasifikasi dan regresi. SVM bekerja dengan memisahkan kelas-kelas secara linier menggunakan *hyperplane*. Untuk masalah nonlinier, SVM menggunakan konsep *kernel*. Persamaan dasar SVM ditunjukkan pada (1).

$$f(x) = w^T x + b \quad (1)$$

dengan w^T adalah *transpose* dari vektor bobot, x adalah vektor masukan, dan b adalah *bias*. Kemudian, persamaan untuk klasifikasi linier disajikan dalam (2).

$$[(w^T.x_i) + b] \geq 1 \text{ untuk } y_i = +1$$
$$[(w^T.x_i) + b] \leq -1 \text{ untuk } y_i = -1 \quad (2)$$

dengan x_i adalah data masukan ke-*i* dan y_i adalah label kelas. *Best hyperplane* diperoleh dengan mengoptimalkan jarak antara dua kelompok objek dari kelas yang berbeda. Margin ini dihitung dengan rumus yang melibatkan norma dari vektor bobot *w*. Metode *quadratic programming* meminimalkan setengah dari norma kuadrat vektor bobot [18].

D. KLASIFIKASI

Klasifikasi adalah langkah krusial pada penambangan data yang melibatkan pembagian data atau objek baru dalam kategori atau label berdasarkan fitur-fitur tertentu [19]. Klasifikasi digunakan untuk mengategorikan *dataset*. Hal ini berbeda dengan pengelompokan, yang memiliki perbedaan dalam penggunaan variabel; pengelompokan tidak memerlukan variabel dependen, sedangkan klasifikasi membutuhkannya [20].

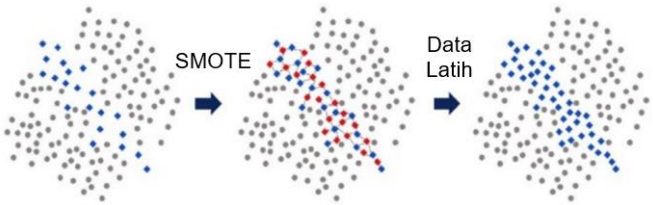
Supervised learning dan *unsupervised learning* adalah dua pendekatan yang digunakan dalam klasifikasi. Algoritma yang disebut *supervised learning* menciptakan fungsi yang menghubungkan masukan ke hasil yang diinginkan. Beberapa teknik yang digunakan dalam klasifikasi *supervised learning* adalah *random forest*, KNN, *naïve Bayes*, SVM, dan regresi logistik [21].

Pada klasifikasi terdapat tiga tahap utama, yakni pembangunan model, penerapan model, dan evaluasi. Pembangunan model mencakup pembuatan model menggunakan data latih dengan atribut dan kelas. Selanjutnya, model tersebut diterapkan pada data atau objek baru untuk menentukan kelasnya. Evaluasi dilakukan untuk menilai akurasi pembangunan dan menerapkan model untuk data baru. Proses klasifikasi terdiri atas dua tahap, yaitu pelatihan dan pengujian. Pada tahap pelatihan, data digunakan untuk membuat model, sedangkan pada tahap pengujian, model yang telah dibuat diuji dengan menggunakan data lain untuk menilai akurasinya.

E. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

SMOTE merupakan salah satu pendekatan untuk menangani data *imbalance* yang tidak sama dengan metode *oversampling* sebelumnya. Di dalam metode *oversampling*, prinsipnya adalah menambah jumlah sampel secara acak. Namun, dalam SMOTE pendekatan yang digunakan adalah penambahan sampel pada kelas minoritas agar seimbang dengan kelas mayoritas melalui pembuatan data sintetis atau buatan. Data buatan tersebut dihasilkan dengan mempertimbangkan atribut dari tetangga terdekat. Jumlah tetangga terdekat dapat ditentukan berdasarkan preferensi pengguna. Proses pembuatan data sintetis untuk atribut berskala numerik berbeda dengan proses pembuatan data sintetis untuk atribut berskala kategorikal [22]. Ilustrasi proses SMOTE ditampilkan pada Gambar 1 [23].

Pada data numerik, jarak ke tetangganya diukur menggunakan metode jarak Euclidean, sedangkan untuk data kategoris, digunakan nilai *value difference metric* (VDM). Proses analisis dimulai dengan pembagian data secara acak menjadi data uji (20%) dan data latih (80%). Selanjutnya, diterapkan metode SMOTE untuk menangani ketidakseimbangan kelas dengan fokus pada kelas minoritas. Model regresi logistik biner kemudian dibangun menggunakan data hasil SMOTE. Kinerja model dievaluasi dengan



Gambar 1. Prinsip teknik pengambilan sampel berlebih minoritas sintetis (SMOTE).

membandingkan hasil sebelum dan sesudah penerapan SMOTE menggunakan tabel kesesuaian klasifikasi dan nilai AUC. Terakhir, model terbaik dipilih untuk penerapan regresi logistik biner.

Teknik SMOTE memiliki beberapa keterbatasan dan potensi bias, yaitu *overfitting*, terutama ketika digunakan tanpa perhatian terhadap pengujian yang ketat [24]. Selain itu, sampel sintetis dihasilkan berdasarkan tetangga terdekat dari kelas minoritas, sehingga model dapat menjadi terlalu sensitif terhadap sampel tersebut, yang mengarah pada kinerja yang sangat baik di data latih, tetapi buruk di data uji. SMOTE juga menghasilkan sampel sintetis secara linier di antara titik data minoritas. Hal ini dapat mengabaikan distribusi kelas sebenarnya, terutama jika kelas minoritas memiliki distribusi yang kompleks atau tersebar. Hasilnya, model dapat menghasilkan keputusan yang kurang akurat ketika berhadapan dengan distribusi data yang tidak teratur.

F. MACHINE LEARNING LIFECYCLE (MLLC)

Siklus hidup pembelajaran mesin (*machine learning lifecyle*, MLLC) adalah serangkaian proses pengembangan model pembelajaran mesin yang dimulai dengan pengumpulan data dan berlanjut hingga model siap untuk digunakan. MLLC berjalan secara progresif dan dapat berulang karena setiap iterasi bertujuan untuk meningkatkan akurasi dan kinerja model secara berkelanjutan [25]. Secara keseluruhan, ada empat aktivitas inti dalam MLLC. Tahap pertama adalah akuisisi data, yaitu proses pengumpulan data yang akan digunakan dalam pengembangan model. Setelah data terkumpul, dilanjutkan ke tahap prapemrosesan data, yang meliputi pembersihan data, seleksi fitur, dan pemisahan data menjadi data latih dan data uji. Tahap berikutnya adalah pelatihan dan evaluasi, yang mencakup pelatihan model menggunakan data latih, evaluasi kinerja menggunakan data uji, serta optimasi model berdasarkan hasil evaluasi. Tahap terakhir adalah *deployment*, yaitu implementasi model yang telah terlatih ke dalam sistem produksi disertai pemantauan kinerja untuk memastikan model berfungsi sesuai kebutuhan [23].

III. METODOLOGI

Metode dalam membandingkan kinerja algoritma KNN dan SVM dalam penelitian ini adalah pendekatan model MLLC. Pendekatan ini dipilih karena mampu memberikan kerangka kerja yang sistematis dalam membangun dan mengevaluasi model pembelajaran mesin, mulai dari pemrosesan data, pelatihan model, hingga evaluasi kinerja.

A. KERANGKA PENELITIAN

Kerangka penelitian yang digunakan dalam penelitian ini mengacu pada pendekatan MLLC, yang terdiri atas beberapa tahapan utama. Pada tahap pertama dilakukan akuisisi data atau pengumpulan data dari berbagai sumber yang relevan, seperti basis data kesehatan atau repositori publik. Pada tahap ini, data mentah dianalisis untuk memahami karakteristiknya, termasuk

jumlah atribut, tipe data, dan distribusi kelas, yang penting untuk langkah selanjutnya.

Setelah pengumpulan data, tahap berikutnya adalah prapemrosesan, yang bertujuan untuk menyiapkan data agar siap digunakan dalam model. Proses ini mencakup pembersihan data untuk menghapus atau memperbaiki data yang hilang dan tidak konsisten, seleksi fitur untuk memilih atribut yang paling relevan, serta *oversampling* dengan teknik SMOTE untuk menangani masalah ketidakseimbangan kelas dalam *dataset*. Selanjutnya, data dibagi menjadi dua set, yaitu data latih dan data uji, untuk memastikan evaluasi model yang objektif.

Pada tahap pelatihan dan evaluasi, algoritma KNN dan SVM diimplementasikan untuk melatih model menggunakan data latih yang telah disiapkan, lalu model yang dihasilkan dievaluasi menggunakan data uji. Evaluasi dilakukan dengan menerapkan *confusion matrix*, yang menghasilkan berbagai metrik evaluasi, termasuk akurasi, presisi, *recall*, dan *F*1-*score*. Melalui tahapan ini, penelitian ini bertujuan untuk memberikan pemahaman yang lebih baik mengenai kinerja algoritma KNN dan SVM dalam klasifikasi penyakit diabetes serta dampak dari penerapan teknik SMOTE terhadap hasil klasifikasi.

B. SUMBER DATA

Dataset yang digunakan dalam penelitian ini adalah data penyakit diabetes yang bersumber dari *kaggle.com*, berjumlah 768 *record* data dengan 9 atribut numerik, yaitu *Pregnancies*, *Glucose*, *Blood Pressure*, *Skin Thickness*, *Insulin*, *body mass index* (BMI), *Diabetes Pedigree Function*, *Age*, dan *Outcome*. Atribut "*Outcome*" menunjukkan pasien menderita diabetes (1) atau tidak (0). Jumlah data untuk kelas 0 (nondiabetes) adalah 500, sedangkan untuk kelas 1 (diabetes) adalah 268, menunjukkan adanya ketidakseimbangan data. Tabel I menyajikan sampel data penyakit diabetes yang ditampilkan dari total 768 *record* data.

IV. HASIL DAN PEMBAHASAN

Hasil penelitian ini akan dijelaskan melalui beberapa tahapan dalam model MLLC. Setiap tahapan, mulai dari pengumpulan data hingga evaluasi model, memiliki peran penting dalam menghasilkan analisis yang akurat. Penjelasan berikut akan merinci setiap langkah dalam proses ini.

A. AKUISISI DATA

Dalam tahap ini, data dikumpulkan dan dianalisis. Analisis ini mencakup pengecekan terhadap keberadaan data duplikat, penanganan nilai yang hilang, distribusi data, deteksi *outlier*, dan eksplorasi korelasi antaratribut. Berikut hasil tahapan akuisisi data dengan melakukan pengumpulan dan analisis data.

1) PENGUMPULAN DATA

Data diperoleh dari platform Kaggle dalam format CSV, terdiri atas 768 data dengan 8 atribut numerik dan 1 label target biner. Atribut tersebut meliputi jumlah kehamilan (*Pregnancies*), kadar glukosa darah dalam mg/dL (*Glucose*), tekanan darah diastolik dalam mmHg (*BloodPressure*), ketebalan kulit dalam mm (*SkinThickness*), kadar insulin dalam mu U/ml (*Insulin*), indeks massa tubuh (BMI), fungsi riwayat diabetes (*DiabetesPedigreeFunction*), dan usia (*Age*), serta label target (*Outcome*) berupa diagnosis diabetes dengan nilai 1 untuk positif dan 0 untuk negatif.

2) ANALISIS DATA

Langkah pertama yang dilakukan pada tahap ini adalah analisis data untuk mengidentifikasi keberadaan data yang

TABEL I
SAMPEL DATA MENTAH PENYAKIT DIABETES

No	*Pregnancies*	*Glucose*	BP	ST	*Insulin*	BMI	DPF	*Age*	*Outcome*
1	6	148	72	35	0	33,6	0,627	50	1
2	1	85	66	29	0	26,6	0,351	31	0
3	8	183	64	0	0	23,3	0,672	32	1
...	...	...	...	...	...	...	...	...	...
768	1	126	60	0	0	30,1	0,349	47	1

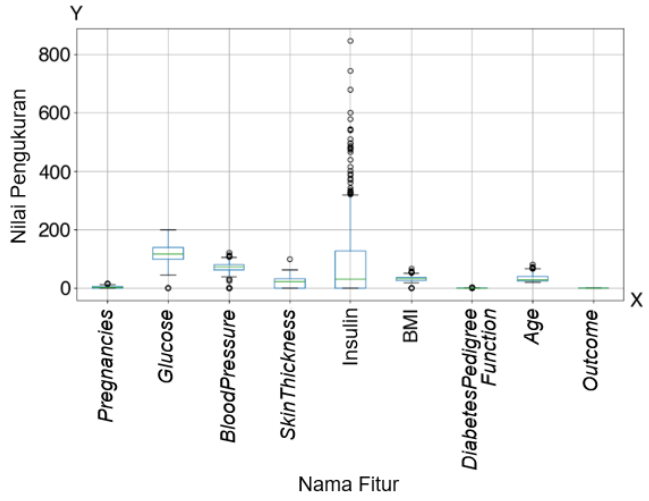
memiliki nilai sama pada beberapa atau seluruh atribut. Keberadaan duplikat dalam *dataset* dapat mengganggu akurasi perhitungan dan analisis, sehingga menghasilkan informasi yang tidak tepat. Hasil analisis menunjukkan bahwa tidak ada data yang memiliki baris duplikat pada setiap atribut dalam *dataset* penyakit diabetes. Hal ini berarti semua data dalam *dataset* bersifat unik, yang dapat mendukung kualitas data dalam proses analisis lebih lanjut serta meminimalkan risiko bias atau informasi yang berlebihan dalam model.

Selanjutnya, analisis terhadap *missing value* dilakukan untuk mengevaluasi jumlah nilai yang hilang pada setiap atribut dalam *dataset* penyakit diabetes untuk memastikan bahwa *dataset* bebas dari nilai kosong atau tidak lengkap yang dapat mengganggu kualitas analisis dan akurasi model prediksi. Dengan memverifikasi setiap atribut, dapat dipastikan bahwa data siap digunakan tanpa adanya ketidaksesuaian atau kehilangan informasi penting. Hasil analisis menunjukkan bahwa tidak ada nilai *null* atau data yang hilang terdeteksi dalam *dataset* diabetes. Hal ini berarti seluruh atribut pada *dataset* memiliki informasi yang lengkap, sehingga tidak diperlukan proses imputasi atau pengisian nilai yang hilang.

Selanjutnya, analisis distribusi kelas dilakukan untuk mengevaluasi proporsi setiap kelas pada label *dataset* penyakit diabetes, yang disebut sebagai *Outcome*. Hasil analisis menunjukkan bahwa distribusi kelas dalam *dataset* tidak seimbang atau *imbalanced*; kelas 0 (nondiabetes) mendominasi dengan 500 data, sedangkan kelas 1 (diabetes) hanya memiliki 268 data. Ketidakseimbangan ini dapat menjadi masalah signifikan dalam pemodelan klasifikasi karena model mungkin lebih cenderung mengklasifikasikan ke kelas mayoritas, yang akhirnya dapat mengurangi akurasi klasifikasi pada kelas minoritas [26]. Untuk menangani ketidakseimbangan ini, SMOTE dapat diterapkan guna menambah data pada kelas minoritas dan memperbaiki kinerja model secara keseluruhan dalam mengklasifikasikan kedua kelas tersebut secara adil.

Gambar 2, yang dihasilkan melalui proses analisis data menggunakan kode pemrograman, menunjukkan informasi tentang *outlier*. Analisis terhadap *outlier* dilakukan untuk mengidentifikasi dan menangani nilai-nilai ekstrem yang secara signifikan berbeda dari mayoritas data dalam dataset. *Outlier* semacam ini dapat mengganggu akurasi prediksi, sehingga memerlukan penanganan yang tepat.

Berdasarkan Gambar 2, terlihat bahwa ada beberapa *outlier* yang signifikan pada berbagai fitur dalam *dataset*. Secara khusus, kolom Insulin menonjol dengan jumlah *outlier* yang sangat banyak dan rentang nilai yang luas dibandingkan dengan kolom lainnya. Fitur seperti *SkinThickness*, *BloodPressure*, *Glucose*, dan BMI juga menunjukkan adanya *outlier* meskipun tidak sebanyak pada kolom Insulin. Sementara itu, kolom *Pregnancies*, *DiabetesPedigreeFunction*, dan *Age* memiliki *outlier* dalam jumlah yang lebih sedikit, tetapi masih signifikan. Tidak terlihat adanya *outlier* pada kolom *Outcome*. Analisis ini menunjukkan perlunya penanganan terhadap nilai-



Gambar 2. Informasi *outlier*.

nilai ekstrem ini, seperti normalisasi untuk mentransformasikan distribusi nilai ekstrem menjadi distribusi yang lebih normal dengan rentang nilai yang sama [27]. Salah satu teknik yang dapat digunakan adalah *MinMax Scaler*, yang membantu dalam menormalkan rentang nilai setiap fitur dalam *dataset*.

Terakhir dilakukan analisis terhadap hubungan antaratribut untuk mengukur tingkat keeratan hubungan dan keterkaitan variabel antaratribut dalam *dataset* penyakit diabetes. Analisis ini menggunakan korelasi Pearson dengan menggunakan fungsi *diabetes.corr()* dari *library* pandas. Visualisasi matriks korelasi ini ditampilkan menggunakan diagram *heatmap* dari *library* Seaborn, yang dapat dilihat pada Gambar 3.

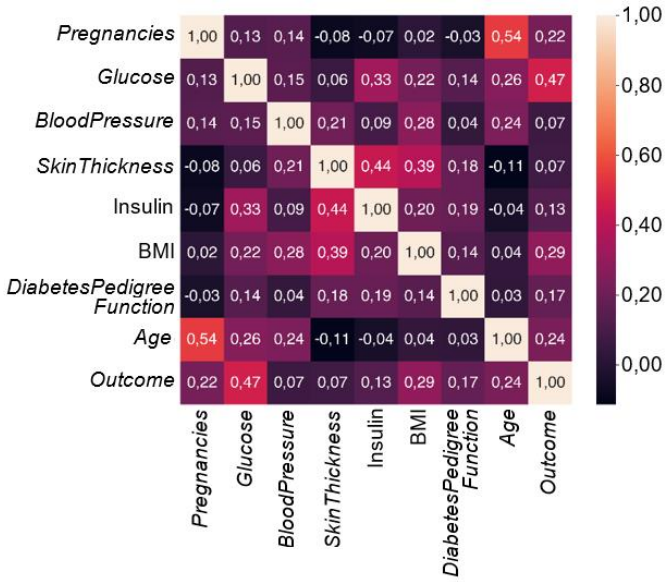
Gambar 3 merupakan hasil visualisasi data yang dihasilkan dari proses analisis menggunakan kode pemrograman. Diagram *heatmap* pada Gambar 3 menunjukkan nilai korelasi antaratribut dalam *dataset*. Nilai korelasi berkisar antara -1 hingga 1. Korelasi yang lebih kuat ditunjukkan oleh warna yang lebih terang, sedangkan korelasi yang lebih lemah ditunjukkan oleh warna yang lebih gelap.

Berdasarkan Gambar 3, terlihat bahwa hubungan yang paling erat antarvariabel terjadi antara suatu variabel dengan variabel itu sendiri. Hal ini ditunjukkan dengan koefisien korelasi Pearson yang bernilai 1, yang menandakan bahwa variabel tersebut memiliki hubungan yang sempurna dengan variabel itu sendiri. Fitur *Age* juga memiliki korelasi positif yang cukup kuat dengan *Pregnancies* (0,54), yang menunjukkan bahwa usia yang lebih tua cenderung berhubungan dengan jumlah kehamilan yang lebih banyak. Selain itu, *Glucose* memiliki korelasi positif yang cukup kuat dengan *Outcome* (0,47), yang menunjukkan bahwa kadar glukosa yang lebih tinggi cenderung berhubungan dengan adanya diabetes. Korelasi antara fitur lainnya cenderung lebih lemah, menunjukkan hubungan yang tidak terlalu signifikan antara fitur-fitur tersebut.

Berdasarkan analisis data yang telah dilakukan, teridentifikasi bahwa *dataset* penyakit diabetes menghadapi dua permasalahan utama, yaitu ketidakseimbangan data (*imbalanced data*) dan keberadaan *outlier*. Untuk mengatasi permasalahan ini, langkah selanjutnya adalah melakukan tahapan prapemrosesan data.

B. PRAPEMROSESAN DATA

Prapemrosesan data dimulai dengan menggunakan *dataset* penyakit diabetes yang telah dianalisis. Tahapan pertama



Gambar 3. Informasi korelasi.

adalah pembersihan data. Pada tahap ini, dilakukan normalisasi data untuk menangani *outlier* menggunakan *MinMax Scaler*. Setelah proses pembersihan data selesai, tahapan kedua melibatkan *balancing data* menggunakan teknik SMOTE untuk mengatasi ketidakseimbangan kelas. Selanjutnya, tahapan ketiga adalah seleksi fitur menggunakan *SelectKBest* dengan skor *f_classif*, disesuaikan dengan kebutuhan klasifikasi penyakit diabetes. Tahap terakhir adalah pemisahan data, yang membagi *dataset* menjadi data latih dan data uji menggunakan *train test split* dengan rasio 70:30. Proses-proses ini serta hasil dari prapemrosesan data dijelaskan sebagai berikut.

1) PENANGANAN *OUTLIER*

Pada tahap pembersihan data, penanganan *outlier* dilakukan untuk memastikan data menjadi bersih, berkualitas, dan siap untuk proses klasifikasi *dataset* penyakit diabetes. Proses penanganan *outlier* melibatkan normalisasi data menggunakan teknik *MinMax Scaler*. Teknik ini mengubah skala setiap fitur secara individual, sehingga nilainya berada dalam rentang tertentu, secara *default* dari 0 hingga 1 [28]. Hasil dari normalisasi data yang dilakukan ditunjukkan pada Tabel II.

Hasil normalisasi yang ditampilkan dalam Tabel II menunjukkan bahwa distribusi data telah diubah menjadi rentang yang konsisten, sehingga data berada dalam skala yang seragam. Hal ini tidak hanya meningkatkan keakuratan model, tetapi juga mengurangi pengaruh nilai ekstrem yang dapat menyebabkan bias dalam proses analisis. Dengan demikian, hasil normalisasi ini membantu menghasilkan data yang lebih bersih, terstruktur, dan siap untuk proses analisis lebih lanjut.

2) SELEKSI FITUR

Pada tahapan seleksi fitur, metode *SelectKBest* dari scikit-learn diterapkan untuk memilih fitur terbaik berdasarkan metode *feature selection univariate*. Metode ini bekerja dengan memilih fitur-fitur terbaik berdasarkan nilai statistik uji *univariate*. Langkah pertama adalah memilih fungsi penilaian untuk mengukur pentingnya setiap fitur dalam memprediksi target. Pada penelitian ini, digunakan fungsi *f_classif*, yang melakukan tes ANOVA F untuk mengukur signifikansi fitur terhadap variabel target. Selanjutnya, objek *SelectKBest* diinisialisasi dengan fungsi penilaian yang dipilih dan

TABEL II
HASIL NORMALISASI DATA

No	*Glucose*	*BloodPressure*	*SkinThickness*	..	*Insulin*
1	0,743719	0,590164	0353535	..	0,000000
2	0,427136	0,540984	0,292929	..	0,000000
3	0,919598	0,524590	0,000000	..	0,000000
...	....	....	....	..	....
768	0,688442	0,327869	0,353535	..	0,198582

parameter k = 5, yang berarti lima fitur terbaik akan dipilih untuk digunakan dalam proses selanjutnya.

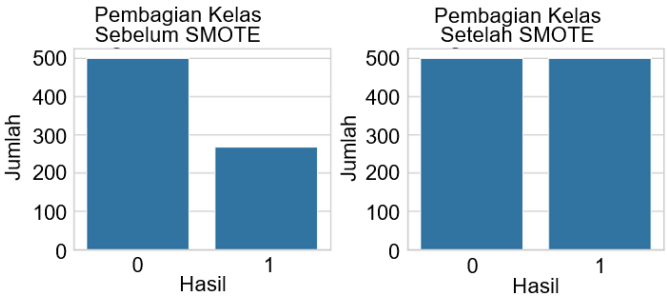
Setelah itu, metode *fit_transform* diterapkan pada data fitur (x) dan target (y). Metode ini menghitung skor untuk setiap fitur berdasarkan fungsi penilaian yang dipilih, dalam hal ini menggunakan *f_classif* yang melibatkan tes ANOVA F. Proses ini kemudian mengubah data fitur dengan mempertahankan lima fitur dengan skor tertinggi sesuai dengan parameter *k* yang telah ditentukan sebelumnya. Hasil dari proses seleksi fitur ini menunjukkan bahwa lima fitur terbaik yang dipilih adalah *Pregnancies*, *Glucose*, BMI, *DiabetesPedigreeFunction*, dan *Age*. Pemilihan fitur ini mengindikasikan bahwa fitur-fitur tersebut memiliki kontribusi yang paling signifikan dalam memprediksi hasil atau target dalam *dataset* diabetes.

3) *BALANCING DATA*

Proses *balancing data* pada *dataset* penyakit diabetes dilakukan dengan menerapkan teknik SMOTE. Pertama, jumlah sampel untuk setiap kelas dalam variabel target dihitung menggunakan fungsi *value_counts()*, yang menghasilkan jumlah sampel untuk kelas mayoritas dan minoritas. Kemudian, *sampling strategy* ditentukan dengan menetapkan jumlah sampel yang diinginkan untuk setiap kelas; kelas minoritas akan disamakan jumlahnya dengan kelas mayoritas. Selanjutnya, objek SMOTE diinisialisasi dengan *sampling strategy* yang telah ditentukan dan *random state* untuk memastikan hasil yang konsisten. Terakhir, SMOTE diterapkan ke seluruh *dataset* dengan menggunakan metode *fit resample*, yang menghasilkan data fitur (*X_resampled*) dan target (*y_resampled*) yang telah di-*oversample*, sehingga setiap kelas memiliki jumlah sampel yang sama. Hasil dari penerapan teknik SMOTE pada jumlah data kelas yang *imbalanced* pada *dataset* penyakit diabetes ditunjukkan pada Gambar 4.

Gambar 4 merupakan hasil visualisasi data yang dihasilkan melalui proses analisis menggunakan kode pemrograman. Diagram ini terdiri atas dua *subplot* yang menggambarkan distribusi kelas dalam *dataset* diabetes sebelum dan sesudah penerapan teknik SMOTE. Pada *subplot* pertama, yang berada di sisi kiri, ditampilkan distribusi kelas sebelum penerapan SMOTE. Di sumbu horizontal (sumbu x) terdapat dua kategori hasil, yaitu 0 (tidak diabetes) dan 1 (diabetes). Sumbu vertikal (sumbu y) menunjukkan jumlah kasus untuk masing-masing kategori. Pada *subplot* kedua, yang berada di sisi kanan, ditampilkan distribusi kelas setelah penerapan SMOTE untuk variabel *y_resampled*, yang merupakan *dataset* setelah teknik SMOTE diterapkan.

Setelah SMOTE diterapkan, seperti yang terlihat pada *subplot* di sisi kanan, distribusi kelas menunjukkan perubahan yang signifikan. SMOTE menambah sampel pada kelas minoritas, yaitu kelas 1, hingga bertambah sebanyak 232 sampel. Dengan demikian, jumlah sampel pada kedua kelas menjadi seimbang, masing-masing memiliki 500 sampel data. Hal ini memungkinkan model untuk menganalisis kedua kelas



Gambar 4. Penerapan teknik SMOTE.

secara lebih adil, sehingga mengurangi potensi bias dalam proses pelatihan dan meningkatkan akurasi model untuk kelas minoritas.

4) PEMISAHAN DATA

Proses terakhir dalam tahapan prapemrosesan data adalah membagi data menjadi data latih dan data uji. Pembagian data ini dilakukan menggunakan fungsi *train_test_split* dari scikit-learn. Pertama, data fitur dan target yang telah di-*resampling* menggunakan SMOTE dibagi menjadi dua bagian, satu untuk melatih model dan satu lagi untuk menguji kinerja model. Dengan menetapkan parameter *test_size* sebesar 0,3, 30% dari data akan digunakan sebagai set pengujian, sedangkan 70% sisanya digunakan untuk pelatihan model. Selain itu, parameter *random_state* sebesar 21 digunakan untuk memastikan bahwa pembagian data ini konsisten setiap kali kode dijalankan, yang penting untuk memastikan reprodusibilitas hasil eksperimen.

Hasil pembagian data menunjukkan bahwa set pelatihan *x_train* terdiri atas 700 sampel, masing-masing dengan 5 fitur, yang berarti ada 700 baris dan 5 kolom dalam data fitur pelatihan. Selain itu, *y_train* memiliki 700 nilai target yang sesuai dengan setiap sampel dalam *x_train*. Kemudian, set pengujian *x_test* terdiri atas 300 sampel dengan 5 fitur masing-masing, sehingga ada 300 baris dan 5 kolom dalam data fitur pengujian. Sementara itu, *y_test* memiliki 300 nilai target yang sesuai dengan setiap sampel dalam *x_test*. Pembagian ini konsisten dengan parameter *test size*, yaitu 0,3, yang membagi 30% dari data untuk pengujian dan 70% untuk pelatihan.

C. PELATIHAN DAN EVALUASI MODEL

Pada tahapan pelatihan dan evaluasi model, dilakukan perbandingan kinerja algoritma SVM. Selain itu, algoritma KNN juga dievaluasi untuk mengetahui kinerjanya.

Tahapan ini dimulai dengan membagi data menjadi data latih dan data uji dengan rasio 70:30 pada *dataset* hasil prapemrosesan. Data latih digunakan untuk melatih dua model utama. Pertama, digunakan algoritma SVM dengan *kernel* RBF. Evaluasi dilakukan untuk mencari hasil terbaik dari SVM dengan memeriksa akurasi pada data latih dan data uji. Selanjutnya, eksplorasi dilakukan untuk menemukan nilai *k* terbaik pada model KNN. Proses ini melibatkan *cross validation* dengan 10 *fold* untuk mengevaluasi berbagai nilai *k* dari 1 hingga 30. Setelah menemukan nilai *k* terbaik, dilakukan *plotting* untuk menampilkan akurasi *cross validation* terhadap nilai *k* yang berbeda. Model KNN kemudian dilatih ulang menggunakan nilai *k* terbaik yang telah ditentukan dari data latih, serta diuji terhadap data uji. Terakhir, hasil dari kedua model diuji dan dievaluasi menggunakan metrik dari *confusion matrix*, yaitu akurasi, presisi, *recall*, dan *F*1-*score*. Penjelasan mendalam mengenai pelatihan dan evaluasi masing-masing model akan disajikan berikutnya.

1) ALGORITMA *SUPPORT VECTOR MACHINE*

Model pertama yang diterapkan dalam penelitian ini adalah SVM dengan kernel RBF, yang digunakan untuk memprediksi kemungkinan seseorang mengidap penyakit diabetes. Implementasi model SVM ini menggunakan prinsip dasar *hyperplane* seperti pada (1).

Tahap pelatihan dimulai dengan inisialisasi model menggunakan parameter *svc = SVC(kernel='rbf', C=1.0, gamma='scale')*. Model ini kemudian mencari *hyperplane* optimal yang memenuhi kondisi pada (2), yaitu $y_i(w^T x + b) \geq 1$, dengan $y_i = +1$ untuk kelas diabetes dan $y_i = -1$ untuk kelas nondiabetes.

Selama proses pelatihan menggunakan *svc.fit(X_train, y_train)*, model berusaha memaksimalkan margin antara dua kelas. Hal ini dilakukan dengan meminimalkan $||w||$ sambil tetap memenuhi kekangan pada (2).

Evaluasi model dilakukan pada data uji, yaitu 30% dari keseluruhan *dataset*, untuk menilai akurasi model di luar data latih. Hasil evaluasi menunjukkan kinerja model dalam mendeteksi diabetes. Hasil ini memberikan gambaran mendetail tentang efektivitas pendekatan SVM dengan *kernel* RBF dalam penelitian ini.

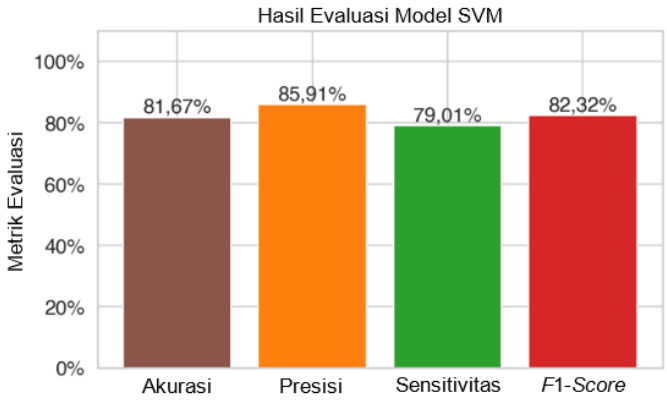
Gambar 5 menunjukkan hasil visualisasi proses evaluasi model menggunakan algoritma SVM dengan *kernel* RBF. Visualisasi ini dihasilkan melalui analisis menggunakan kode pemrograman. Hasil evaluasi pada Gambar 5 menunjukkan bahwa model SVM dengan *kernel* RBF berhasil mencapai kinerja yang cukup baik dalam memprediksi penyakit diabetes. Model ini memperoleh nilai akurasi sebesar 81,67%, yang mengindikasikan bahwa 81,67% dari total prediksi yang dilakukan oleh model adalah benar. Selain itu, presisi model tercatat sebesar 85,91%, yang menandakan kemampuan model untuk secara tepat mengidentifikasi kasus diabetes di antara semua prediksi positif yang dibuat. Nilai *recall* sebesar 79,01% menunjukkan bahwa model cukup baik dalam mendeteksi kasus positif diabetes dari seluruh kasus sebenarnya. *F*1-*score*, yang merupakan rata-rata harmonis dari presisi dan *recall*, berada di angka 82,32%, mencerminkan keseimbangan antara presisi dan *recall* yang dicapai oleh model ini.

Untuk melihat kesalahan prediksi yang dilakukan oleh model, evaluasi dilanjutkan dengan menganalisis *confusion matrix* yang disajikan pada Gambar 6. Analisis ini menunjukkan jumlah prediksi benar dan salah pada masing-masing kelas.

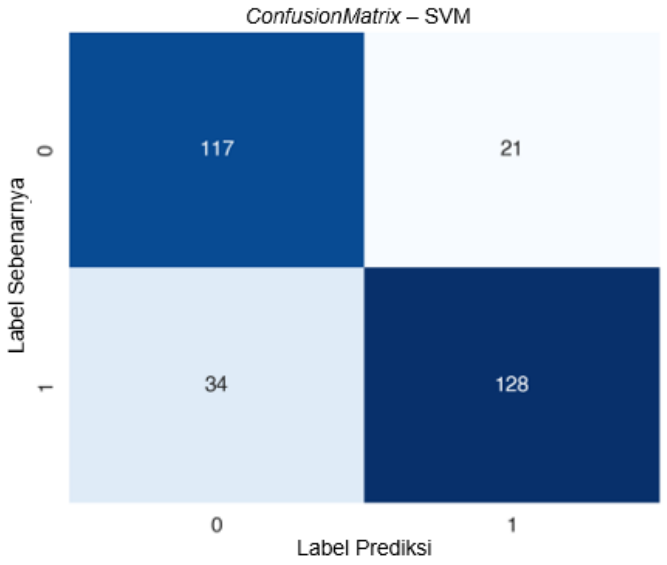
Hasil visualisasi analisis kesalahan prediksi menggunakan *confusion matrix* pada Gambar 6 diperoleh melalui proses analisis menggunakan kode pemrograman. Berdasarkan hasil evaluasi pada gambar tersebut. menggunakan *confusion matrix*, jumlah data yang digunakan untuk pengujian adalah sekitar 230 data. Dari data tersebut, model SVM RBF mampu memprediksi dengan benar 128 pasien terkena diabetes dan 117 pasien tidak terkena diabetes. Namun, terdapat 21 sampel yang sebenarnya termasuk kelas 0 (tidak terkena diabetes), tetapi diprediksi sebagai kelas 1 (terkena diabetes) oleh model; serta 34 sampel yang sebenarnya termasuk kelas 1 (terkena diabetes), tetapi diprediksi sebagai kelas 0 (tidak terkena diabetes) oleh model.

2) ALGORITMA *K-NEAREST NEIGHBOR*

Penerapan model kedua untuk klasifikasi penyakit diabetes menggunakan algoritma KNN dilakukan dengan bantuan modul *KNeighborsClassifier* dari *library* sklearn. Pada tahap ini, nilai *k* yang optimal dicari menggunakan metode *k-fold cross-validation*, yang membagi data secara bergantian sebagai



Gambar 5. Hasil evaluasi algoritma SVM RBF.



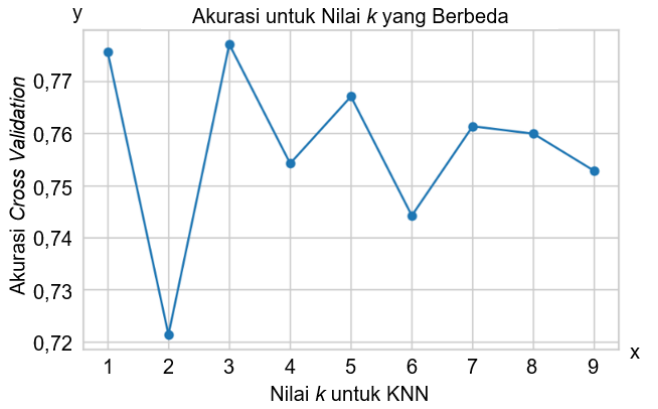
Gambar 6. *Confusion matrix* algoritma SVM RBF.

data latih dan data uji dalam 10 *fold*. Dalam proses ini, nilai *k* divariasikan dari 1 hingga 10 untuk menemukan pengaturan terbaik bagi model KNN.

Setiap nilai *k* dievaluasi melalui sepuluh iterasi dan skor akurasi rata-rata dihitung untuk setiap iterasi. Skor rata-rata ini memberikan gambaran tentang stabilitas kinerja model pada setiap pengaturan *k*. Hasil evaluasi ini digunakan untuk memilih nilai *k* dengan skor akurasi tertinggi, yang ditampilkan pada Gambar 7. Grafik tersebut menunjukkan perubahan kinerja KNN di setiap nilai *k* dan membantu menentukan parameter terbaik agar model dapat menghasilkan prediksi yang paling akurat.

Hasil visualisasi pada Gambar 7 dihasilkan dari proses analisis menggunakan kode pemrograman. Pada grafik, sumbu horizontal (sumbu x) menunjukkan nilai *k* yang digunakan dalam model KNN, sedangkan sumbu vertikal (sumbu y) menampilkan akurasi rata-rata dari *cross-validation* untuk setiap nilai. Garis biru pada grafik menggambarkan akurasi rata-rata tersebut, dengan titik-titik pada garis menunjukkan akurasi untuk masing-masing nilai *k* yang diuji. Hasil dari pencarian nilai *K* terbaik ditunjukkan pada Tabel III.

Berdasarkan Tabel III, nilai *K* terbaik ditemukan pada *K* = 1 dan *K* = 3, dengan nilai akurasi hasil *cross-validation* mencapai 0,78. Setelah menentukan nilai *K* yang optimal, langkah selanjutnya adalah membangun model KNN menggunakan data latih yang telah disiapkan sebelumnya. Model ini kemudian diuji dengan data uji (*X_test*) untuk melakukan prediksi kelas berdasarkan fitur yang ada. Setelah



Gambar 7. Penentuan nilai *K* terbaik.

TABEL III
AKURASI SETIAP PERCOBAAN NILAI *K*

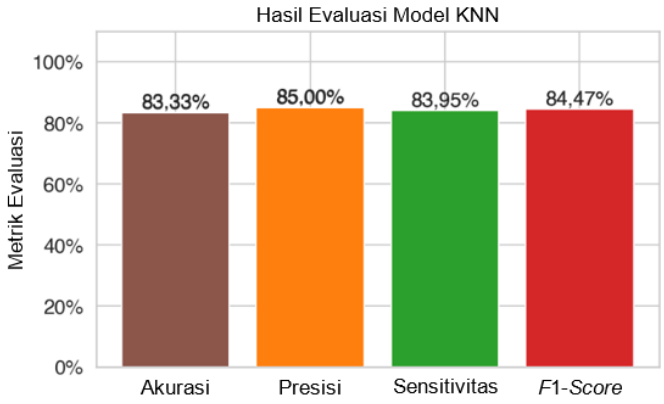
Nilai *K*	Akurasi
K=1	0,78
K=2	0,72
K=3	0,78
K=4	0,75
K=5	0,77
K=6	0,74
K=7	0,76
K=8	0,76
K=9	0,75
K=10	0,75

proses prediksi, hasil yang diperoleh dibandingkan dengan nilai sebenarnya dari data uji (*y_test*) untuk menghitung akurasi pengujian model. Selain itu, evaluasi akurasi juga dilakukan pada 30% data latih untuk memastikan bahwa model tidak hanya baik dalam mengklasifikasikan data yang baru, tetapi juga data yang telah dilatih.

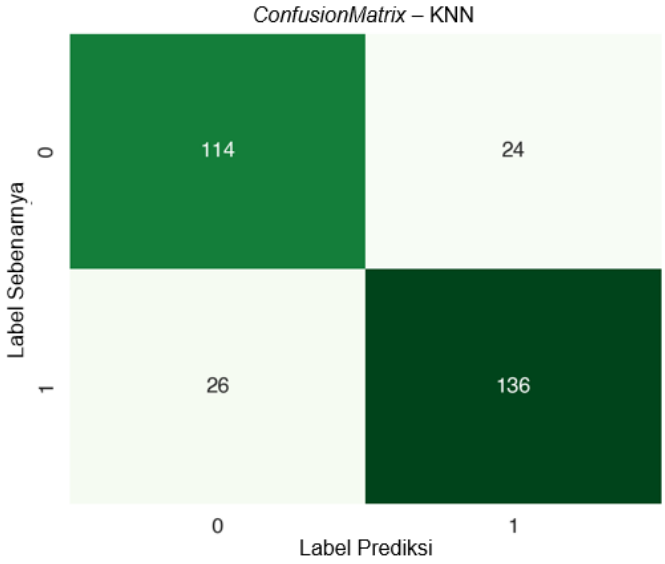
Untuk mendapatkan gambaran yang lebih komprehensif mengenai kinerja model, evaluasi akhir dilakukan dengan menggunakan *confusion matrix*. Hasil evaluasi dari model KNN ditunjukkan pada Gambar 8. yang memberikan pemahaman mengenai efektivitas model dalam mengidentifikasi penyakit diabetes berdasarkan *dataset* yang digunakan.

Gambar 8 merupakan hasil visualisasi dari analisis akhir menggunakan *confusion matrix* yang diperoleh melalui proses analisis menggunakan kode pemrograman. Hasil evaluasi model KNN menunjukkan kinerja yang cukup baik dalam klasifikasi penyakit diabetes. Model ini berhasil mencapai nilai akurasi sebesar 83,33%, yang menunjukkan persentase data uji yang berhasil diprediksi dengan benar oleh model. Selain itu, nilai presisi yang diperoleh adalah 85,00%, mengindikasikan proporsi prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dibuat oleh model. Nilai *recall* yang mencapai 83,95% menggambarkan kemampuan model dalam mengidentifikasi kasus positif (diabetes) secara tepat, sehingga model dapat diandalkan dalam mendeteksi kondisi ini. Terakhir, nilai *F1-score* sebesar 84,47% mencerminkan keseimbangan antara presisi dan *recall*, menunjukkan bahwa model tidak hanya fokus pada satu aspek, tetapi berusaha untuk mencapai hasil yang seimbang dalam klasifikasi.

Untuk melihat kesalahan prediksi yang dilakukan oleh model, evaluasi dilanjutkan dengan menganalisis *confusion matrix* yang disajikan pada Gambar 9. Analisis ini menunjukkan jumlah prediksi benar dan salah pada masing-masing kelas.



Gambar 8. Hasil evaluasi algoritma KNN.



Gambar 9. *Confusion matrix* algoritma KNN.

TABEL IV
PERSENTASE PERBANDINGAN HASIL

Metrik	SVM Sebelum SMOTE	KNN Sebelum SMOTE	SVM Sesudah SMOTE	KNN Sesudah SMOTE
Akurasi	69,70	70,13	81,67	83,33
Presisi	71,79	71,43	85,91	85,00
Recall	32,18	34,48	79,01	83,95
F1-score	44,44	46,51	82,32	84,47

Hasil visualisasi analisis kesalahan prediksi menggunakan *confusion matrix* pada Gambar 9 diperoleh melalui proses pengolahan data menggunakan kode pemrograman untuk mengevaluasi kinerja klasifikasi model. Berdasarkan hasil evaluasi kesalahan hasil prediksi model KNN pada Gambar 9. menggunakan *confusion matrix*, jumlah data yang digunakan untuk pelatihan berjumlah kurang lebih 230 data. Dari data tersebut, model KNN mampu memprediksi 136 pasien terkena penyakit diabetes dan 114 pasien tidak terkena penyakit diabetes. Namun, terdapat 24 sampel yang sebenarnya termasuk kelas 0 (tidak terkena diabetes), tetapi diprediksi sebagai kelas 1 (terkena diabetes) oleh model dan 26 sampel yang sebenarnya termasuk kelas 1 (terkena diabetes), tetapi diprediksi sebagai kelas 0 (tidak terkena diabetes) oleh model.

3) PERBANDINGAN HASIL EVALUASI

Beberapa metrik evaluasi, seperti akurasi, presisi, *recall*, dan *F1-score*, digunakan untuk mengukur efektivitas model SVM dan KNN, baik sebelum maupun setelah penerapan

teknik SMOTE. Hasil perbandingan ini ditampilkan dalam Tabel IV. Tampak bahwa model KNN (k = 3) menunjukkan kinerja yang lebih baik dibandingkan dengan model SVM (*kernel* RBF). Perbandingan ini terlihat dalam hal akurasi, presisi, *recall*, dan *F*1-*score*.

V. KESIMPULAN

Berdasarkan hasil yang diperoleh dan pembahasan yang dilakukan, dapat disimpulkan bahwa dalam perbandingan antara algoritma SVM dengan *kernel* RBF dan KNN dengan nilai *k* terbaik, yaitu 3, menunjukkan kinerja yang lebih unggul. KNN mencapai akurasi sebesar 83,33%, presisi 85,00%, *recall* 83,95%, dan *F*1-*score* 84,47%, yang semuanya lebih tinggi dibandingkan dengan SVM, yang mencapai akurasi 81,67%, presisi 85,91%, *recall* 79,01%, dan *F*1-*score* 82,32%. Hal ini menunjukkan bahwa KNN lebih efektif dalam klasifikasi penyakit diabetes pada *dataset* ini dibandingkan dengan SVM. KNN mengungguli SVM karena algoritma ini lebih cocok untuk *dataset* sederhana dengan distribusi kelas yang jelas. Metode KNN membuat prediksi berdasarkan tetangga terdekat, yang sangat efektif pada *dataset* terbatas seperti ini. Sebaliknya, SVM lebih sesuai untuk *dataset* dengan pola yang lebih kompleks dan sering memerlukan penyesuaian parameter yang lebih mendalam. Selain itu, KNN cenderung lebih tahan terhadap data yang tidak konsisten, sedangkan SVM dapat dipengaruhi oleh variasi data tersebut. Terakhir, penggunaan teknik SMOTE terbukti efektif dalam mengatasi ketidakseimbangan kelas pada *dataset* ini, dengan meningkatkan kinerja model klasifikasi secara keseluruhan.

KONFLIK KEPENTINGAN

Penulis menyatakan tidak terdapat konflik kepentingan.

KONTRIBUSI PENULIS

Konseptualisasi, Sarah Khoerunisa; metodologi, Sarah Khoerunisa; perangkat lunak, Sarah Khoerunisa; validasi, Asri Mulyani; penulisan—penyusunan draf asli, Sarah Khoerunisa; penulisan—peninjauan dan penyuntingan, Asri Mulyani, Sarah Khoerunisa, dan Dede Kurniadi; visualisasi, Sarah Khoerunisa; pengawasan, Asri Mulyani dan Dede Kurniadi; pendanaan, Asri Mulyani dan Dede Kurniadi (Institut Teknologi Garut).

UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya disampaikan kepada Institut Teknologi Garut atas dukungan dan pendanaan yang diberikan dalam penelitian ini. Tanpa bantuan dan komitmen yang kuat dari pihak tersebut, penelitian ini tidak akan dapat terlaksana dengan baik. Selain itu, ucapan terima kasih juga disampaikan kepada semua pihak yang telah berkontribusi, baik secara langsung maupun tidak langsung, dalam proses penelitian ini.

REFERENSI

[1] F.M. Hana, "Klasifikasi penderita penyakit diabetes menggunakan algoritma decision tree C4.5," *J. Sist. Komput. Kecerdasan Buatan*, vol. IV, no. 1, hal. 32–39, Sep. 2020, doi: 10.47970/siskom-kb.v4i1.173.

[2] IDF, "International Diabetes Federation," 2024. https://idf.org/ (accessed Mar. 13, 2024).

[3] World Health Organization, "Diabetes type 1 and type 2 Causes of diabetes," 2024. https://www.who.int/health-topics/diabetes?gad_source=1&gclid=Cj0KCQjw-r-vBhC-ARIsAGgUO2ATe6b9pbM8tg01IGtkszHXAxW4PvDAnxhK_9-YhqlJNnhkLdVFKHgaAguwEALw_wcB#tab=tab_1 (accessed Mar. 13, 2024).

[4] Gunawan dkk., "Penerapan linear sampling dan information gain pada algoritma decision tree untuk diagnosis penyakit diabetes," *Multinetics*, vol. 7, no. 1, hal. 124–131, Nov. 2021, doi: 10.32722/multinetics.v7i2.3796.

[5] K.R. Widiasari, I.M.K. Wijaya, dan P.A. Suputra, "Diabetes melitus tipe 2: Faktor risiko, diagnosis, dan tatalaksana," *Ganesha Med. J.*, vol. 1, no. 2, hal. 114-120, Sep. 2021, doi: 10.23887/gm.v1i2.40006.

[6] N.M. Putry dan B.N. Sari, "Komparasi algoritma KNN dan naïve Bayes untuk klasifikasi diagnosis penyakit diabetes mellitus," *EVOLUSI J. Sains Manaj.*, vol. 10, no. 1, hal. 45-57, Sep. 2022, doi: 10.31294/evolusi.v10i1.12514.

[7] N.W. Mardiyyah, N. Rahaningsih, dan I. Ali, "Penerapan data mining menggunakan algoritma k-nearest neighbor pada prediksi pemberian kredit di sektor finansial," *JATI (J. Mhs. Tek. Inform.)*, vol. 8, no. 2, hal. 1491–1499, Apr. 2024, doi: 10.36040/jati.v8i2.9010.

[8] J.A. Wibowo, V.C. Mawardi, dan T. Sutrisno, "Penerapan support vector machine untuk analisis sentimen fitur layanan pada ulasan Gojek," *J. Ilmu Komput. Sist. Inf.*, vol. 12, no. 1, hal. 1–8, Jan. 2024, doi: 10.24912/jiksi.v12i1.28211.

[9] N.K. Sowabi, N.A. Widiastuti, dan N.A. Maori, "Optimasi algoritma k-nearest neighbors menggunakan teknik Bayesian optimization untuk klasifikasi diabetes," *J. Inf. Syst. Res. (JOSH)*, vol. 6, no. 1, pp. 294–301, Okt. 2024, doi: 10.47065/josh.v6i1.5975.

[10] A.M. Argina, "Penerapan metode klasifikasi k-nearest neigbor pada dataset penderita penyakit diabetes," *Indonesian J.Data Sci.*, vol. 1, no. 2, hal. 29–33, Jul. 2020, doi: 10.33096/ijodas.v1i2.11.

[11] A.W. Mucholladin, F.A. Bachtiar, dan M.T. Furqon, "Klasifikasi penyakit diabetes menggunakan metode support vector machine," *J. Pengemb. Teknol. Inf. Ilmu Komput.*, vol. 5, no. 2, hal. 622–633, Feb. 2021.

[12] H.S.W. Hovi, A. Id Hadiana, dan F.R. Umbara, "Prediksi penyakit diabetes menggunakan algoritma support vector machine (SVM)," *Inform. Digit. Expert (INDEX)*, vol. 4, no. 1, hal. 40–45, Mei 2022, doi: 10.36423/index.v4i1.895.

[13] H.A.D. Fasnuari, H. Yuana, dan M.T. Chulkamdi, "Penerapan algoritma k-nearest neighbor (K-NN) untuk klasifikasi penyakit diabetes melitus Studi kasus : Warga Desa Jatitengah," *ANTIVIRUS: J. Ilm. Tek. Inform.*, vol. 16, no. 2, hal. 133–142, Nov. 2022, doi: 10.35457/antivirus.v16i2.2445.

[14] D. Kurniadi, F. Nuraeni, dan M. Firmansyah, "Klasifikasi masyarakat penerima bantuan langsung tunai dana desa menggunakan naïve Bayes dan SMOTE," *J. Teknol. Inf. Ilmu Komput.*, vol. 10, no. 2, hal. 309–320, Apr. 2023, doi: 10.25126/jtiik.20231026453.

[15] I.D.A.E.C. Astutisari, A.A.A.Y. Darmini, dan I.A.P. Wulandari, "Hubungan pola makan dan aktivitas fisik dengan kadar gula darah pada pasien diabetes melitus tipe 2 di Puskesmas Manggis I," *J. Ris. Kesehat. Nas.*, vol. 6, no. 2, hal. 79–87, Okt. 2022, doi: 10.37294/jrkn.v6i2.350.

[16] R. Kosasih, "Klasifikasi tingkat kematangan pisang berdasarkan ekstraksi fitur tekstur dan algoritme KNN," *J. Nas. Tek. Elekt. Teknol. Inf.*, vol. 10, no. 4, hal. 383–388, Nov. 2021, doi: 10.22146/jnteti.v10i4.462.

[17] F.A. Tyas, M. Nurayuni, dan H. Rakhmawati, "Optimasi algoritma k-nearest neighbors berdasarkan perbandingan analisis outlier (berbasis jarak, kepadatan, LOF)," *J. Nas. Tek. Elekt. Teknol. Inf.*, vol. 13, no. 2, hal. 108–115, Mei 2024, doi: 10.22146/jnteti.v13i2.9579.

[18] N. Ikhwana, M. Nusrang, dan Sudarmin, "Perbandingan metode PCA-SVM dan SVM untuk klasifikasi indeks kepuasan masyarakat terhadap layanan pendidikan di Kabupaten Jeneponto," *VARIANSI: J. Stat. Its Appl. Teach. Res.*, vol. 3, no. 3, hal. 148–155, 2021, doi: 10.35580/variansiunm22988.

[19] L.U. Khasanah, Y.N. Nasution, dan F.D.T. Amijaya, "Klasifikasi penyakit diabetes melitus menggunakan algoritma naïve Bayes classifier," *JUSTINDO (J. Sist. Teknol. Inf. Indones.*, vol. 7, no. 1, hal. 59–66, Feb. 2022, doi: 10.32528/justindo.v7i1.4949.

[20] N. Saputra dkk., "Improving foreign language proficiency in society by decision tree classification," *AIP Conf. Proc.*, vol. 3001, no. 1, Feb. 2024, doi: 10.1063/5.0183888.

[21] F.S. Pamungkas, B.D. Prasetya, dan I. Kharisudin, "Perbandingan metode klasifikasi supervised learning pada data bank customers menggunakan Python," *PRISMA: Pros. Semin. Nas. Mat.*, vol. 3, hal. 692–697, 2020.

[22] K. Akbar dan M. Hayaty, "Data balancing untuk mengatasi imbalance dataset pada prediksi produksi padi," *J. Ilm. Intech: Inf. Technol. J. UMUS*, vol. 2, no. 2, hal. 1–14, Nov. 2020, doi: 10.46772/intech.v2i02.283.

[23] J. Chen dkk., "Machine learning-based classification of rock discontinuity trace: SMOTE oversampling integrated with GBT

ensemble learning," *Int. J. Min. Sci. Technol.*, vol. 32, no. 2, hal. 309–322, Mar. 2022, doi: 10.1016/j.ijmst.2021.08.004.

[24] Karfindo, R. Turaina, dan R. Saputra, "Optimalisasi klasifikasi umpan balik mahasiswa terhadap layanan kampus dengan sinergi random forest dan Smote," *J. Nas. Komput.Teknol. Inf.*, vol. 6, no. 6, hal. 820–827, Des. 2023, doi: 10.32672/jnkti.v6i6.7269.

[25] I. Daqiqil, *Machine Learning: Teori, Studi Kasus dan Implementasi Menggunakan Phyton*, 1st ed. Pekanbaru, Indonesia: UR PRESS, 2021.

[26] R.D. Fitriani, H. Yasin, dan Tarno, "Penanganan klasifikasi kelas data tidak seimbang dengan random oversampling pada naive Bayes (Studi Kasus: Status Peserta KB IUD di Kabupaten Kendal)," *J. Gaussian*, vol. 10, no. 1, hal. 11–20, Feb. 2021, doi: 10.14710/j.gauss.v10i1.30243.

[27] M.R. Kusnaidi, T. Gulo, dan S. Aripin, "Penerapan normalisasi data dalam mengelompokkan data mahasiswa dengan menggunakan metode k-means untuk menentukan prioritas bantuan uang kuliah tunggal," *J. Comput. Syst. Inform.*, vol. 3, no. 4, hal. 330–338, Agu. 2022, doi: 10.47065/josyc.v3i4.2112.

[28] M.F. Naufal dkk., "Analisis perbandingan algoritma machine learning untuk prediksi potensi hilangnya nasabah bank," *Techno.COM*, vol. 22, no. 1, hal. 1–11, Feb. 2023, doi: 10.33633/tc.v22i1.7302.