Journal of System & Technology
ISSN: 3110-0171
Vol. 2, No. 1, June 2026

PERBANDINGAN ALGORITMA RANDOM FOREST, SVM, KNN, DAN DECISION TREE UNTUK PREDIKSI PENYAKIT STROK

Christian Gideon Valent[1], Jeff Clarence[2], Raissa Nabila Putri Riski[3], Dian Ramadhani[4]*
[1]Dept.of Informatics Engineering, Faculty of Engineering, Universitas Riau, Pekanbaru, Indonesia
christian.gideon6140@student.unri.ac.id, jeffclarence@student.unri.ac.id, raissa.nabila2016@student.unri.ac.id, dianramadhani@lecturer.unri.ac.id
*Corresponding author: dianramadhani@lecturer.unri.ac.id

Abstract: Strok merupakan salah satu penyebab kematian tertinggi di dunia, sehingga deteksi dini sangat penting untuk mengurangi risiko fatalitas. Namun, data klinis terkait penyakit ini umumnya mengalami ketidakseimbangan kelas (*class imbalance*) yang ekstrem, yang dapat memengaruhi keandalan model prediktif. Penelitian ini membandingkan performa empat algoritma *machine learning*, yaitu Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Decision Tree untuk memprediksi risiko strok menggunakan *Healthcare Strok Dataset* dari Kaggle yang terdiri atas 5.110 data pasien berdasarkan 11 fitur prediktor (proporsi kelas tidak strok 95,13% dan strok 4,87%). Untuk mengatasi ketidakseimbangan data, metode *Synthetic Minority Over-sampling Technique* (SMOTE) diterapkan pada data pelatihan, diikuti dengan normalisasi *StandardScaler* dan pengujian validitas melalui *5-fold stratified cross-validation*. Hasil penelitian menunjukkan bahwa Random Forest memperoleh akurasi keseluruhan tertinggi sebesar 93,05%, disusul oleh Decision Tree (90,02%), KNN (79,94%), dan SVM (75,44%). Sebaliknya, SVM menunjukkan kemampuan terbaik dalam mengidentifikasi kasus positif dengan nilai *Recall* kelas strok tertinggi sebesar 60,00%, dibandingkan KNN (36,00%), Decision Tree (22,00%), dan Random Forest (16,00%). Hasil ini menunjukkan adanya *trade-off* yang jelas antara akurasi global dan sensitivitas deteksi kelas minoritas. Dalam konteks diagnosis medis, meminimalkan kegagalan deteksi pasien sakit (*False Negative*) dinilai lebih krusial dibandingkan mengejar akurasi keseluruhan, sehingga model SVM dengan integrasi SMOTE direkomendasikan sebagai pendekatan yang paling sesuai untuk kebutuhan prediksi klinis strok.

Keywords—Klasifikasi, Machine learning, Prediksi Strok, Random Forest, SMOTE, Support Vector Machine

Article Information:
Received: 10-Jun-2026; Revised: 22-Jun-2026; Accepted: 28-Jun-2026; Published: 1-Jul-2026

This work is licensed under a **CC BY-SA**. Copyright ©2026 by Author. Published by Universitas Riau.

I. PENDAHULUAN

Strok merupakan salah satu penyebab utama kematian dan kecacatan di dunia [1]. Menurut World Health Organization (WHO) [1], pada tahun 2021 strok menjadi penyebab kematian dan kecacatan terbesar ketiga di dunia dengan estimasi 93,8 juta kasus serta 11,9 juta kasus baru setiap tahunnya. Penyakit ini terjadi akibat terganggunya aliran darah menuju otak yang dapat memicu kerusakan otak, kecacatan permanen, hingga kematian. Di Indonesia, strok berkontribusi sebesar 11,2% terhadap total kecacatan dan 18,5% terhadap total kematian. Berdasarkan data Survei Kesehatan Indonesia tahun 2023, prevalensi strok di Indonesia mencapai 8,3 per 1.000 penduduk [2]. Tingginya angka ini menuntut upaya deteksi dini dan pencegahan yang efektif.

Perkembangan teknologi kecerdasan buatan (*Artificial Intelligence*/AI) memberikan peluang besar dalam membantu proses medis. Salah satu cabang AI yang berkembang pesat adalah *machine learning* (ML), yaitu metode yang memungkinkan sistem belajar dari data untuk mengenali pola dan melakukan inferensi [3, 4]. Dalam bidang kesehatan, *machine learning* sangat bermanfaat untuk klasifikasi kondisi medis dan prediksi risiko penyakit [5]. Beberapa algoritma klasifikasi yang banyak digunakan meliputi Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), dan Decision Tree [6].

Masing-masing algoritma memiliki pendekatan yang berbeda. Random Forest unggul dalam menangani interaksi fitur yang kompleks secara *ensemble* [7]. SVM memisahkan data menggunakan fungsi linear pada ruang fitur berdimensi tinggi [8]. KNN menentukan klasifikasi berdasarkan mayoritas kedekatan jarak antar data [9], sedangkan Decision Tree merepresentasikan data dalam bentuk aturan pohon keputusan [10].

Tantangan utama dalam penerapan klasifikasi pada data klinis adalah ketidakseimbangan kelas (*class imbalance*). Kondisi ini terjadi karena jumlah sampel pasien sehat umumnya jauh lebih besar dibandingkan pasien strok. Akibatnya, model cenderung bias terhadap kelas mayoritas sehingga performa deteksi pada kelas minoritas menjadi tidak optimal. Salah satu solusi efektif untuk mengatasi masalah ini adalah *Synthetic Minority Oversampling Technique* (SMOTE), yang bekerja dengan menghasilkan data sintetis pada kelas minoritas agar distribusi data menjadi lebih seimbang [11].

Penelitian terdahulu telah banyak menerapkan ML di bidang kesehatan, seperti komparasi algoritma Decision Tree dan SVM untuk klasifikasi kanker paru [10]. Namun,

terdapat kesenjangan penelitian (*research gap*) di mana sebagian besar studi prediksi penyakit strok hanya berfokus pada pencapaian metrik akurasi global [27, 28]. Penelitian-penelitian tersebut sering kali mengabaikan efektivitas model dalam mengenali kelas minoritas yang sebenarnya jauh lebih kritis.

Kebaruan (*novelty*) dari penelitian ini terletak pada analisis komparatif secara mendalam terhadap *trade-off* (tarik-ulur) antara akurasi keseluruhan dan sensitivitas deteksi kelas minoritas (*Recall*) pada prediksi strok. Penelitian ini secara spesifik membandingkan empat algoritma (Random Forest, SVM, KNN, dan Decision Tree) dengan mengintegrasikan metode SMOTE secara eksplisit untuk melihat dampak nyata penanganan *class imbalance* [18, 19]. Dalam konteks medis, meminimalkan kegagalan deteksi pada pasien strok (*False Negative*) dinilai lebih krusial dibandingkan sekadar mengejar akurasi yang tinggi. Oleh karena itu, penelitian ini bertujuan menemukan model yang paling optimal dan aman untuk kebutuhan prediksi klinis.

II. METODE PENELITIAN

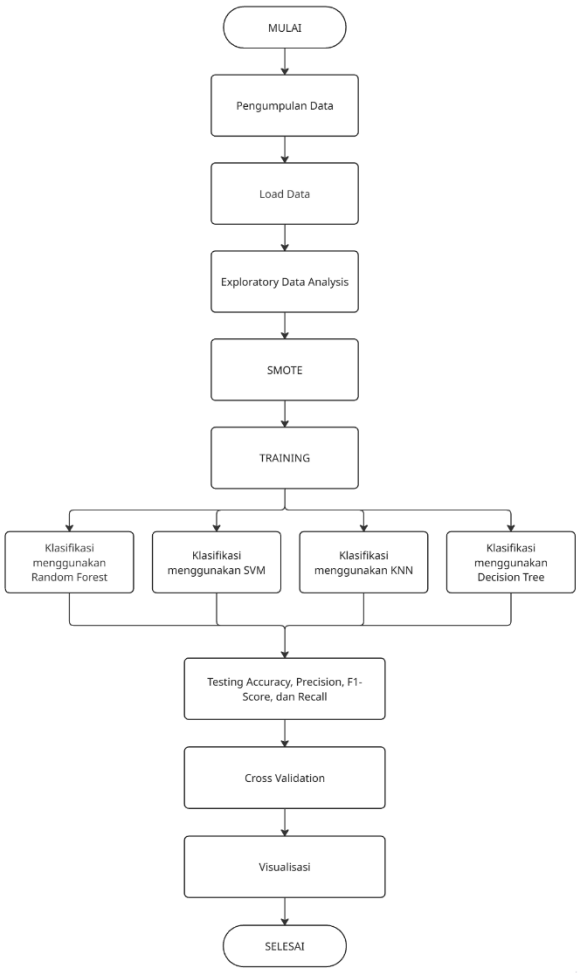
Adapun dalam penelitian ini terdapat sebelas tahapan yang terdiri dari pengumpulan data, *load* data atau *import* data, lalu dilakukan *exploratory data analysis* (EDA), SMOTE, *training data* menggunakan empat klasifikasi berbeda yaitu *Random Forest*, *Support Vector Machine, K-Nearest Neighbor* dan *Decision Tree*. Lalu dilakukan validasi atau *testing* menggunakan akurasi, presisi, *Recall* dan *F1-Score*. Dilanjutkan dengan *cross validation* melalui SMOTE serta visualisasi *output*, dijabarkan tahapan tersebut pada Gambar 1.

A. Landasan Teori

Data Mining

Menurut Ikhrom dalam Irsyad et al. [12], *data mining* adalah proses untuk menemukan pola-pola tersembunyi atau pengetahuan baru dari kumpulan data besar melalui teknik statistik, pemrograman, atau kecerdasan buatan. *Data mining* digunakan untuk memperbaiki pengambilan keputusan di masa yang akan datang berdasarkan informasi masa lalu, misalnya untuk prediksi, estimasi, asosiasi, clustering, dan deskripsi. Penerapan *data mining* dalam bidang kesehatan mendukung deteksi dini strok dengan mengidentifikasi serta menganalisis berbagai hubungan antarvariabel klinis pasien. *Data mining* biasanya digunakan untuk memperbaiki pengambilan keputusan di masa yang akan datang berdasarkan informasi yang diperoleh dari masa lalu. Misalnya untuk prediksi, estimasi, asosiasi *clustering*, dan deskripsi [13].

Model *data mining* seperti klasifikasi telah diterapkan dalam berbagai penelitian prediksi diabetes, menunjukkan bahwa metode ini mampu meningkatkan akurasi dan pendekatan yang dapat digunakan untuk mendukung deteksi dini strok adalah penerapan *data mining* pada data kesehatan pasien. Melalui proses *data mining*, berbagai pola dan hubungan antarvariabel yang berkaitan dengan risiko strok dapat diidentifikasi dan dianalisis. Hasil analisis tersebut kemudian dapat dimanfaatkan untuk membangun model klasifikasi yang mampu memprediksi atau mengelompokkan pasien berdasarkan kemungkinan mengalami strok [14].



Gambar 1. Metode Penelitian

Klasifikasi (Classification)

Klasifikasi secara umum dapat diartikan sebagai tindakan mengatur atau mengelompokkan sesuatu ke dalam golongan atau kelas-kelas tertentu. Lebih spesifik lagi, Klasifikasi data adalah proses menemukan model atau fungsi yang menjelaskan dan membedakan kelas data atau konsep guna memasukkan suatu objek ke dalam kelas tertentu yang telah ditentukan terlebih dahulu berdasarkan karakteristik yang sama [15]. Metode klasifikasi dievaluasi menggunakan metrik akurasi, presisi, *recall*, dan *F1-score* yang diperoleh melalui *confusion matrix*. Beberapa jenis algoritma klasifikasi yang umum digunakan meliputi *Decision Tree* (DT), *Support Vector Machine* (SVM), *Neural Network* (NN), *K-Nearest Neighbor* (KNN), dan *Naïve Bayes* (NB) [12]. Klasifikasi adalah proses memasukkan suatu anggota ke dalam kelas tertentu, namun kelas tersebut ditentukan terlebih dahulu, kemudian anggota-anggota kelas tersebut ditempatkan dalam suatu kelas berdasarkan karakteristik data yang sama [16].

Beberapa jenis algoritma berbeda yang dapat digunakan untuk klasifikasi, diantaranya DT, SVM, NN, K-NN dan NB. DT adalah *Decision Tree*, SVM adalah *Support Vector Machine*, NN adalah Neural Network, K-NN adalah K-Nearest Neighbour dan NB adalah Naïve

Bayes [16]. Dalam sebuah penelitian serupa mengenai strok oleh Iskandar *et al*. Klasifikasi dapat menghasilkan nilai *Accuracy* yang dihasilkan sebesar 95.2%, sensitivity sebesar 4.1%, specificity sebesar 99.8%, *Precision* sebesar 66.7%, dan F-measure sebesar 7.6%, menggunakan *Random Forest* [17].

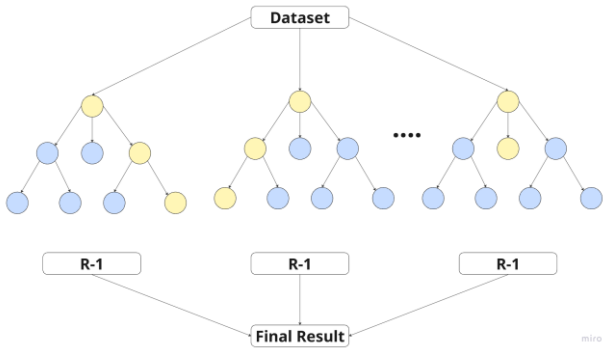
SMOTE

Metode SMOTE merupakan metode yang populer diterapkan dalam rangka menangani ketidak seimbangan kelas. Teknik ini mensintesis sampel baru dari kelas minoritas untuk menyeimbangkan dataset dengan cara sampling ulang sampel kelas minoritas [18]. Dalam sebuah penelitian oleh Pasiolo *et al*. Mengenai SMOTE pada klasifikasi penyakit strok. penerapan SMOTE meningkatkan akurasi, presisi, dan *Recall*, dengan kernel RBF mencapai akurasi tertinggi 92% pada parameter Cost 100 dan Gamma 1. Temuan ini menunjukkan bahwa penggunaan SMOTE dan optimasi parameter SVM dapat menghasilkan model klasifikasi yang lebih efektif dalam mendeteksi risiko strok pada data tidak seimbang [19].

Random Forest

Random Forest (RF) merupakan metode ensemble learning yang terdiri atas kumpulan pohon keputusan (*Decision Tree*), di mana setiap pohon dibangun menggunakan sampel data acak yang diambil dari distribusi yang sama. Algoritma ini menggabungkan hasil prediksi dari berbagai pohon keputusan untuk menghasilkan keputusan akhir yang lebih akurat dan stabil. Keunggulan utama *Random Forest* terletak pada proses pemilihan fitur secara acak selama pembentukan pohon, sehingga mampu mengurangi risiko *overfitting* dan menghasilkan tingkat kesalahan prediksi yang relatif rendah [13].

Pada penelitian yang dilakukan oleh Muttakin, *et al*. Mengenai analisis perbandingan algoritma klasifikasi untuk prediksi risiko penyakit jantung [25]. Pada penelitian tersebut, hasil evaluasi menunjukkan *Decision Tree* dan *Random Forest* memiliki akurasi terbaik yaitu mencapai 99%. Sedangkan KNN dan SVM memiliki akurasi 84% dan 88%. Oleh karena itu, pemilihan model harus mempertimbangkan keseimbangan antara akurasi dan kemampuan generalisasi.



Gambar 2. Algoritma Random Forest

Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan salah satu metode pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. Algoritma ini dirancang untuk menangani permasalahan linier maupun non-linier dengan memanfaatkan fungsi kernel yang memetakan data ke dalam ruang fitur berdimensi lebih tinggi. Melalui pendekatan tersebut, SVM mampu menemukan hyperplane optimal yang memisahkan data antar kelas dengan margin maksimum [13].

Konsep dasar SVM berawal dari klasifikasi linier pada data yang dapat dipisahkan secara linier, yang kemudian dikembangkan untuk mengatasi data dengan pola yang lebih kompleks [13]. SVM sangat efektif digunakan untuk data berdimensi tinggi dan dapat menangani kasus klasifikasi linear maupun non-linear melalui pemanfaatan fungsi kernel [12]. Persamaan 1 merupakan fungsi dari SVM itu sendiri.

$$f(x) = \sum_{i=1}^{n_s} a_i y_i K(x_i x) + b \quad (1)$$

(f(x)) merupakan fungsi keputusan yang digunakan untuk menentukan kelas dari data yang akan diklasifikasikan. Simbol (n_s) menyatakan jumlah *support vector* yang digunakan dalam model. Nilai (a_i) merupakan koefisien atau *Lagrange multiplier* yang menunjukkan kontribusi masing-masing *support vector* terhadap proses klasifikasi, sedangkan (y_i) adalah label kelas dari *support vector* ke-(i) yang umumnya bernilai +1 atau -1. Fungsi ($K(x_i,x)$) merupakan fungsi kernel yang digunakan untuk mengukur tingkat kemiripan antara *support vector* (x_i) dan data baru (x) yang akan diprediksi. Adapun (b) adalah nilai bias atau konstanta yang diperoleh selama proses pelatihan model dan berfungsi untuk menentukan posisi *hyperplane* pemisah antar kelas.

Nilai (f(x)) yang dihasilkan akan menentukan kelas suatu data. Jika ($f(x) > 0$), maka data diklasifikasikan ke dalam kelas positif. Sebaliknya, jika ($f(x) < 0$), maka data diklasifikasikan ke dalam kelas negatif. Dengan memanfaatkan fungsi kernel, SVM mampu menangani permasalahan klasifikasi yang bersifat non-linier dengan memetakan data ke ruang fitur berdimensi lebih tinggi sehingga pemisahan antar kelas dapat dilakukan secara lebih optimal.

K-Nearest Neighbor (KNN)

K-Nearest Neighbor (KNN) merupakan salah satu algoritma klasifikasi yang mengelompokkan suatu objek berdasarkan kedekatannya dengan data latih yang telah diketahui kelasnya. Algoritma ini bekerja dengan mencari sejumlah tetangga terdekat (*nearest neighbors*) dari data uji berdasarkan ukuran jarak tertentu. Kelas dari data uji kemudian ditentukan berdasarkan mayoritas kelas dari tetangga terdekat tersebut [13]. Algoritma KNN pada umumnya digunakan dalam mengklasifikasikan objek berdasarkan pada data pembelajaran yang memiliki nilai selisih kecil dan jarak tetangga terdekat dengan objek, dapat bermanfaat juga dalam prediksi strok [20].

Dalam proses klasifikasi, KNN menggunakan karakteristik atau fitur yang dimiliki oleh data latih dan data uji untuk menghitung tingkat kemiripan antar data. Salah satu metode yang paling umum digunakan untuk

mengukur kedekatan antar data adalah jarak Euclidean (*Euclidean Distance*) [13]. Semakin kecil nilai jarak yang diperoleh, semakin tinggi tingkat kemiripan antara kedua data tersebut. Jika terdapat dua buah data, yaitu ((a_1, a_2, a_3, ..., a_n)) dan ((b_1, b_2, b_3, ..., b_n)), maka jarak Euclidean antara keduanya dapat dihitung menggunakan persamaan 2.

$$d(a,b) = \sqrt{\sum_{\{i=1\}}^{n} (a_i - b_i)^2} \quad (2)$$

Pada persamaan 2, (d(a,b)) menyatakan jarak antara dua data, yaitu data (a) dan data (b). Variabel (a_i) dan (b_i) masing-masing menunjukkan nilai atribut ke-(i) dari kedua data, sedangkan (n) merupakan jumlah atribut yang digunakan. Nilai jarak yang lebih kecil menunjukkan tingkat kemiripan yang lebih tinggi antar data. Dalam algoritma *K-Nearest Neighbor* (KNN), jarak ini digunakan untuk menentukan tetangga terdekat yang menjadi dasar proses klasifikasi.

Decision Tree

Decision Tree adalah algoritma klasifikasi yang merepresentasikan data dalam bentuk struktur pohon keputusan untuk menghasilkan prediksi atau keputusan. Algoritma ini bekerja dengan membagi data berdasarkan atribut tertentu hingga terbentuk beberapa percabangan. Pada pohon keputusan, simpul internal menunjukkan proses pengujian atribut, sedangkan simpul daun merepresentasikan hasil akhir atau kelas dari proses klasifikasi [10].

Decision Tree dapat digunakan atau diimplementasikan dalam banyak kasus seperti klasifikasi medis dalam bentuk kanker [10], strok [21]. Struktur *Decision Tree* terdiri atas *root node* (akar), *internal node* (cabang pengujian atribut), dan *leaf node* (daun keputusan kelas). Proses pembagian data (*splitting*) ditentukan berdasarkan nilai *Information Gain* terbesar yang dihitung melalui persamaan 3.

$$Gain(S, \mathrm{A}) = \mathrm{E}(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \mathrm{E}S_v \quad (3)$$

Pada persamaan 3, (Gain (S, A)) menyatakan nilai gain dari atribut (A), (Entropy(S)) adalah entropy data sebelum dilakukan pemisahan, sedangkan (Entropy (S_v)) merupakan *entropy* setiap subset data setelah dibagi berdasarkan nilai atribut (A). Atribut dengan nilai *Information Gain* terbesar akan dipilih sebagai *node* karena mampu memberikan pemisahan data yang paling baik.

Confusion Matrix

Confusion Matrix adalah metode yang digunakan untuk mengevaluasi kinerja model klasifikasi. Evaluasi dilakukan menggunakan beberapa metrik, yaitu *Accuracy*, *Precision*, dan *Recall*. *Recall* mengukur kemampuan model dalam mengenali data positif yang sebenarnya, *Precision* menunjukkan ketepatan prediksi positif yang dihasilkan model, sedangkan *Accuracy* menggambarkan persentase keseluruhan prediksi yang berhasil diklasifikasikan dengan benar [22].

Tabel 1. Confusion Matrix

Actual	Predicted (Strok)	Predicted (Tidak Strok)
Strok	TP	FN
Tidak Strok	FP	TN

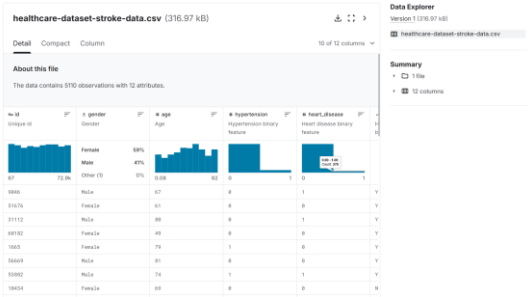
1) TP (*True Positive*): Pasien positif strok yang berhasil diprediksi dengan benar oleh model sebagai kelas strok.
2) FN (*False Negative*): Pasien tidak strok yang berhasil diprediksi dengan benar oleh model sebagai kelas tidak strok.
3) FP (*False Positive*): Pasien tidak strok namun salah diprediksi oleh model sebagai kelas strok.
4) FN (*False Negative*): Pasien positif strok namun salah diprediksi oleh model sebagai kelas tidak strok.

Dalam sebuah penelitian oleh Sari, *et al*. Mengenai deteksi dini strok menggunakan *machine learning* [24]. Akurasi (*Accuracy*) sebagai rasio prediksi benar terhadap total prediksi. Presisi dan *Recall* sebagai mengukur performa dalam mengidentifikasi kasus strok. *F1-Score* untuk harmonik rata-rata dari presisi dan *Recall*. *ROC Curve* dan AUC digunakan untuk mengukur *trade-off* antara TPR dan FPR.

III. HASIL DAN PEMBAHASAN

A. *Load Data*

Dataset yang digunakan dalam penelitian ini adalah *Healthcare* Strok *Dataset* yang diperoleh dari *platform* Kaggle terdiri dari 5.110 data pasien [23]. Dataset tersebut memiliki dua kelas target, yaitu kelas tidak strok sebanyak 4.861 sampel (95,13%) dan kelas strok sebanyak 249 sampel (4,87%), sehingga menunjukkan adanya ketidakseimbangan kelas (class imbalance). Setiap data pasien memiliki 11 fitur yang digunakan sebagai variabel prediktor, yaitu gender (jenis kelamin), age (usia), hypertension (riwayat hipertensi), heart_disease (riwayat penyakit jantung), ever_married (status pernikahan), work_type (jenis pekerjaan), Residence_type (tipe tempat tinggal), avg_glucose_level (rata-rata kadar glukosa darah), bmi (Body Mass Index), dan smoking_status (status merokok). Adapun variabel strok digunakan sebagai atribut target yang menunjukkan apakah pasien mengalami strok (1) atau tidak mengalami strok (0).



Gambar 3. Healthcare Strok Dataset Kaggle

B. *Exploratory Data Analysis (EDA)*

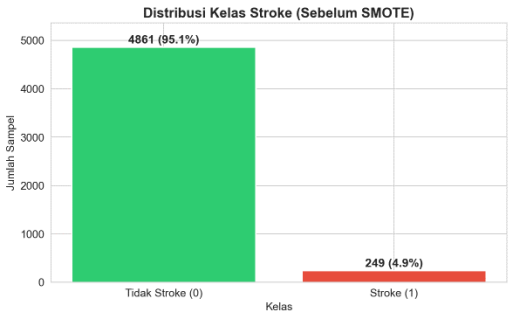
Exploratory Data Analysis (EDA) merupakan tahap awal yang dilakukan untuk memahami karakteristik dataset sebelum proses *preprocessing* dan pemodelan. Pada tahap ini dilakukan analisis terhadap distribusi kelas, pemeriksaan nilai yang hilang (missing values), analisis statistik deskriptif, analisis korelasi antar fitur, serta visualisasi distribusi data berdasarkan kelas target.

```
============================================================
2. EXPLORATORY DATA ANALYSIS (EDA)
============================================================

--- Distribusi Kelas (Sebelum SMOTE) ---
  Tidak Stroke (0) : 4861 (95.13%)
  Stroke (1)       : 249 (4.87%)
  Rasio            : 1 : 19 (Sangat Tidak Seimbang)
  [OK] Grafik distribusi kelas (sebelum SMOTE) disimpan

--- Missing Values ---
bmi    201

--- Statistik Deskriptif (Fitur Numerik) ---
                age  avg_glucose_level          bmi       stroke
count  5110.000000        5110.000000  4909.000000  5110.000000
mean     43.226614         106.147677    28.893237     0.048728
std      22.612647          45.283560     7.854067     0.215320
min       0.080000          55.120000    10.300000     0.000000
25%      25.000000          77.245000    23.500000     0.000000
50%      45.000000          91.885000    28.100000     0.000000
75%      61.000000         114.090000    33.100000     0.000000
max      82.000000         271.740000    97.600000     1.000000
  [OK] Heatmap korelasi disimpan
  [OK] Boxplot fitur disimpan
```

Gambar 4. Exploratory Data Analysis

Analisis distribusi kelas dilakukan untuk mengetahui proporsi data pada setiap kelas target. Hasil analisis menunjukkan bahwa jumlah data tidak strok lebih banyak dibandingkan data strok sehingga dataset mengalami ketidakseimbangan kelas (*class imbalance*). Pemeriksaan missing values dilakukan untuk mengidentifikasi atribut yang memiliki data kosong sehingga dapat ditentukan metode penanganan yang sesuai sebelum proses pemodelan.

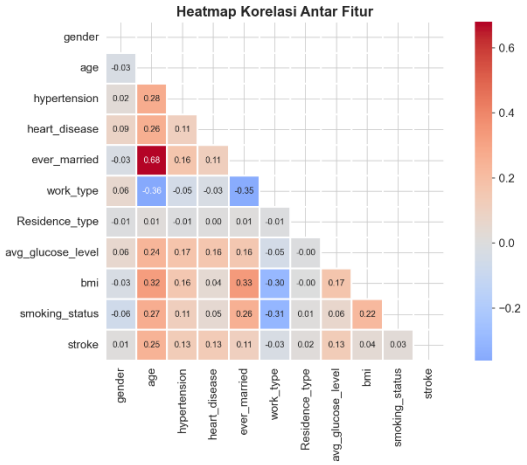
Analisis statistik deskriptif digunakan untuk mengetahui karakteristik fitur numerik, seperti nilai minimum, maksimum, rata-rata, dan standar deviasi. Heatmap korelasi digunakan untuk melihat hubungan antar fitur dalam dataset serta mengetahui fitur yang memiliki keterkaitan dengan variabel target. Visualisasi *boxplot* digunakan untuk membandingkan distribusi fitur numerik berdasarkan kelas strok dan tidak strok serta mendeteksi adanya nilai ekstrem (*outlier*).



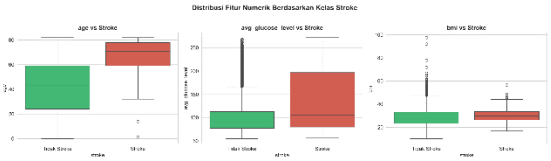
Gambar 5. Distribusi kelas strok

Berdasarkan grafik distribusi kelas, jumlah data pada kelas tidak strok sebanyak 4.861 sampel (95,1%), sedangkan kelas strok hanya sebanyak 249 sampel (4,9%). Hal ini menunjukkan bahwa dataset mengalami ketidakseimbangan kelas (*class imbalance*).

Berdasarkan heatmap korelasi Gambar 6, variabel *age* memiliki korelasi tertinggi terhadap variabel strok dengan nilai sebesar 0,25. Selain itu, variabel *hypertension* dan *heart_disease* juga menunjukkan korelasi positif terhadap kejadian strok dengan nilai masing-masing sebesar 0,13. Secara umum, tidak terdapat korelasi yang sangat kuat antar fitur sehingga seluruh fitur masih layak digunakan dalam proses klasifikasi.



Gambar 6. Heatmap korelasi

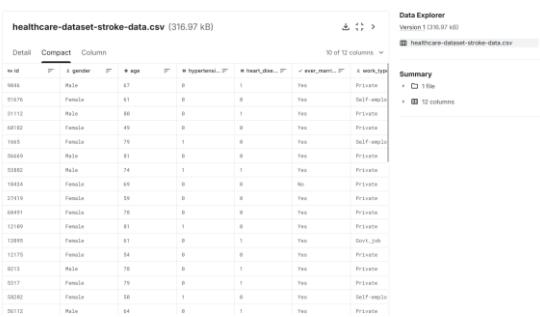


Gambar 7. Boxplot fitur

Berdasarkan *boxplot*, pasien yang mengalami strok cenderung memiliki usia (*age*) yang lebih tinggi dibandingkan pasien yang tidak mengalami strok. Selain itu, kelompok strok juga menunjukkan nilai *avg_glucose_level* yang cenderung lebih tinggi. Sementara itu, distribusi nilai BMI pada kedua kelas relatif mirip sehingga pengaruhnya terhadap kejadian strok tidak terlihat sekuat variabel usia dan kadar glukosa darah.

C. *Preprocessing dan Penerapan SMOTE*

Tahap *preprocessing* dilakukan untuk mempersiapkan data sebelum digunakan dalam proses pelatihan model. Langkah pertama adalah menghapus atribut id karena tidak memiliki pengaruh terhadap proses klasifikasi. Selanjutnya, nilai yang hilang (*missing value*) pada atribut BMI diatasi dengan menggantinya menggunakan nilai median. Data dengan kategori gender bernilai “Other” juga dihapus karena jumlahnya sangat sedikit dan dapat memengaruhi proses analisis.

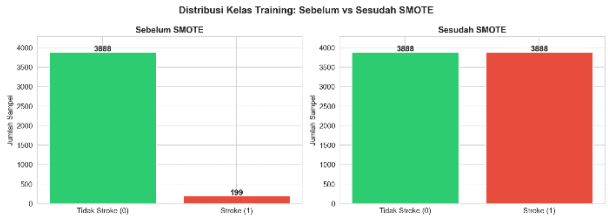


Gambar 8. Healthcare Strok Dataset

Setelah itu, atribut kategorikal seperti gender, ever_married, work_type, Residence_type, dan

smoking_status diubah ke dalam bentuk numerik menggunakan teknik Label Encoding. Data kemudian dipisahkan menjadi data latih (*training data*) sebesar 80% dan data uji (*testing data*) sebesar 20% menggunakan metode *stratified split* agar proporsi kelas tetap terjaga.

Dilakukan normalisasi menggunakan StandardScaler untuk menyamakan skala antar fitur. Tahap terakhir adalah penerapan SMOTE (*Synthetic Minority Oversampling Technique*) pada data latih untuk mengatasi ketidakseimbangan kelas dengan menghasilkan sampel sintetis pada kelas minoritas.



Gambar 9. Distribusi sebelum dan sesudah SMOTE

Berdasarkan Gambar 9 yaitu grafik distribusi SMOTE, sebelum penerapan SMOTE jumlah data pada kelas tidak strok sebanyak 3.888 sampel, sedangkan kelas strok hanya sebanyak 199 sampel. Kondisi ini menunjukkan adanya ketidakseimbangan kelas yang cukup signifikan. Setelah penerapan SMOTE, jumlah sampel pada kelas strok bertambah menjadi 3.888 sampel sehingga jumlah data pada kedua kelas menjadi seimbang. Dengan distribusi yang lebih seimbang, model diharapkan dapat mempelajari pola pada kedua kelas dengan lebih baik dan mengurangi bias terhadap kelas mayoritas. SMOTE dapat menyeimbangkan distribusi *data training* dari 3.888:199 menjadi 3.888:3.888, sehingga permasalahan *class imbalance* dapat diminimalkan sebelum proses pelatihan model dilakukan.

D. Training dan Evaluation

Setelah tahap *preprocessing* selesai dilakukan, proses selanjutnya adalah pelatihan (*training*) dan evaluasi model klasifikasi. Pada penelitian ini digunakan empat algoritma *machine learning*, yaitu *Random Forest*, *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Decision Tree*. Keempat algoritma tersebut dipilih karena memiliki karakteristik yang berbeda dalam melakukan proses klasifikasi sehingga dapat dibandingkan performanya dalam mendeteksi penyakit strok.

Setelah model dilatih, dilakukan prediksi terhadap *data testing* untuk memperoleh hasil klasifikasi. Kinerja masing-masing model kemudian dievaluasi menggunakan beberapa metrik pengukuran, yaitu *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. *Accuracy* digunakan untuk mengukur tingkat ketepatan model dalam mengklasifikasikan seluruh data. *Precision* digunakan untuk mengukur ketepatan prediksi pada suatu kelas, sedangkan *Recall* digunakan untuk mengukur kemampuan model dalam mendeteksi seluruh data yang benar-benar termasuk dalam kelas tertentu. *F1-Score* merupakan rata-rata harmonik antara *Precision* dan *Recall* yang digunakan untuk memberikan gambaran performa model secara lebih seimbang.

Selain menghitung metrik evaluasi secara keseluruhan, penelitian ini juga melakukan evaluasi khusus pada kelas strok. Evaluasi tersebut dilakukan dengan menghitung nilai *Precision*, *Recall*, dan *F1-Score* untuk kelas strok guna mengetahui kemampuan model dalam mengidentifikasi pasien yang berisiko mengalami strok. Hal ini penting karena tujuan utama penelitian adalah meningkatkan kemampuan deteksi terhadap kelas minoritas, yaitu pasien strok.

Selanjutnya, hasil prediksi dianalisis menggunakan *Classification Report* yang menampilkan nilai *Precision*, *Recall*, *F1-Score*, dan jumlah sampel pada masing-masing kelas. *Classification Report* memberikan informasi yang lebih rinci mengenai performa model dalam mengklasifikasikan data strok maupun tidak strok.

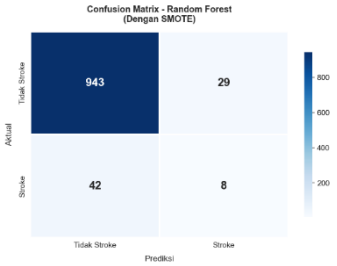
Untuk memastikan kestabilan dan kemampuan generalisasi model, dilakukan validasi menggunakan metode *5-Fold Cross Validation*. Pada metode ini, dataset dibagi menjadi lima bagian, di mana empat bagian digunakan sebagai data pelatihan dan satu bagian digunakan sebagai data pengujian secara bergantian hingga lima kali percobaan. Pada setiap *fold*, proses normalisasi dan SMOTE diterapkan melalui *pipeline* sehingga tidak terjadi kebocoran data (*data leakage*). Hasil validasi kemudian dihitung dalam bentuk rata-rata dan standar deviasi untuk mengetahui konsistensi performa masing-masing algoritma. Model dengan nilai evaluasi terbaik akan dipilih sebagai model yang paling efektif dalam melakukan klasifikasi penyakit strok.

Tabel 2. Parameter Algoritma

Algoritma	Hyper-parameter	Nilai	Fungsi
Random Forest	n_estimators	100	Jumlah pohon keputusan paralel
	criterion	'gini'	Fungsi pengukur kualitas pemisahan cabang
	random_state	42	Kontrol konsistensi keacakan sampel
Support Vector Machine	kernel	'rbf'	Jenis fungsi transformasi non-linier data
	C	1.0	Parameter penalti toleransi salah klasifikasi
	gamma	'scale'	Koefisien lebar jangkauan pengaruh kernel
K-Nearest Neighbor	n_neighbors	5	Jumlah tetangga terdekat dasar *voting*
	metric	'minkowski'	Penggunaan fungsi jarak setara Euclidean

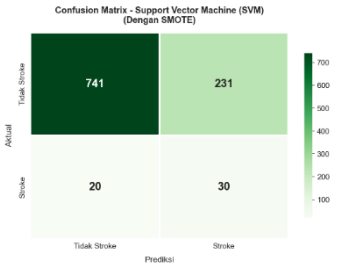
	weights	(p=2) 'uniform'	Bobot suara yang sama untuk semua tetangga
Decision Tree	criterion	'gini'	Metrik seleksi fitur pembentuk cabang
	max_depth	None (unlimited)	Batas kedalaman pertumbuhan pohon keputusan
	random_state	42	Kontrol replikasi hasil *splitting* acak

E. Visualisasi



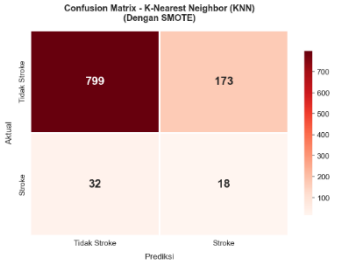
Gambar 10. Confusion matrix random forest

Random Forest berhasil mengklasifikasikan 943 data tidak strok dengan benar (*True Negative*) dan 8 data strok dengan benar (*True Positive*). Namun, terdapat 29 data tidak strok yang salah diklasifikasikan sebagai strok (*False Positive*) serta 42 data strok yang salah diklasifikasikan sebagai tidak strok (*False Negative*).



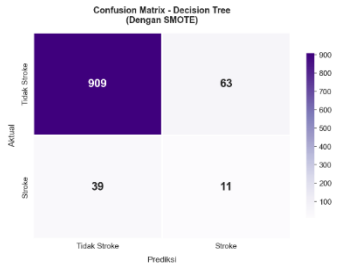
Gambar 11. Confusion matrix SVM

Pada algoritma SVM, sebanyak 741 data tidak strok berhasil diklasifikasikan dengan benar dan 30 data strok berhasil terdeteksi dengan benar. Selain itu, terdapat 231 data tidak strok yang diprediksi sebagai strok dan 20 data strok yang tidak berhasil terdeteksi. Dibandingkan *Random Forest*, SVM mampu mendeteksi lebih banyak kasus strok, namun menghasilkan kesalahan prediksi yang lebih tinggi pada kelas tidak strok.



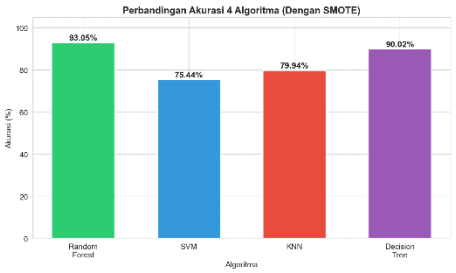
Gambar 12. Confusion matrix KNN

Berdasarkan hasil Confusion Matrix, algoritma KNN berhasil mengklasifikasikan 799 data tidak strok dengan benar dan 18 data strok dengan benar. Terdapat 173 data tidak strok yang salah diprediksi sebagai strok serta 32 data strok yang salah diprediksi sebagai tidak strok.



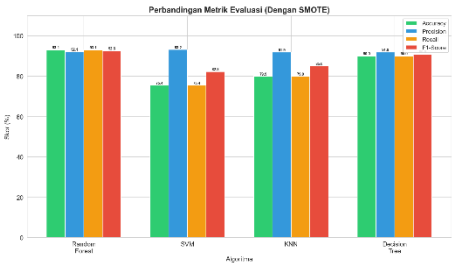
Gambar 13. Confusion matrix Decision Tree

Decision Tree berhasil mengklasifikasikan 909 data tidak strok dengan benar (*True Negative*) dan 11 data strok dengan benar (*True Positive*). Namun, masih terdapat 63 data tidak strok yang salah diprediksi sebagai strok (*False Positive*) serta 39 data strok yang salah diprediksi sebagai tidak strok (*False Negative*).



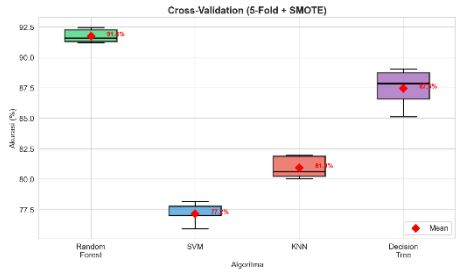
Gambar 14. Perbandingan akurasi algoritma

Grafik perbandingan akurasi menunjukkan bahwa algoritma *Random Forest* memperoleh nilai akurasi tertinggi sebesar 93,05%, diikuti oleh *Decision Tree* sebesar 90,02%, KNN sebesar 79,94%, dan SVM sebesar 75,44%. Hasil ini mengindikasikan bahwa *Random Forest* memiliki kemampuan terbaik dalam mengklasifikasikan data secara keseluruhan setelah penerapan SMOTE. Sementara itu, SVM menghasilkan akurasi terendah dibandingkan algoritma lainnya.



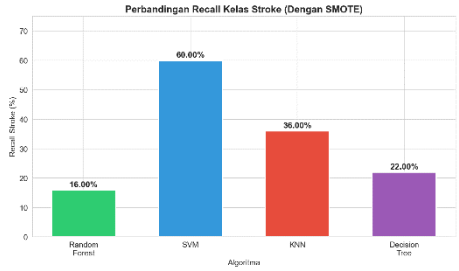
Gambar 15. Perbandingan metrik evaluasi

Grafik perbandingan metrik evaluasi memperlihatkan nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dari setiap algoritma. *Random Forest* menunjukkan performa yang paling seimbang dengan seluruh metrik berada di atas 90%. *Decision Tree* juga menghasilkan performa yang cukup baik dengan nilai metrik yang relatif tinggi dan stabil. Sementara itu, KNN dan SVM memiliki nilai *Accuracy* dan *Recall* yang lebih rendah dibandingkan dua algoritma lainnya.



Gambar 16. 5-Fold Cross Validation

Hasil cross-validation menggunakan metode 5-Fold menunjukkan tingkat konsistensi performa dari masing-masing algoritma. *Random Forest* memiliki rata-rata akurasi tertinggi dengan variasi nilai yang relatif kecil, yang menandakan bahwa model ini stabil ketika diuji pada berbagai pembagian data. *Decision Tree* berada pada posisi kedua dengan rata-rata akurasi yang cukup tinggi, sedangkan KNN dan SVM memperoleh rata-rata akurasi yang lebih rendah.



Gambar 17. Perbandingan Recall

Grafik *Recall* kelas strok menunjukkan kemampuan masing-masing algoritma dalam mendeteksi kasus strok. SVM memperoleh nilai *Recall* tertinggi sebesar 60%, diikuti oleh KNN sebesar 36%, *Decision Tree* sebesar 22%, dan *Random Forest* sebesar 16%.

Tabel 3. Performa Seluruh Metrik Klasifikasi

Algoritma	Accuracy	Precision	Recall	F1-Score	Rata-rata Akurasi
Random Forest	93,05%	>90,00%	16,00%	>90,00%	Tertinggi
SVM	90,02%	Tinggi	22,00%	Tinggi	Tinggi
KNN	79,94%	Rendah	36,00%	Rendah	Rendah
Decision Tree	75,44%	Rendah	60,00%	Rendah	Terendah

IV. KESIMPULAN

Hasil penelitian menunjukkan bahwa *Random Forest* memperoleh akurasi tertinggi sebesar 93,05%, sedangkan SVM memiliki kemampuan terbaik dalam mendeteksi kasus strok dengan nilai *recall* kelas strok sebesar 60%. KNN dan *Decision Tree* juga menunjukkan performa yang cukup baik, namun masih berada di bawah *Random Forest* dan SVM.

Penelitian ini membuktikan bahwa nilai akurasi yang tinggi tidak selalu menunjukkan kemampuan deteksi penyakit yang baik, terutama pada kasus *imbalanced data*. Dalam konteks medis, kemampuan mendeteksi pasien strok lebih penting dibandingkan akurasi keseluruhan karena kesalahan deteksi dapat berakibat fatal bagi pasien.

Penelitian ini memiliki beberapa keterbatasan, di antaranya ketiadaan analisis komparatif performa *baseline* kuantitatif pada metrik sekunder sebelum penerapan SMOTE, serta cakupan *hyperparameter tuning* yang belum dieksplorasi secara maksimal. Untuk penelitian selanjutnya, disarankan melakukan optimasi parameter lanjutan secara spesifik pada SVM guna menekan rasio salah prediksi (*False Positive*) tanpa mengorbankan sensitivitas kelas strok, serta mengkaji metode *feature selection* untuk efisiensi dimensionalitas data.

DAFTAR PUSTAKA

[1] World Health Organization. (2021). Strok. https://www.who.int/news-room/fact-sheets/detail/strok

[2] Kementerian Kesehatan Republik Indonesia. (2024). Cegah Strok dengan Aktivitas Fisik. https://www.kemkes.go.id/id/cegah-strok-dengan-aktivitas-fisik

[3] Ahmad, A. (2017). Mengenal Artificial Intelligence, Machine learning, Neural Network, dan Deep Learning. Jurnal Teknologi Indonesia, Yayasan Cahaya Islam.

[4] Nurhalizah, R. S., Ardianto, R., & Purwono. (2024). Analisis supervised dan unsupervised learning pada machine learning: Systematic literature review. Jurnal Ilmu Komputer dan Informatika (JIKI), 4(1), 61–72. https://doi.org/10.54082/jiki.168

[5] Akbar, I., Supriadi, F., & Junaedi, D. I. (2025). Pemanfaatan machine learning di bidang kesehatan. ATI (Jurnal Mahasiswa Teknik Informatika), 9(1), 1744.

[6] Diba, F., Lydia, M. S., & Sihombing, P. (2023). Analisis random forest menggunakan principal component analysis pada data berdimensi tinggi. Indonesian Journal of Computer Science, 12(4), 2152.

[7] Dwiyanti, Z., & Prianto, C. (2023). Prediksi cuaca Kota Jakarta menggunakan metode Random Forest: Studi optimalitas. Jurnal Tekno Insentif, 17(2), 127–137. https://doi.org/10.36787/jti.v17i2.1136

[8] Puspitasari, A. M., Ratnawati, D. E., & Widodo, A. W. (2018). Klasifikasi penyakit gigi dan mulut menggunakan metode Support Vector Machine. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 2(2), 802–810.

[9] Akbar, F. M. N. (2024). Metode KNN (K-Nearest Neighbor) untuk menentukan kualitas air. Jurnal TEKNO KOMPAK, 18(1), 28–40.

[10] Septhya, D., Rahayu, K., Rabbani, S., Fitria, V., Rahmaddeni, Irawan, Y., & Hayami, R. (2023). Implementation of Decision Tree algorithm and Support Vector Machine for lung cancer classification. MALCOM: Indonesian Journal of Machine learning and Computer Science, 3(1), 15–19. https://journal.irpi.or.id/index.php/malcom/article/view/591

[11] Ridwan, Hermaliani, E. H., & Ernawati, M. (2024). Penerapan metode SMOTE untuk mengatasi imbalanced data pada klasifikasi ujaran kebencian. Computer Science (CO-SCIENCE), 4(1), 80.

[12] Hatta Irsyad, H. ., Ikran Syafwan, M., & ramadhani, dian. (2025). Analisis Perbandingan Kinerja Algoritma K-Nearest Neighbors dan Support Vector Machine untuk Klasifikasi Penyakit Diabetes. Journal of System & Technology (SYSTEC), 1(2), 43-48. https://systec.ejournal.unri.ac.id/index.php/systec/article/view/20

[13] Y. Azhar, A. K. Firdausy, dan P. J. Amelia, "Perbandingan Algoritma Klasifikasi *Data mining* untuk Prediksi Penyakit Strok," SINTECH Journal, vol. 5, no. 2, Okt. 2022, https://doi.org/10.31598.

[14] A. F. Riany dan G. Testiana, "Penerapan *Data mining* untuk Klasifikasi Penyakit Strok Menggunakan Algoritma Naïve Bayes," Jurnal Saintekom: Sains, Teknologi, Komputer, dan Manajemen, vol. 13, no. 1, pp. 42–54, 2023, https://doi.org/10.33020/saintekom.v13i1.352

[15] "Analisis Komparasi Algoritma Klasifikasi Data Mining Dalam Klasifikasi Website Phishing", Komputika, vol. 11, no. 1, pp. 59–66, Jan. 2022, doi: 10.34010/komputika.v11i1.4350.

[16] S. Rahayu dan Y. Yamasari, "Klasifikasi Penyakit Strok dengan Metode Support Vector Machine (SVM)," JINACS (Journal of Informatics and Computer Science), vol. 5, no. 3, pp. 440–447, 2024.

[17] N. A. Iskandar, I. Ernawati, dan Y. Widiastiwi, "Klasifikasi Diagnosis Penyakit Strok Dengan Menggunakan Metode Random Forest," dalam Seminar Nasional Mahasiswa Ilmu Komputer dan

Aplikasinya (SENAMIKA), Jakarta, Indonesia, 20 Agustus 2022, pp. 676–685.
[18] N. Sulistiyowati dan M. Jajuli, "Integrasi Naïve Bayes dengan Teknik Sampling SMOTE untuk Menangani Data Tidak Seimbang," Jurnal Nuansa Informatika, vol. 14, no. 1, pp. 34–37, Jan. 2020.
[19] L. Pasiolo, I. Afrianty, E. Budianita, dan R. Abdillah, "Penerapan Teknik SMOTE pada Klasifikasi Penyakit Strok dengan Algoritma Support Vector Machine," ZONAsi: Jurnal Sistem Informasi, vol. 7, no. 1, pp. 61–73, Jan. 2025.
[20] Naja Maskuri, M., Harliana, H., Sukerti, K., & Herdian Bhakti, R. M. (2022). Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Prediksi Penyakit Strok. Jurnal Ilmiah Intech: Information Technology Journal of UMUS, 4(01), 130–140. https://doi.org/10.46772/intech.v4i01.751
[21] Hendriyansyah, A. I. Purnamasari, dan T. Suprapti, "Penerapan Algoritma Decision Tree dalam Klasifikasi Penyakit Strok Otak," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 8, no. 3, pp. 3038–3043, Jun. 2024.
[22] Rininda, G., Santi, I. H., & Kirom, S. (2023). Penerapan Svm Dalam Analisis Sentimen Pada Edlink Menggunakan Pengujian Confusion Matrix. JATI (Jurnal Mahasiswa Teknik Informatika), 7(5), 3335-3342.
[23] Fedesoriano. (2020). Strok Prediction Dataset. Kaggle. Retrieved from https://www.kaggle.com/datasets/fedesoriano/strok-prediction-dataset
[24] Sari, K., Fadli, M., Fudholi, M. F., & Susanto, E. R. (2025). Deteksi Dini Strok Menggunakan Machine learning. INSOLOGI: Jurnal Sains dan Teknologi, 4(4), 706-720.
[25] Muhammad Muttakin, Najwa Rokhan Rusmana, & ramadhani, dian. (2025). Analisis Perbandingan Algoritma Decision Tree, Random Forest, KNN, dan SVM Dalam Prediksi Penyakit Jantung. Journal of System & Technology (SYSTEC), 1(2), 35-42. https://systec.ejournal.unri.ac.id/index.php/systec/article/view/18