



Implementasi Algoritma *Support Vector Machine* (SVM) Untuk Klasifikasi Penyakit *Stroke*

Danis Rifa Nurqotimah¹, Ahsanun Naseh Khudori², Risqy Siwi Pradini³

^{1,2,3}Informatika, Fakultas Sains dan Teknologi, Institut Teknologi, Sains, dan Kesehatan RS. DR. Soepraoen Kesdam
V/BRW, Malang

¹920407.danisrifa@gmail.com, ²ahsanunnaseh@itsk-soepraoen.ac.id, ³risqypradini@itsk-soepraoen.ac.id

Abstract

Stroke or known as Cerebrovascular Accident (CVA) is a functional disorder caused by impaired blood flow function from within the human brain. Stroke carries a high risk of brain damage, paralysis, speech disorders, visual impairment, even death. Classification is one of a few methods in predicting stroke symptoms with the aim of obtaining accurate prediction of disease. The researchers implemented a method to classify stroke with the Support Vector Machine (SVM) algorithm. The SVM is a learning method used in medical diagnosis for classification, the researchers processed data sets using the Orange tool. The study used data sets from the data.world.com site with a total of 40,910 data. Using the Orange tool, the study managed to classify stroke disease well using the RBF kernel with cross validation techniques resulting in an accuracy of 94.8%. The results of this study can be concluded that the stroke classification model developed has excellent performance. Overall, these results indicate that the Stroke classification model developed is highly reliable and effective, with excellent ability to detect stroke cases and provide accurate predictions. Making better and quicker medical judgments can be aided by using this approach to diagnose strokes.

Keywords: *Stroke, support vector machine, classification*

Abstrak

Stroke atau yang dikenal dengan Cerebrovaskular Accident (CVA) ialah gangguan fungsional yang diakibatkan dari permasalahan fungsi aliran darah dari pada otak manusia. Stroke membawa risiko tinggi pada kerusakan otak, kelumpuhan, gangguan bicara, gangguan penglihatan, hingga kematian. Klasifikasi menjadi satu Peneliti mengimplementasikan metode untuk mengklasifikasi penyakit stroke dengan algoritma Support Vector Machine (SVM). Algoritma SVM merupakan suatu metode pembelajaran yang digunakan dalam diagnosis medis untuk klasifikasi, peneliti melakukan pengolahan dataset dengan menggunakan tool Orange. Penelitian ini menggunakan dataset dari situs data.world.com dengan jumlah 40.910 data. Dengan menggunakan tool Orange, penelitian ini berhasil mengklasifikasi penyakit stroke dengan baik menggunakan kernel RBF dengan teknik cross validation menghasilkan akurasi sebesar 94,8%. Hasil penelitian ini dapat disimpulkan bahwa model klasifikasi penyakit stroke yang dikembangkan memiliki performa yang sangat baik. Secara keseluruhan, hasil ini mengindikasikan bahwa model klasifikasi stroke yang dikembangkan sangat andal dan efektif, dengan kemampuan yang sangat baik dalam mendeteksi kasus stroke dan memberikan prediksi yang akurat. Model ini dapat diandalkan sebagai alat bantu dalam diagnosis penyakit stroke yang dapat membantu dalam pengambilan keputusan medis yang lebih baik dan tepat waktu.

Kata kunci: *Stroke, support vector machine, klasifikasi*

1. Pendahuluan

Stroke atau dikenal dengan Cerebrovascular Accident (CVA) merupakan gangguan fungsional yang diakibatkan terganggunya fungsi aliran darah dari pada otak manusia [1]. Menurut World Health Organization (WHO), stroke didefinisikan sebagai pendarahan yang terjadi di otak dalam waktu 24 jam atau kurang dan dapat berkembang cukup cepat sehingga menyebabkan kematian [2][3]. Pecahnya pembuluh darah di otak adalah penyebab masalah sirkulasi darah. Stroke secara tiba-tiba yang ditandai dengan kelemahan bagian tubuh

pada lengan dan kaki, wajah asimetris, atau artikulasi bicara. Gejala ini akhirnya dapat menyebabkan sel-sel dalam tubuh mati [4]. Stroke membawa risiko tinggi pada kerusakan otak, kelumpuhan, gangguan bicara, gangguan penglihatan, hingga kematian. Diseluruh dunia stroke menjadi salah satu masalah yang sangat serius [5]. Penyebab utama kematian dan cacat disebabkan oleh stroke menurut laporan Global Burden of Disease (GDB) tahun 2019. Lebih dari 101 juta orang diseluruh dunia menderita stroke, dengan lebih dari 12,2 juta kasus baru dilaporkan setiap tahunnya, menurut



Lisensi

Lisensi Internasional Creative Commons Attribution-ShareAlike 4.0.

World Stroke Organization (WSO). Diperkirakan 9 juta *stroke* dan 4,5 juta kematian terkait *stroke* terjadi setiap tahunnya. Ditemukan 12,2 juta kasus terkini di dunia pada tahun 2022, dengan 62% kasus terjadi pada individu kurang dari 70 tahun. Secara global, ada 101,4 juta *stroke* serta 6,5 juta kematian yang terkait dengan *stroke* di seluruh dunia [6].

Berdasarkan data Organisasi Kesehatan Dunia (WHO) menunjukkan bahwa *stroke* menyebabkan 7.9% dari semua kematian di Indonesia. Penyakit *stroke* masuk kedalam 10 jenis penyakit yang paling mematikan di Indonesia [7]. Serangan gejala *stroke* yang terkait dengan tekanan darah tinggi dapat menyebabkan kerusakan pada dinding pembuluh darah sehingga menyebabkan dinding pembuluh darah menjadi tidak rata. Biasanya penyakit ini sering diketahui ketika telah parah [8]. Bila ditemui riwayat menderita *stroke* di keluarga, maka akan lebih baik untuk diwaspadai karena berpeluang lebih besar menderita *stroke*. Hal ini menyebabkan penyakit *stroke* harus diwaspadai.

Pencegahan secara dini melalui pemeriksaan dokter perlu dilakukan karena tingginya angka penyakit *stroke*. Pendeteksian risiko penyakit *stroke* mudah ditentukan apabila mengetahui parameter yang lengkap dan terstruktur [9]. Namun terkadang pendeteksian risiko penyakit *stroke* sulit ditentukan jika ada faktor risiko yang kurang jelas dan tidak terstruktur karena dapat menghalangi para tenaga medis saat mendiagnosis pasien [10]. Penanganan yang cepat dapat mengurangi tingkat kerusakan pada otak dan kemungkinan timbulnya komplikasi. Maka perlu dilakukan prediksi seseorang terserang *stroke* atau bukan *stroke*. Salah satu metode untuk memprediksi penyakit *stroke* yakni dengan klasifikasi guna menetapkan penyakit secara pasti. Melalui pemeriksaan yang sesuai dapat memudahkan praktisi kesehatan untuk memutuskan tindakan dengan yang lebih tepat dan dalam waktu yang cepat.

Dunia medis menggunakan sejumlah metode untuk mendiagnosis penyakit *stroke* diantaranya melalui pemeriksaan fisik dan pemeriksaan riwayat medis untuk mengindikasikan penyakit ini. Selain itu juga perlu mengetahui riwayat kesehatan yang mungkin menjadi faktor risiko penyakit *stroke*, evaluasi *neurologis* untuk memeriksa fungsi otak dan sistem saraf [11]. Nilai NIHSS (*National Institutes of Health Stroke Scale*) merupakan alat yang digunakan untuk mengukur tingkat keparahan *stroke*, analisis laboratorium melalui tes darah untuk memeriksa faktor risiko seperti kadar gula darah, dan *Elektrokardiogram* (EKG) untuk memeriksa aktivitas jantung [12].

Memilih pendekatan yang tepat untuk mengidentifikasi penyakit *stroke* sangat diperlukan karena berdampak terhadap hasil yang akan ditampilkan. Peneliti mengimplementasikan metode untuk mendiagnosis penyakit *stroke* dengan algoritma *Support Vector*

Machine (SVM). Dalam diagnostik medis, SVM adalah teknik belajar yang digunakan untuk klasifikasi. SVM menemukan sebuah *hyperplane* yang memiliki kemungkinan fraksi poin terbesar dari kelas yang sama pada bidang yang sama. *Hyperplane* adalah fungsi yang digunakan untuk membedakan antar fitur [13]. SVM mampu menangani data yang kompleks dan menghasilkan klasifikasi yang baik. SVM dapat digunakan dengan data klinis atau *dataset* yang beragam untuk meningkatkan akurasi klasifikasi penyakit *stroke*. Implementasi SVM dalam sistem kesehatan dapat membantu dokter dalam membuat keputusan yang cepat dan tepat dalam penanganan *stroke*. Dengan kemajuan teknologi dan ketersediaan data yang lebih besar, SVM dan metode pembelajaran mesin lainnya memiliki potensi besar untuk merevolusi klasifikasi dan pengobatan *stroke*.

Beberapa penelitian terkait penggunaan algoritma SVM untuk klasifikasi penyakit *stroke* menggunakan *dataset* yang bersumber dari *Kaggle "healthcare-dataset-stroke-data"* dengan jumlah data 5110 dan 12 atribut. Penelitian ini menghasilkan akurasi sebesar 90.1% [14]. Penelitian lain untuk deteksi dini penyakit *stroke* dengan metode SVM menggunakan data hasil rekam medis berisi asal risiko *stroke* terdiri dari umur, gula darah, tekanan darah, dan LDL menghasilkan nilai akurasi sebesar 84% [15]. Penelitian lainnya dilakukan oleh Yuwike (2023) dengan menggunakan cara pengumpulan data dari *website Kaggle*. Data ini bisa digunakan sebagai data *training*, sedangkan untuk data *testing* hanya diambil data hasil atau keluaran. Penelitian ini menghasilkan nilai akurasi sebesar 85% [10]. Penelitian lainnya terkait analisis algoritma SVM dalam klasifikasi penyakit *stroke* yang menggunakan 1000 baris data dan 12 kolom data untuk pengujian menghasilkan akurasi sebesar 77% untuk kernel linier dan 76% untuk kernel *polynomial* [16].

Beberapa penelitian lain dilakukan oleh Nabilla Yolanda terkait klasifikasi penyakit *stroke* menggunakan metode *Naive Bayes* dengan jumlah data sebanyak 200 dan 11 variabel menghasilkan akurasi sebesar 80% [17]. Penelitian lain dilakukan oleh Hendriyansyah terkait penerapan algoritma *Decision Tree* untuk klasifikasi penyakit *stroke* menghasilkan akurasi sebesar 85.81% dari 4982 data dan 11 atribut [18]. Penelitian lain terkait klasifikasi *stroke* menggunakan metode *Random Forest* dari data 5110 dan 12 atribut menghasilkan akurasi sebesar 86.82% [19].

Berdasarkan penelitian sebelumnya dapat disimpulkan bahwa algoritma SVM dapat diterapkan untuk klasifikasi penyakit *stroke* dengan akurasi yang cukup baik. Namun, akurasi yang dihasilkan masih bisa dioptimalkan sehingga penelitian ini bermaksud memodifikasi penelitian terdahulu yang menerapkan metode SVM untuk klasifikasi *stroke* dengan menerapkan *tools Orange* untuk proses *learning* SVM untuk mencari nilai akurasi dengan lebih mudah dan

menghasilkan akurasi yang lebih baik menggunakan *dataset* yang lebih besar. Peningkatan akurasi dapat dilakukan dengan cara pembersihan data untuk menghapus fitur yang tidak relevan atau redundan yang dapat mengganggu performa model, melakukan normalisasi atau standarisasi data untuk memastikan bahwa semua fitur memiliki skala yang sama, optimasi parameter dengan mencoba berbagai kernel dan dilihat performa yang terbaik. Pemilihan metode SVM berdasarkan keunggulan dalam proses klasifikasi data. SVM mampu mengidentifikasi *hyperplane* terpisah yang memaksimalkan margin antara dua kelas yang berbeda. Berdasarkan hal ini, peneliti bermaksud untuk mengolah *dataset* yang sudah didapatkan menggunakan *tool Orange*. Selain SVM, berbagai metode lain telah digunakan dalam penelitian untuk klasifikasi penyakit stroke seperti *K-Nearest Neighbors (KNN)*, *Decision Tree*, *Random Forest*, *Naive Bayes*, dan lainnya.

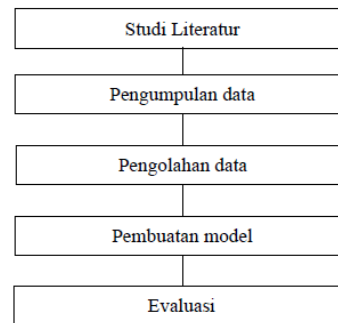
Penelitian ini bertujuan untuk memodifikasi penelitian terdahulu yang menggunakan algoritma SVM untuk klasifikasi penyakit *stroke* menggunakan teknik normalisasi dengan berbagai jenis kernel SVM.

2. Metode Penelitian

Metode dalam penelitian ini menggunakan algoritma SVM untuk klasifikasi penyakit *stroke*. SVM merupakan salah satu algoritma yang sangat efektif untuk tugas klasifikasi, termasuk dalam bidang kesehatan seperti klasifikasi penyakit *stroke*. SVM memiliki kemampuan dapat menangani data berdimensi tinggi yaitu dengan banyak fitur atau variabel yang digunakan dalam prediksi [20][21]. SVM dirancang untuk menemukan *hyperplane* yang optimal untuk memisahkan kelas-kelas data dengan *margin* terbesar.

Banyak penelitian yang telah menunjukkan bahwa SVM dapat mencapai tingkat akurasi yang tinggi dalam klasifikasi medis. Penggunaan SVM untuk klasifikasi penyakit *stroke* didorong oleh kemampuannya untuk menangani data dengan banyak fitur, meminimalkan risiko *overfitting*, dan memberikan performa prediksi yang baik. Dalam tahapannya, SVM diolah menggunakan *tool Orange*. *Orange* merupakan platform visualisasi data dan analisis yang berbasis *open-source*, yang dirancang untuk memudahkan analisis data dan pembelajaran mesin. *Orange* menyediakan antarmuka grafis yang intuitif yang memungkinkan pengguna untuk membangun dan mengevaluasi model klasifikasi tanpa memerlukan keterampilan pemrograman yang mendalam [22]. Penggunaan *Orange* dalam penelitian ini memungkinkan pengguna untuk mengatur alur kerja analisis data tanpa perlu menulis kode. *Orange* mendukung berbagai algoritma pembelajaran mesin, termasuk SVM. *Orange* memungkinkan evaluasi model memberikan berbagai matrix kinerja, seperti akurasi, presisi, *recall*, dan *F1-score*. Memungkinkan proses yang lebih efisien dan hasil yang mudah dipahami,

terutama bagi pengguna yang mungkin tidak memiliki banyak pengalaman dalam pemrograman atau analisis data lanjutan. Sehingga penerapan algoritma SVM diharapkan sesuai dengan kebutuhan pengguna untuk mengklasifikasi penyakit *stroke* dengan baik. Tahap-tahap penelitian ini ditunjukkan pada Gambar 1. Tahap penelitian dimulai dari tahap studi literatur, dilanjutkan dengan tahap pengumpulan data, pengolahan data, pembuatan model, dan di akhir tahapan akan dilakukan evaluasi model.



Gambar 1. Tahap-tahap penelitian

2.1 Studi Literatur

Tahap studi literatur dilakukan untuk mencari, mengumpulkan, mengevaluasi, dan menganalisis informasi bahan pustaka maupun referensi yang selaras pada topik penelitian yang dilaksanakan. Sumber informasi bisa diperoleh dari buku, jurnal ilmiah, artikel, dan sumber-sumber lainnya. Tujuan dari studi literatur untuk memahami apa yang sudah diketahui tentang suatu topik. Studi literatur penting untuk memastikan bahwa penelitian baru dibangun di atas pengetahuan yang ada dan tidak mengulang pekerjaan yang sudah dilakukan, serta membantu dalam merumuskan hipotesis yang lebih tepat dan metodologi yang lebih efektif.

Pada tahap ini, peneliti mengumpulkan penelitian-penelitian terdahulu yang membahas topik *stroke* dan algoritma SVM. Berdasarkan penelitian yang telah ditemukan dan analisis, peneliti mencoba mengisi gap penelitian yang ada. Pada penelitian terdahulu memang sudah ada penelitian yang serupa akan tetapi tingkat akurasinya masih kurang bagus [10][15]. Pada penelitian ini, peneliti mencoba membuat model dengan bantuan *tool Orange* dan diharapkan tingkat akurasinya bisa lebih dari 90%.

2.2 Pengumpulan data

Pada tahap ini dilakukan pengumpulan data yang merujuk pada proses sistematis yang digunakan untuk mengumpulkan data atau informasi yang diperlukan. Data ini dapat bersumber dari berbagai media tergantung pada jenis penelitian yang digunakan. Penulis mengumpulkan dataset *stroke* dari situs *data.world.com* [23]. Pada dataset ini diperoleh data dengan jumlah 40.910 baris dan terdiri dari 11 atribut. 10 atribut akan dijadikan fitur dan 1 atribut dijadikan kelas *stroke* [23].

2.3 Pengolahan data

Tahap ini dilakukan pemeriksaan data dan pembersihan data. Pemeriksaan data bertujuan untuk mengevaluasi dan verifikasi data yang telah dikumpulkan untuk memastikan kualitas, konsistensi, dan keandalannya sebelum data tersebut dianalisis lebih lanjut. Sedangkan pembersihan data bertujuan untuk mengidentifikasi, memperbaiki, menghapus data yang tidak akurat, tidak lengkap, atau duplikat. Proses ini sering melibatkan pemeriksaan data untuk kesalahan input, anomali, dan inkonsistensi. Pembersihan data merupakan tahap untuk memastikan bahwa data yang digunakan dalam penelitian memiliki kualitas yang baik dan dapat diandalkan. Informasi yang digunakan dalam penelitian berdasarkan kriteria data lengkap (tanpa *missing values*), data tidak *redundant*, serta tipe data yang sesuai. Pengolahan dataset dengan melakukan normalisasi data, mencari bobot dan bias, dan mencari akurasi nilai. Normalisasi dilakukan untuk memastikan bahwa tiap atribut pada data tidak memiliki variabel yang dapat mengganggu proses pengolahan data. Normalisasi dilakukan dengan mengubah data kedalam skala atau rentang nilai yang sama agar memudahkan untuk proses perbandingan. Normalisasi menggunakan rumus 1.

$$X_{normalisasi} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

dengan x adalah nilai asli dari fitur, $min(x)$ adalah nilai minimum dari fitur, $max(x)$ merupakan nilai maksimum dari fitur.

2.4 Pembuatan model

Pada tahap ini dilakukan pembuatan model pembelajaran untuk mengklasifikasi penyakit *stroke* dengan menggunakan algoritma SVM. Pembuatan model dengan algoritma SVM untuk klasifikasi penyakit *stroke* melibatkan beberapa langkah utama, mulai dari pemilihan dan persiapan data hingga evaluasi model. Klasifikasi data dapat dilakukan dengan membangun aturan klasifikasi yang menggunakan data *training* disebut sebagai tahapan pembelajaran, dan pengujian digunakan sebagai data *testing*.

Langkah kerja algoritma SVM pada umumnya adalah sebagai berikut: (a). Menentukan *hyperplane* yang merupakan permukaan dalam ruang dimensi tinggi yang membagi ruang tersebut menjadi dua kelas data. Dalam konteks SVM untuk klasifikasi biner, *hyperplane* memisahkan titik data dari dua kelas yang berbeda. Dalam ruang dua dimensi *hyperplane* adalah garis dan dalam ruang tiga dimensi *hyperplane* adalah bidang. Untuk dimensi yang lebih tinggi, *hyperplane* tetap disebut *hyperplane* meskipun sulit divisualisasikan [24]. SVM berusaha menemukan *hyperplane* yang memaksimalkan *margin* yaitu jarak antara *hyperplane* dengan titik data terdekat dari kedua kelas (disebut *support vectors*). *Hyperplane* optimal adalah yang

memiliki *margin* terbesar. Persamaan *hyperplane* dapat ditulis pada rumus 2.

$$w \cdot x + b = 0 \quad (2)$$

Dengan w adalah vektor normal (parameter yang menentukan orientasi *hyperplane*), x adalah vektor fitur dari data, b adalah bias atau *intercept* (menggeser *hyperplane* dari asal koordinat) [25]. (b). Menentukan bobot dan bias. Bobot (*weight*) dan bias (*bias*) digunakan untuk menentukan *hyperplane* yang memisahkan kelas-kelas data. Bias dalam SVM adalah konstanta yang menentukan seberapa jauh batas keputusan dipindahkan dari asalnya. Hal ini memungkinkan SVM untuk menangani kasus di mana garis pemisah tidak melalui titik (0,0) dalam ruang fitur. Bobot dalam SVM adalah vektor yang menentukan arah dan kemiringan dari batas keputusan. Bobot dapat dilihat sebagai koefisien yang mengukur kontribusi masing-masing fitur terhadap keputusan klasifikasi. (c). Melakukan klasifikasi, setelah *hyperplane*, bobot, dan bias ditemukan, SVM akan menggunakan *hyperplane* tersebut untuk klasifikasi data baru. Klasifikasi terdiri dari 2 kelas yaitu *stroke* dan tidak *stroke*.

2.5 Evaluasi

Evaluasi model dalam algoritma SVM untuk klasifikasi penyakit *stroke* sangat penting untuk dilakukan karena dapat digunakan untuk memastikan bahwa model yang dibuat mampu memprediksi dengan akurasi yang tinggi dan dapat diandalkan. Proses evaluasi model menggunakan *tool* Orange untuk klasifikasi penyakit *stroke*. *Tool* Orange adalah perangkat lunak yang memungkinkan analisis data visual dan *machine learning* dengan antarmuka pengguna berbasis grafis. Orange dapat mempermudah proses evaluasi ini dan memungkinkan pengguna untuk memahami hasil dari analisis *machine learning* dengan baik.

Pada proses pengolahan data dengan *tool* Orange yang pertama dilakukan yaitu dengan cara mengimpor *dataset* yang akan digunakan dengan menggunakan *widget file* pada Orange. *Widget file* berfungsi untuk titik awal mengimpor dan memuat *dataset* ke dalam alur kerja analisis data. *Widget* ini merupakan *widget* dasar yang sangat penting karena memungkinkan pengguna untuk mengakses data yang akan dianalisis dan diproses menggunakan berbagai alat dan teknik di Orange. Setelah mengimpor data kemudian ditentukan atribut yang akan dijadikan sebagai *features* dan atribut yang akan dijadikan target pada klasifikasi. Atribut yang dipilih sebagai *features* diantaranya *sex*, *age*, *hypertension*, *heart_disease*, *ever_married*, *work_type*, *residence_type*, *avg_glucose_level*, *BMI*, dan *smoking_status*. Dan yang dijadikan target adalah *stroke*.

Langkah selanjutnya adalah menentukan kernel yang akan digunakan. Pada SVM terdapat 4 kernel yaitu kernel linier, kernel RBF (*Radial Basis Function*), kernel

sigmoid, dan kernel polinomial. Langkah berikutnya dengan memasukkan *widget data table* pada halaman *Orange* yang dihubungkan dengan *widget file*. Fungsi dari *widget data table* untuk memungkinkan pengguna melihat dan menganalisis *dataset* dalam bentuk tabel yang mudah dipahami. Kemudian ditambahkan *widget data sampler* kedalam halaman *Orange* yang dihubungkan dengan *widget file*. Fungsi dari *widget data sampler* untuk mengambil sampel dari *dataset* yang lebih besar. *Widget* ini adalah alat yang berguna untuk mengurangi ukuran *dataset*, melakukan validasi silang (*cross-validation*), atau membagi *dataset* menjadi *subset* untuk keperluan pelatihan dan pengujian model.

Dengan menggunakan *widget data sampler*, pengguna dapat dengan mudah mengelola *dataset* besar, membuat *subset* data yang seimbang, dan memastikan bahwa model dapat dilatih dan diuji secara efektif. Setelah itu ditambahkan *widget test & score* pada halaman *Orange* yang dihubungkan dengan *widget data sampler*. *Widget test & score* adalah alat yang sangat penting untuk mengevaluasi kinerja model pembelajaran *machine learning*. *Widget* ini memungkinkan pengguna untuk menguji model pada data uji dan menghitung berbagai *matrix* evaluasi untuk menilai seberapa baik model tersebut bekerja. Teknik yang dapat digunakan ada beberapa macam diantaranya *cross validation*, *random sampling*, *leave one out*, *test on train data*, *test on test data*.

Dalam proses ini menggunakan 80% data pelatihan dan 20% data pengujian. Kemudian dilakukan tahap evaluasi model. Dalam tahap evaluasi model pada *dataset* yang telah diuji dengan sebagian teknik, didapatkan hasil yaitu:

- Akurasi digunakan untuk membandingkan data sampel yang diperkirakan benar dengan jumlah sampel total [1]. Untuk menghitung akurasi menggunakan rumus 3.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

- Presisi yang merupakan rasio jumlah data perkiraan benar atas jumlah data yang diperkirakan positif. Ini adalah salah satu *matrix* evaluasi yang penting untuk menilai kinerja model klasifikasi, terutama ketika fokus pada keakuratan prediksi pada kelas tertentu. Presisi dihitung melalui rumus 4.

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (4)$$

- *Recall* adalah rasio jumlah data perkiraan benar atas jumlah data yang sebenarnya positif. *Recall* merupakan salah satu *matrix* evaluasi yang penting dalam mengevaluasi kinerja model klasifikasi, terutama ketika fokus pada kemampuan model dalam mendeteksi semua kasus positif yang ada. *Recall* dihitung melalui rumus 5.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (5)$$

- *F1-Score* adalah kombinasi dari akurasi dan presisi. *F1-score* merupakan metrik evaluasi yang menyatukan presisi dan *recall* dalam satu nilai tunggal, memberikan gambaran yang lebih komprehensif tentang kinerja model klasifikasi. *F1-score* berguna ketika ingin memiliki keseimbangan antara presisi dan *recall*, dan memperhitungkan keduanya secara bersamaan. *F1-score* dihitung dengan rumus 6.

$$F1 - score = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (6)$$

Dengan $TP = \text{True Positive}$ dengan arti bahwa nilai prediksi serta nilai aktualnya bernilai positif, $TN = \text{True Negative}$ dengan arti bahwa nilai prediksi serta nilai aktualnya bernilai negatif, $FP = \text{False Positive}$ yang berarti nilai prediksi bernilai positif sedangkan nilai aktualnya *false* dan $FN = \text{False Negative}$ yang berarti nilai prediksi bernilai *false* sedangkan nilai aktualnya positif.

3. Hasil dan Pembahasan

Pada penelitian ini, *dataset* yang digunakan berasal dari situs *data.world.com* [23]. Dalam *dataset* tersebut terdapat atribut yang dijadikan sebagai parameter utama dalam memprediksi kemungkinan pasien mengidap *stroke*. Atribut-atribut tersebut yakni *sex*, *age*, *hypertension*, *heart disease*, *work type*, *ever married*, *avg-glucose level*, *BMI*, *residence type*, serta *smoking status*. Penjelasan atribut dan dekripsi *dataset* dapat dilihat pada Tabel 1.

3.1. Pembuatan model

Tahap awal yang dilakukan yaitu melakukan normalisasi data untuk memastikan bahwa setiap atribut pada data tidak memiliki variabel yang dapat mengganggu proses pengolahan data. Dari *dataset* yang diperoleh dilakukan pembersihan data untuk memastikan apakah data tersebut sudah sesuai dengan kriteria.

Setelah dilakukan proses pembersihan data, hasil *dataset* tersebut dinormalisasi untuk menyetarakan skala atau rentang nilai yang sama agar memudahkan proses pengolahan data. Dimana normalisasi data menggunakan 5 data *testing* dari *dataset* yang sudah ada. Selanjutnya dilakukan proses pencarian bobot dan bias dengan menggunakan *tools Orange* yang menghasilkan nilai bobot untuk setiap fitur dari *sex* sampai *smoking status* bernilai -0,0001 sampai -0,0007 dan menghasilkan nilai bias 0,0002. Data yang sudah melalui tahap normalisasi, sudah memiliki bobot dan bias kemudian diuji dengan algoritma SVM menggunakan *tool Orange*.

Pada pengujian ini menggunakan beberapa teknik dan kernel pada SVM. Algoritma SVM akan melakukan klasifikasi terhadap data *training*, sehingga dapat memperoleh nilai akurasi, presisi, *recall*, dan *F1-score*.

Tabel 1. Deskripsi Dataset

Atribut	Deskripsi
Sex	Jenis kelamin pasien, 0 jenis kelamin perempuan, 1 jenis kelamin laki-laki
Age	Usia pasien
Hypertension	Riwayat hipertensi pasien, 0 tidak mengalami hipertensi, 1 mengalami hipertensi
Heart_disease	Riwayat penyakit jantung pasien, 0 pasien tidak memiliki penyakit jantung, 1 pasien memiliki penyakit jantung
Ever_married	Status perkawinan pasien, 0 pasien yang belum menikah, 1 pasien yang sudah menikah
Work_type	Jenis pekerjaan pasien, 1 <i>self-employed</i> , 2 <i>private</i> , 3 <i>govt_job</i> , 4 <i>children</i>
Residence_type	Wilayah tinggal pasien, 1 pasien yang tinggal di daerah <i>urban</i> , 2 pasien yang tinggal di daerah <i>rural</i>
Avg_glucose_level	Nilai kadar gula darah pasien
Bmi	<i>Body massa index</i> pasien
Smoking_status	Status merokok pasien, 0 pasien tidak merokok, 1 pasien yang merokok
Stroke	Kesimpulan atau target, 0 pasien tidak mengalami <i>stroke</i> , 1 pasien mengalami <i>stroke</i>

3.2. Evaluasi

Pada proses evaluasi data diolah menggunakan *tools Orange* untuk mencari akurasi dari keseluruhan data training yang terdiri dari 40910 data. Hasil pengujian menggunakan *tool Orange* pada Tabel 2.

Tabel 2. Klasifikasi kernel RBF dan teknik *cross validation*

Model	CA	F1	Prec	Recall
SVM	0.948	0.923	0.950	0.948

Tabel 3. Klasifikasi kernel RBF dan teknik *random sampling*

Model	CA	F1	Prec	Recall
SVM	0.948	0.923	0.935	0.948

Tabel 4. Klasifikasi kernel linear dan teknik *cross validation*

Model	CA	F1	Prec	Recall
SVM	0.947	0.922	0.898	0.947

Tabel 5. Klasifikasi kernel linear dan teknik *random sampling*

Model	CA	F1	Prec	Recall
SVM	0.947	0.922	0.898	0.947

Dari hasil beberapa tabel diatas, klasifikasi menggunakan kernel RBF dan teknik *cross validation* memperoleh hasil akurasi sebesar 94.8%, nilai *F1-Score* sebesar 92.3%, nilai presisi sebesar 95.0%, dan nilai *recall* sebesar 94.8%. Klasifikasi yang dihasilkan dari kernel RBF dengan teknik *random sampling* menghasilkan akurasi sebesar 94.8%, nilai *F1-Score* sebesar 92.3%, nilai presisi sebesar 93.5%, dan nilai *recall* sebesar 94.8%. Klasifikasi yang dihasilkan dari kernel linear dengan teknik *cross validation* menghasilkan akurasi sebesar 94.7%, nilai *F1-Score*

sebesar 92.2%, nilai presisi sebesar 89.8%, dan nilai *recall* sebesar 94.7%. Klasifikasi yang dihasilkan dari kernel linear dengan teknik *random sampling* menghasilkan akurasi sebesar 94.7%, nilai *F1-Score* sebesar 92.2%, nilai presisi sebesar 89.8%, dan nilai *recall* sebesar 94.7%. Akurasi dari semua metode yang digunakan berada pada kisaran yang tinggi. Perbedaan akurasi kernel RBF dan kernel linear, baik dengan *cross validation* maupun *random sampling* sangat kecil hanya 0.1%. Nilai *F1-score* juga menunjukkan performa yang konsisten dengan perbedaan yang sangat kecil. Kernel RBF sedikit lebih unggul dibandingkan kernel linear. Presisi dari kernel RBF dengan *cross validation* lebih tinggi dibandingkan dengan kernel RBF dengan *random sampling* dan kernel linear dengan kedua teknik. Presisi yang lebih tinggi menunjukkan bahwa model kernel RBF dengan *cross validation* lebih baik dalam mengidentifikasi kasus positif *stroke* dengan benar. Recall konsisten di 94.8% untuk kernel RBF dan 94.7% untuk kernel linear, menunjukkan bahwa kedua model tersebut memiliki kemampuan yang baik dalam mengidentifikasi hampir semua kasus positif *stroke*. Kernel RBF sedikit lebih unggul dibandingkan kernel linear dalam hal presisi dan *F1-score*, walaupun perbedaannya sangat kecil. Hal ini menunjukkan bahwa kernel RBF dapat menangani kompleksitas data dengan lebih baik. Teknik *cross validation* menghasilkan presisi yang lebih tinggi untuk kernel RBF dibandingkan dengan teknik *random sampling*. Hal ini menunjukkan bahwa *cross validation* lebih efektif dalam memanfaatkan data secara keseluruhan untuk pelatihan dan evaluasi, sehingga memberikan estimasi performa model yang lebih stabil dan akurat. Performa yang didapatkan dari pengujian algoritma SVM menggunakan kernel RBF dengan teknik *cross validation* memperoleh hasil akurasi yang lebih baik sebesar 94.8%, nilai *F1-Score* sebesar 92.3%, nilai presisi sebesar 95.0%, dan nilai *recall* sebesar 94.8%. Angka tersebut menunjukkan bahwa algoritma SVM memiliki nilai akurasi yang baik dalam mengklasifikasi penyakit *stroke*.

4. Kesimpulan

Penelitian menggunakan algoritma SVM untuk mengklasifikasi penyakit *stroke* telah berhasil dilakukan dengan menggunakan jumlah data training yang terdiri dari 40910 data dan 10 atribut meliputi *sex*, *age*, *hypertension*, *heart disease*, *work type*, *ever married*, *avg-glucose level*, BMI, *residence type*, serta *smoking status*. Dalam proses pengolahan data menggunakan *tool Orange* untuk mempermudah dan mempercepat proses pengolahan data sehingga mendapatkan hasil akurasi yang baik untuk klasifikasi penyakit *stroke*.

Setelah model SVM untuk klasifikasi penyakit *stroke* terbangun, metode penilaian yang digunakan adalah akurasi, *F1-Score*, presisi, dan *recall*. *Orange* berhasil

mengklasifikasi penyakit *stroke* dengan baik menggunakan kernel RBF dengan teknik *cross validation* memperoleh hasil akurasi 94.8%, nilai *F1-Score* sebesar 92.3%, nilai presisi sebesar 95.0%, dan nilai *recall* sebesar 94.8%.

Berdasarkan hasil penelitian ini dapat disimpulkan bahwa model klasifikasi penyakit *stroke* yang dikembangkan memiliki performa yang sangat baik. Secara keseluruhan, hasil ini mengindikasikan bahwa model klasifikasi penyakit *stroke* yang dikembangkan sangat andal dan efektif, dengan kemampuan yang sangat baik dalam mendeteksi kasus *stroke* dan memberikan prediksi yang akurat. Model ini dapat diandalkan sebagai alat bantu dalam diagnosis penyakit *stroke* yang dapat membantu dalam pengambilan keputusan medis yang lebih baik dan tepat waktu.

Ucapan Terimakasih

Terimakasih kepada para dosen Institut Teknologi, Sains, dan Kesehatan Rumah Sakit dr. Soepraoen Kesdam V/BRW yang telah membantu dan membimbing hingga selesainya penelitian ini.

Daftar Pustaka

- [1] Patmawati, "Prediksi Penyakit Stroke Menggunakan Support Vector Machine (Svm)," *Bull. Netw. Eng. Informatics*, vol. 1, no. 1, pp. 12–17, 2023, [Online]. Available: <https://bufnets.tech/index.php/bufnets/article/view/5/11>
- [2] N. Permatasari, "Perbandingan Stroke Non Hemoragik dengan Gangguan Motorik Pasien Memiliki Faktor Resiko Diabetes Melitus dan Hipertensi," *J. Ilm. Kesehat. Sandi Husada*, vol. 11, no. 1, pp. 298–304, 2020, doi: 10.35816/jiskh.v11i1.273.
- [3] A. Byna and M. Basit, "Penerapan Metode Adaboost Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan Algoritma Naïve Bayes," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 9, no. 3, pp. 407–411, 2020, doi: 10.32736/sisfokom.v9i3.1023.
- [4] R. L. Smanjuntak, "Komparasi Algoritma KNN dan SVM dalam Memprediksi Penyakit Stroke," vol. 3, no. 3, pp. 60–74, 2023.
- [5] R. Adriadi, R. Tribuana, and A. W. Rahmazah, "Edukasi Bahaya Stroke Untuk Meningkatkan Kualitas Hidup Masyarakat Di Desa Bumi Ayu Rt 17 Rw 04 Kecamatan Selebar Kota Bengkulu," *J. Ilm. Mhs. Kuliah Kerja Nyata*, vol. 2, no. 1, pp. 123–126, 2022, doi: 10.36085/jimakukerta.v2i1.3335.
- [6] D. Mualfah, W. Fadila, and R. Firdaus, "Teknik SMOTE untuk Mengatasi Imbalance Data pada Deteksi Penyakit Stroke Menggunakan Algoritma Random Forest," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 3, no. 2, pp. 107–113, 2022, doi: 10.37859/coscitech.v3i2.3912.
- [7] E. R. Pambudi, Sriyanto, and Firmansyah, "Klasifikasi Penyakit Stroke Menggunakan Algoritma Decision Tree C.45," *Ijccs*, vol. 16, No.02, no. x, pp. 221–226, 2022.
- [8] I. Iqbal and Z. Zahrah, "Sistem Pakar Diagnosa Gejala Awal Penyakit Stroke Dengan Menggunakan Metode Fuzzy," *J. TIKA*, vol. 7, no. 3, pp. 235–242, 2022, doi: 10.51179/tika.v7i3.1538.
- [9] N. A. Y. S. Adilla Laela Tusifaiyah, "Penerapan Metode Forward Chaining Untuk Diagnosa Penyakit Penyebab Stroke," *Infos J.*, vol. 14, no. 1, p. 97, 2022, [Online]. Available: www.nusamandiri.ac.id
- [10] Y. Ayuningtyas and I. Made Suartana, "Klasifikasi Penyakit Stroke Menggunakan Support Vector Machine (SVM) dan Particle Swarm Optimization (PSO)," vol. 04, no. 2022, pp. 452–457, 2023.
- [11] Y. P. Diwanto, "Jurnal Abdimas Saintika Jurnal Abdimas Saintika," 2020, doi: <http://dx.doi.org/10.30633/jas.v1i1.483>.
- [12] Z. M. Razdiq and Y. Imran, "Hubungan antara tekanan darah dengan keparahan stroke menggunakan National Institute Health Stroke Scale," *J. Biomedika dan Kesehat.*, vol. 3, no. 1, pp. 15–20, 2020, doi: 10.18051/jbiomedkes.2020.v3.15-20.
- [13] H. S. W. Hovi, A. Id Hadiana, and F. Rakmat Umbara, "Prediksi Penyakit Diabetes Menggunakan Algoritma Support Vector Machine (SVM)," *Informatics Digit. Expert*, vol. 4, no. 1, pp. 40–45, 2022, doi: 10.36423/index.v4i1.895.
- [14] S. H. Sinaga, A. A. M. Duha, and J. Banjarnahor, "Analisis Prediksi Deteksi Stroke Dengan Pendekatan Eda Dan Perbandingan Algoritma Machine Learning," *J. Ilm. Betrik*, vol. 14, no. 02 AGUSTUS, pp. 355–367, 2023, doi: <https://doi.org/10.36050/betrik.v14i02%20AGUSTUS.120>.
- [15] B. P. Tomasouw and F. Y. Rumlawang, "Penerapan Metode SVM Untuk Deteksi Dini Penyakit Stroke (Studi Kasus : RSUD Dr. H. Ishak Umarella Maluku Tengah dan RS Sumber Hidup-GPM)," *Tensor Pure Appl. Math. J.*, vol. 4, no. 1, pp. 37–44, 2023, doi: 10.30598/tensorvol4iss1pp37-44.
- [16] K. R. Sulaeman, C. Setianingsih, and R. E. Saputra, "Analisis Algoritma Support Vector Machine Dalam Klasifikasi Penyakit Stroke," *e-Proceeding Eng.*, vol. 9, no. 3, pp. 922–928, 2022.
- [17] Y. N. Paramitha, A. Nuryaman, A. Faisol, E. Setiawan, and D. E. Nurvazly, "Klasifikasi Penyakit Stroke Menggunakan Metode Naïve Bayes," *J. Siger Mat.*, vol. 04, no. 01, pp. 11–16, 2023, doi: <https://doi.org/10.23960/jsm.v4i1.9236>.
- [18] A. Irma Purnamasari and T. Suprapti, "Penerapan Algoritma Decision Tree Dalam Klasifikasi Penyakit Stroke Otak," *J. Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 3038–3043, 2024, doi: <https://doi.org/10.36040/jati.v8i3.9602>.
- [19] M. F. Banjar, I. Irawati, F. Umar, and L. N. Hayati, "Analysis of Stroke Classification Using Random Forest Method," *Ilk. J. Ilm.*, vol. 14, no. 3, pp. 186–193, 2022, doi: 10.33096/ilkom.v14i3.1252.186-193.
- [20] Syamsiah and A. Darwaman, "Analisa Particle Swarm Optimization Terhadap Keputusan," *Semin. Nas. Ris. dan Teknol. (SEMNAS RISTEK)*, pp. 143–148, 2020, doi: <https://doi.org/10.30998/semnasristek.v4i1.2455>.
- [21] D. Irawan, E. B. Perkasa, Y. Yurindra, D. Wahyuningsih, and E. Helmut, "Perbandingan Klasiifikasi SMS Berbasis Support Vector Machine, Naive Bayes Classifier, Random Forest dan Bagging Classifier," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 10, no. 3, pp. 432–437, 2021, doi: 10.32736/sisfokom.v10i3.1302.
- [22] E. Mardiani *et al.*, "Membandingkan Algoritma Data Mining Dengan Tools Orange untuk Social Economy," *Digit. Transform. Technol.*, vol. 3, no. 2, pp. 686–693, 2023, doi: 10.47709/digitech.v3i2.3256.
- [23] gymprathap, "Hypertension-Stroke-Diabetes-Dataset." [Online]. Available: <https://data.world/gymprathap/hypertension-stroke-diabetes-dataset>
- [24] D. Diana Dewi, N. Qisthi, S. S. S. Lestari, and Z. H. S. Putri, "Perbandingan Metode Neural Network Dan Support Vector Machine Dalam Klasifikasi Diagnosa Penyakit Diabetes," *Cerdika J. Ilm. Indones.*, vol. 3, no. 09, pp. 828–839, 2023, doi: 10.59141/cerdika.v3i09.662.
- [25] S. R. K. W. Tommy Rustandi, D. Suhaedi, and Y. Pemasari, "Pemetaan Hyperplane Pada Support Vector Machine," *Bandung Conf. Ser. Math.*, vol. 3, no. 2, pp. 109–119, 2023, doi: 10.29313/bcsm.v3i2.8187.