KLASIFIKASI RISIKO STROKE BERDASARKAN FAKTOR MEDIS DAN GAYA HIDUP MENGGUNAKAN *RANDOM FOREST* DAN SMOTE

SKRIPSI

Oleh :
AN NISA' PUJA KARIMAH ATTAMIMI
NIM. 210605110078

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2025

KLASIFIKASI RISIKO STROKE BERDASARKAN FAKTOR MEDIS DAN GAYA HIDUP MENGGUNAKAN *RANDOM FOREST* DAN SMOTE

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
AN NISA' PUJA KARIMAH ATTAMIMI
NIM. 210605110078

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2025

HALAMAN PERSETUJUAN

KLASIFIKASI RISIKO STROKE BERDASARKAN FAKTOR MEDIS DAN GAYA HIDUP MENGGUNAKAN *RANDOM FOREST* DAN SMOTE

SKRIPSI

Oleh :
AN NISA'PUJA KARIMAH ATTAMIMI
NIM. 210605110078

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 13 Oktober 2025

Pembimbing I,	Pembimbing II,
Prof. Dr. Suhartono, S.Si, M.Kom	Fajar Rohman Hariri, M.Kom
NIP. 19680519 200312 1 001	NIP. 19890515 201801 1 001

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

KLASIFIKASI RISIKO STROKE BERDASARKAN FAKTOR MEDIS DAN GAYA HIDUP MENGGUNAKAN *RANDOM FOREST* DAN SMOTE

SKRIPSI

Oleh :
AN NISA' PUJA KARIMAH ATTAMIMI
NIM. 210605110078

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 28 November 2025

Susunan Dewan Penguji

Ketua Penguji	: Prof. Dr. Muhammad Faisal, M.T NIP. 19740510 200501 1 007	( )
Anggota Penguji I	: Syahiduz Zaman, M.Kom NIP. 19700502 200501 1 005	( )
Anggota Penguji II	: Prof. Dr. Suhartono, M.Kom NIP. 19680519 200312 1 001	( )
Anggota Penguji III	: Fajar Rohman Hariri, M.Kom NIP. 19890515 201801 1 001	( )

Mengetahui dan Mengesahkan,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	: An Nisa' Puja Karimah Attamimi
NIM	: 210605110078
Fakultas / Program Studi	: Sains dan Teknologi / Teknik Informatika
Judul Skripsi	: Klasifikasi Stroke Berdasarkan Faktor Medis dan Gaya Hidup Menggunakan *Random Forest* dan SMOTE

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 28 November 2025
Yang membuat pernyataan,

An Nisa' Puja Karimah Attamimi
NIM.210605110078

MOTTO

Life what you live

HALAMAN PERSEMBAHAN

Dengan penuh rasa Syukur kepada Allah SWT. atas nikmat yang selalu Dia berikan, penulis mempersembahkan karya tulisan ini kepada:

Ibu tercinta, yang doanya selalu dicurahkan tiada henti dan dukungan untuk kesuksesan selalu penulis.

Kakak-kakak tercinta, yang dukungannya selalu diberikan kepada penulis.

Dosen pembimbing, yang membantu dan memberikan ilmu yang berharga.

Teman-teman tercinta, yang membantu dan mendukung yang diberikan.

Diri sendiri, yang telah bekerjasama bertahan berjuang hingga akhir penulisan karya ini.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji Syukur kepada Allah SWT. atas nikmat dan hidayah yang telah diberikan, sehingga dapat terselesaikan karya tulisan skripsi ini dengan judul "Klasifikasi Risiko Stroke Berdasarkan Faktor Medis dan Gaya Hidup Menggunakan *Random Forest* dan SMOTE". Sholawat serta Salam senantiasa dihaturkan kepada Rasulullah SAW. yang telah membawa kita dari jalan jahiliyah yang gelep menuju jalan Islamiyah yang terang.

Skripsi ini disusun dan diajukan untuk memenuhi salah satu syarat guna memperoleh gelar sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Skripsi ini disusun tak lepas dari bantuan dari beberapa pihak, oleh karena itu, penulis ingin mengucapkan terima kasih sebesar-besarnya pada kesempatan kali ini kepada:

1. Rektor Universitas Maulana Malik Ibrahim Malang Prof. Dr. Hj. Ilfi Nur Diana, M.Si.
2. Dekan Fakultas Sains dan Teknologi, Dr. H. Agus Mulyono, M.Si.
3. Ketua Program Studi Teknik Informatika, bapak Supriyono, M.Kom.
4. Prof. Dr. Suhartono, S.Si, M.Kom, sebagai dosen pembimbing I, atas ilmu dan bimbingan yang telah diberikan kepada penulis dari awal hingga akhir penulisan.

5. Bapak Fajar Rohman Hariri, M.Kom, sebagai dosen pembimbing II, atas ilmu integritas islam dan bimbingannya dari awal hingga akhir penulisan skripsi ini kepada penulis.
6. Prof. Dr. Muhammad Faisal, M.T dan bapak Syahiduz Zaman, M.Kom, sebagai dosen penguji I dan II, atas waktu untuk menguji serta ilmu dan masukan sehingga tercipta skripsi ini.
7. Segenap Dosen, Admin, dan jajaran Staff Teknik Informatika yang telah memberikan ilmu dan dukungan yang diberikan kepada penulis, sehingga penulis menyelesaikan perjalanan studi selama 4 tahun di Teknik Informatika Universitas Islam Maulana Malik Ibrahim Malang dengan baik.
8. Ibu tercinta, Nur Thohuroh, atas doa, dukungan, dan semangat yang selalu diberikan kepada penulis. Terima kasih atas waktu dan dedikasi yang dicurahkan untuk merawat dan membesarkan penulis dan masih terlalu banyak kata terima kasih yang tidak bisa diutarakan pada kata pengantar ini.
9. Kakak-kakak tercinta, Bayudh, Afam, dan Kiki, atas dukungan dan dorongan kepada penulis untuk menyelesaikan skripsi ini. Terima kasih telah menjadi saudara yang baik dan loyal.
10. Keluarga besar, atas dukungan yang diberikan kepada penulis.
11. Teman sepembimbing yang telah mengakhiri perjalanan skripsi ini lebih dahulu, Ummi, Otul, dan Sita, atas dukungan dan bantuan dari awal skripsi hingga akhir.

12. Teman-teman Angkatan 2021 “ASTER”, atas kebersamaan selama 4 tahun perjalanan studi.
13. Terakhir, diri sendiri, atas kerjasamanya untuk menyelesaikan skripsi ini. Terima kasih telah berjuang dan bertahan sampai akhir dan tidak menyerah. Terima kasih atas dedikasinya untuk menyelesaikan perjalanan studi ini meskipun banyak rintangan dan keluh kesah. Dan masih banyak kata terima kasih lain untuk diri sendiri yang tidak bisa ditulis di sini.

Penulis menyadari bahwa skripsi ini tidak akan selesai tanpa dukungan dari banyak pihak dan masih jauh dari kata sempurna. Akhir kata, semoga skripsi ini bisa bermanfaat bagi penulis dan pembaca.

Wassalamu’alaikum Warahmatullahi Wabarakatuh.

Malang, 28 November 2025

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Attamimi, An Nisa' Puja Karimah. 2025. **Klasifikasi Risiko Stroke Berdasarkan Faktor Medis dan Gaya Hidup menggunakan *Random Forest* dan SMOTE.** Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Prof. Dr. Suhartono, S.Si, M.Kom (II) Fajar Rohman Hariri, M.Kom.

.

Kata kunci: Strok, Klasifikasi, *Random Forest*, SMOTE.

Stroke merupakan penyakit yang menyerang saraf dan otak sehingga dapat menyebabkan kelumpuhan bahkan meninggal dan menjadi salah satu penyakit mematikan nomer 2 setelah penyakit jantung di dunia. Penelitian ini bertujuan untuk mengklasifikasi risiko stroke berdasarkan faktor medis dan gaya hidup dengan menggunakan model *Random Forest* dan teknik keseimbangan data SMOTE. Dilakukan dengan membagi data secara *Hold-out* dan *K-Fold cross validation* dan tuning parameter sebelum mengklasifikasi data publik yang tidak seimbang dengan model *Random Forest* tanpa SMOTE, menggunakan Borderline-SMOTE1 dan menggunakan Borderline-SMOTE2 yang kemudian dilakukan analisa performa model. Hasil rata-rata akurasi dari klasifikasi menggunakan *Random Forest* dengan tuning *GridSearchCV* tanpa SMOTE sebesar 94% menggunakan pembagian data *Hold-out* dan 96% menggunakan pembagian data *K-Fold cross validation*. Hasil rata-rata akurasi dengan Borderline-SMOTE1 sebesar 91% menggunakan pembagian data *Hold-out* dan 96% menggunakan pembagian data *K-Fold cross validation*. Hasil rata-rata akurasi dengan Borderline-SMOTE2 sebesar 91% menggunakan pembagian data *Hold-out* dan 96% menggunakan pembagian data *K-Fold cross validation*. Dengan tuning *RandomizedSearchCV* klasifikasi *Random Forest* tanpa SMOTE sebesar 94% menggunakan pembagian data *Hold-out* dan 96% menggunakan pembagian data *K-Fold cross validation*. Dengan Borderline-SMOTE1 sebesar 91% menggunakan pembagian data *Hold-out* dan 96% menggunakan pembagian data *K-Fold cross validation*. Dengan Borderline-SMOTE2 sebesar 92% menggunakan pembagian data *Hold-out* dan 96% menggunakan pembagian data *K-Fold cross validation*.

ABSTRACT

Attamimi, An Nisa' Puja Karimah. 2025. **Classification of Stroke Risks Based on Medical Factors and Life Style Using the Random Forest and SMOTE**. Thesis. Informatics Engineering Study Program Faculty of Science and Technology Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisor: (I) Prof. Dr. Suhartono, S.Si, M.Kom (II) Fajar Rohman Hariri, M.Kom.

Keywords: Stroke, Classification, Random Forest, SMOTE.

Stroke is a disease that attacks the nerves and brain, causing paralysis and even death, making it the second deadliest disease after heart disease worldwide. This study aims to classify stroke risk based on medical and lifestyle factors using the *Random Forest* model and the SMOTE data balancing technique. This was done by dividing the data into *Hold-out* and *K-Fold cross validation* and tuning parameters before classifying unbalanced public data with the *Random Forest* model without SMOTE, using Borderline-SMOTE1 and using Borderline-SMOTE2, followed by model performance analysis. The average accuracy of classification using *Random Forest* with *GridSearchCV* tuning without SMOTE was 94% using *Hold-out* data division and 96% using *K-Fold cross validation* data division. The average accuracy with Borderline-SMOTE1 was 91% using *Hold-out* data division and 96% using *K-Fold cross validation* data division. The average accuracy results with Borderline-SMOTE2 were 91% using *Hold-out* data division and 96% using *K-Fold cross validation* data division. With *RandomizedSearchCV* tuning, *Random Forest* classification without SMOTE was 94% using *Hold-out* data division and 96% using *K-Fold cross validation* data division. With Borderline-SMOTE1, the accuracy was 91% using *Hold-out* data division and 96% using *K-Fold cross validation* data division. With Borderline-SMOTE2, the accuracy was 92% using *Hold-out* data division and 96% using *K-Fold cross validation* data division.

الملخص

التميمي، النسأء فوجا كاريمة 2025. **تصنيف مخاطر الإصابة بالسكتة الدماغية بناءً على العوامل الطبية ونمط الحياة باستخدام Random Forest و SMOTE**. أطروحة. قسم هندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية، مالانج. المشرفون: (I) الأستاذ الدكتور سوهارتونو، بكالوريوس، ماجستير (II) فاجار رحمن حريري، ماجستير

الكلمات المفتاحية: السكتة الدماغية، التصنيف ،SMOTE, Random Forest.

السكتة الدماغية هي مرض يصيب الأعصاب والدماغ وقد يؤدي إلى الشلل أو حتى الوفاة، وتُعدّ ثاني أكثر الأمراض فتكًا في العالم بعد أمراض القلب. يهدف هذا البحث إلى تصنيف مخاطر السكتة الدماغية بناءً على العوامل الطبية ونمط الحياة باستخدام نموذج *Random Forest* وتقنية الموازنة.SMOTE تم إجراء البحث من خلال تقسيم البيانات باستخدام طريقتي *Holdout* و *K-Fold Cross Validation*، بالإضافة إلى ضبط المعاملات قبل تصنيف البيانات العامة غير المتوازنة باستخدام نموذج *Random Forest*بدونSMOTE ، ومعBorderline-SMOTE1 ، ومعBorderline-SMOTE2 ، ثم تحليل أداء النموذج.بلغ متوسط دقة التصنيف باستخدام *Random Forest* مع ضبط المعاملات بواسطة *GridSearchCV* وبدون SMOTEنسبة 94% باستخدام *Holdout* و96% باستخدام *K-Fold Cross Validation*. أما باستخدام Borderline-SMOTE1فقد بلغت الدقة 91% مع *Holdout* و96% مع *K-Fold Cross Validation*. كما بلغت الدقة باستخدام Borderline-SMOTE2 نسبة 91% مع *Holdout* و96% مع *K-Fold Cross Validation*. وعند استخدام الضبط بواسطة*RandomizedSearchCV* ، بلغ أداء نموذج Random Forest بدون SMOTE نسبة 94% باستخدام *Holdout* و96% باستخدام *K-Fold Cross Validation*. أما باستخدام Borderline-SMOTE1 فقد بلغت الدقة 91% مع *Holdout* و96% مع *K-Fold Cross Validation*، في حين بلغت باستخدام-Borderline SMOTE2نسبة 92% مع *Holdout* و96% مع *K-Fold Cross Validation*.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Stroke merupakan penyakit yang terjadi ketika arteri di otak tersumbat atau bocor sehingga menyebabkan sel-sel otak kekurangan oksigen dan mulai mati dalam beberapa menit. Dampak dari penyakit ini bisa meyebabkan kematian dan juga penderita akan mengalami disabilitas seumur hidup. Stroke merupakan salah satu penyakit yang paling banyak menyebabkan kematian serta kecacatan yang serius dan saat ini menempati posisi nomer 2 sebagai penyakit paling mematikan di dunia. Berdasarkan data Organisasi Kesehatan Dunia (WHO), stroke menyumbang sekitar 11% dari total kematian global setiap tahunnya. Di Indonesia, stroke menjadi penyebab kematian tertinggi di antara penyakit tidak menular lainnya (Andriani dkk., 2024).

Menteri Kesehatan RI, Budi Gunadi Sakidin, mengatakan bahwa sebanyak 300 ribu orang per tahun meninggal dunia akibat Stroke dan sebanyak 900 ribu orang meninggal dunia akibat Stroke pada saat masa pandemi COVID-19 (Rahmadania, 2024). Prevalensi Stroke di Indonesia sendiri meningkat 56% selama 5 tahun dari 7 per 1000 penduduk pada tahun 2013, menjadi 10,9 per 1000 penduduk pada tahun 2018 (Kemenkes, 2023). Berdasarkan Global Stroke Fact Sheet 2022 yang dirilis oleh World Stroke Organization (WSO) (Feigin dkk., 2022), risiko seumur hidup yang diakibatkan oleh Stroke telah meningkat sebesar 50% selama 17 tahun terakhir. Dari tahun 1990 hingga tahun 2019, telah terjadi peningkatan penyakit Stroke sebesar 70%, peningkatan kematian akibat Stroke

sebesar 43%, peningkatan prevalensi Stroke sebesar 102%, dan peningkatan sebesar 143% dalam Disability Adjusted Life Years (DALY) (Feigin dkk., 2022). Stroke sendiri dapat dicegah dengan melakukan beberapa langkah salah satunya yaitu mendeteksi secara dini risiko stroke. Untuk melakukan deteksi dini risiko stroke kita perlu mengetahui apa saja faktor yang dapat memengaruhi terjadinya stroke.

Faktor risiko penyebab terjadinya stroke bisa terjadi karena faktor risiko medis dan gaya hidup. Faktor risiko medis yang bisa membuat seseorang terkena stroke antara lain tekanan darah yang tinggi, riwayat keluarga stroke, trauma otak, dan aneurisma otak. Tekanan darah tinggi atau hipertensi dapat menyebabkan stroke dengan cara memicu penyempitan, kebocoran, pecahnya, atau penyumbatan pembuluh darah dan mengganggu aliran darah yang membawa oksigen dan nutrisi ke otak. Kadar kolesterol yang tinggi juga menyebabkan terjadinya stroke dengan cara penumpukan dan pengerasan pembuluh darah oleh kolesterol jahat sehingga terjadilah penyumbatan pembuluh darah (Junaidi, 2011). Selain faktor risiko medis, faktor gaya hidup juga dapat mempengaruhi penyakit stroke.

Gaya hidup yang tidak sehat seperti merokok, mengonsumsi makanan cepat saji secara berlebihan, meminum alkohol, dan tidak pernah berolahraga. Mengonsumsi makanan cepat saji secara berlebihan yang tinggi akan kalori, lemak, dan gula, meminum alkohol, dan tidak pernah berolahraga atau beraktivitas menyebabkan naiknya BMI(Body Mass Index) sehingga terjadilah obesitas (Azahra dkk., 2025). Yang mana obesitas sendiri merupakan salah satu faktor yang menyebabkan risiko stroke. Selain itu merokok juga dapat menyebabkan penyakit

jantung dan penyakit jantung sendiri juga faktor risiko penyakit stroke (Balatif & Sukma, 2021). Dengan membiasakan gaya hidup yang tidak sehat seperti itu sejak masih muda dan tidak pernah memperbaiki gaya hidup, mungkin akan terbawa sampai masa tua, sehingga nanti pada masa tua penyakit penyakit akan menyerang salah satunya stroke.

Dalam Al-Qur'an, Allah Subhaanahu Wa Ta'ala telah memerintahkan kita untuk selalu hidup sehat dengan memakan makanan yang halal dan baik seperti yang telah tertera pada Q.S. Al-A'raf (7) ayat 31 yang berbunyi:

يَٰبَنِيٓ ءَادَمَ خُذُواْ زِينَتَكُمۡ عِندَ كُلِّ مَسۡجِدٖ وَكُلُواْ وَٱشۡرَبُواْ وَلَا تُسۡرِفُوٓاْۚ إِنَّهُۥ لَا يُحِبُّ ٱلۡمُسۡرِفِينَ

*"Hai anak Adam, pakailah pakaianmu yang indah di setiap (memasuki) masjid, makan dan minumlah, dan janganlah berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang-orang yang berlebih-lebihan." (Al-a'raf [7]: 3*1*).*

Pada penggalan ayat وَكُلُواْ وَٱشۡرَبُواْ وَلَا تُسۡرِفُوٓاْ menurut Markaz Tafsir Riyadh dalam buku tafsir Al-Mukhtashar (Markaz Tafsir Riyadh, 1996) dijelaskan bahwa Allah memerintahkan kita untuk selalu memakan dan meminum apa apa yang baik dan halal dan kita tidak boleh mengonsumsi secara berlebihan. Seperti yang kita tau salah satu faktor gaya hidup penyebab risiko stroke adalah makan makanan yang tidak jelas kandungannya secara berlebihan meskipun itu halal sekalipun. Mengonsumsi makanan tersebut secara berlebihan dapat menyebabkan kita mengalami faktor faktor medis penyebab risiko stroke. Meminum alkohol juga menjadi salah satu faktor gaya hidup yang menyebabkan risiko stroke dan seperti yang kita tau dalam islam minuman alkohol merupakan minuman yang diharamkan. Karena itulah kita harus menjaga pola makan kita.

Selain menjaga pola makan dan memakan serta meminum yang baik dan halal, dalam ajaran islam kita juga dianjurkan untuk menjaga tubuh kita agar tetap sehat dengan melakukan olahraga. Sebagaimana penggalan hadits yang diriwayatkan oleh Muslim yaitu:

اَلْمُؤْمِنُ الْقَوِيُّ خَيْرٌ وَأَحَبُّ إِلَى اللهِ مِنَ الْمُؤْمِنِ الضَّعِيْفِ

"Orang beriman yang kuat lebih baik dan lebih dicintai Allah daripada orang beriman yang lemah." (HR. Muslim, no. 2664).

Pada penggalan hadits diatas dikatakan bahwa Allah menyukai orang beriman yang kuat. Tubuh kuat dapat diperoleh dari olahraga dan menjaga tubuh kita tetap sehat luar dalam. Seperti yang kita tahu olahraga memiliki banyak sekali manfaat seperti menjaga fisik agar tetap kuat dan membuat fisik kita semakin kuat. Selain itu olahraga juga bisa menjaga jantung kita agar tetap sehat, menurunkan risiko diabetes dan obesitas, menurunkan kolesterol, menjaga tekanan darah kita agar tetap stabil, serta meningkatkan fungsi otak kita. Sebagaimana yang kita tahu faktor-faktor di atas merupaka faktor yang dapat menyebabkan terjadinya stroke, maka dari itu perlu dilakukan olahraga agar menurunkan risiko penyakit stroke pada tubuh kita.

Dengan adanya faktor-faktor tersebut dan melakukan klasifikasi kita bisa mengetahui bagaimana data tersebut termasuk ke dalam kategori stroke dan tidak stroke berdasarkan fitur-fitur dari faktor penyebab stroke. Sehingga dengan adanya klasifikasi ini bisa mempermudah untuk menentukan apa kita berisiko terkena stroke. Klasifikasi merupakan salah satu metode *data mining* yang dilakukan dengan mengelompokkan data ke dalam kategori kelas yang telah ditentukan

berdasarkan pola tertentu (J. Han dkk., 2012). Klasifikasi risiko stroke ini dilakukan dengan menggunakan model *machine learning* yang dilatih dengan data yang telah memiliki label kelas stroke dan tidak stroke.

Dalam klasifikasi ini dilakukan dengan menggunakan algoritma *Random Forest*. *Random Forest* merupakan salah saltu algoritma model *machine learning* dengan berbasis ensamble learning yang menggabungkan banyak pohon keputusan untuk mengambil keputusan (Setiawan dkk., 2024). Dalam memproses data adakalanya data yang digunakan tidak seimbang kelasnya sehingga yang terjadi apabila tidak diatasi model mengalami overfitting dan dapat memengaruhi hasil akhir klasifikasi. Untuk mengatasi data yang tidak seimbang kelasnya dapat dilakukan dengan metode *oversampling* yaitu SMOTE. SMOTE (*Synthetic Minority Over-sampling Technique*) adalah teknik yang digunakan untuk mengatasi ketidakseimbangan kelas yang ada pada data dengan menggunakan metode oversampling (Arifiyanti & Wahyuni, 2020).

Firda (Putri, 2024) melakukan penelitian dengan meneliti pengaruh penanganan ketidaseimbangan data stroke terhadap *Random Forest* yang mana pada penelitian ini metode penyeimbang data menggunakan beberapa metode salah satunya yaitu SMOTE. Pada penelitian ini metode SMOTE didapat hasil akurasi sebesar 87% dan merupakan metode *balancing* dengan performa paling baik diantara metode *balancing* lain yang digunakan pada penelitian ini.

Pada penelitian lain yang dilakukan oleh Fitri dan Reny (Fitri Handayani & Reny Medikawati Taufiq, 2024) yang membandingkan beberapa algoritma dengan teknik SMOTE untuk klasifikasi stroke. Pada penelitian ini hasil akurasi tertinggi

dihasilkan oleh algoritma *Logistic Regression*, *Random Forest*, dan *Support Vector Machine* sebesar 95%.

Penelitian yang dilakukan oleh Nabilah dan Nur (Sharfina & Ramadhan, 2023) mendapatkan hasil akurasi dari pengujian tanpa SMOTE sebesar 93% sedangkan pada pengujian menggunakan SMOTE mendapatkan hasil akurasi sebesar 97% pada penggunaan algortitma *Random Forest* dan pada algoritma *Naïve Bayes* mendapatkan hasil akurasi tanpa SMOTE sebesar 88% dan mendapatkan hasil akurasi sebesar 89% setelah menggunakan SMOTE.

Hal ini membuktikan bahwa penggunaan SMOTE dan *Random Forest* efektif dalam menangani data tidak seimbang dan meningkatkan hasil akurasi. Meskipun hasil akurasi yang diperoleh cukup tinggi, penelitian terdahulu hanya menggunakan algoritma *Random Forest* dan teknik SMOTE standar tanpa menggunakan variasi teknik SMOTE lain seperti Borderline-SMOTE*,* SMOTE-*IPF,* SMOTE-*LOF*, dan lainnya. Misalnya, penelitian yang dilakukan oleh (Mutmainah, 2021) yang mengklasifikasi penyakit stroke dengan hanya menggunakan algoritma *Random Forest* dan teknik SMOTE standar yang membuat data sintetis pada seluruh data minoritas tanpa variasi teknik SMOTE lainnya. Penelitian yang secara khusus mengkaji pengaruh variasi teknik Borderline-SMOTE*,* yang membuat data sintetis pada data minoritas yang hanya berada di area yang dekat dengan data mayoritas, terhadap performa algoritma *Random Forest* dalam klasifikasi risiko stroke masih sangat terbatas, sehingga diperlukan kajian lebih lanjut untuk mengetahui potensi peningkatan performa klasifikasi melalui penerapan variasi teknik SMOTE, yaitu Borderline-SMOTE.

Dengan adanya permasalahan di atas, maka dilakukan penelitian klasifikasi risiko stroke berdasarkan faktor medis dan gaya hidup menggunakan *Random Forest* dan variasi teknik SMOTE. Diharapkan penelitian ini dapat menghasilkan model prediksi yang lebih akurat dan berkontribusi dalam pengembangan teknologi deteksi dini penyakit stroke.

1.2 Rumusan Masalah

Rumusan masalah yang diusulkan pada penelitian ini yaitu bagaimana menerapkan model *Random Forest* dan SMOTE untuk klasifikasi stroke berdasarkan faktor medis dan gaya hidup?

1.3 Batasan Masalah

Adapun batasan masalah pada penelitian ini antara lain:

1. Data yang digunakan merupakan data publik yang diambil dari kagle.com tahun 2020.
2. Metode yang digunakan adalah *Random Forest*.
3. Penggunaan metode *preproses* SMOTE untuk menangani data tidak seimbang.
4. Penelitian berfokus pada pengolahan data, pelatihan model, dan evaluasi performa.

1.4 Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan untuk:

1. Menerapkan model *Random Forest* untuk klasifikasi stroke berdasarkan faktor medis dan gaya hidup
2. Menerapkan teknik SMOTE untuk menyeimbangkan data.

3. Mengevaluasi performa model *Random Forest* dengan dan tanpa teknik SMOTE.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat:

1. Membantu menganalisis risiko stroke berdasarkan data pasien sebagai upaya pencegahan awal risiko stroke.
2. Mengkaji metode *pre-proses* SMOTE dalam menangani data yang tidak seimbang dan pengaruh terhadap performa model *Random Forest*.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Penelitian oleh Fadilla, dkk (Fadmadika dkk., 2024) dilakukan dengan menganilisis pengaruh SMOTE pada performa algoritma *Random Forest* dan *Gradient Boosting* dalam prediksi penyakit stroke. Pada penelitian ini data latih dan data uji dibagi dengan perbandingan sekitar 80:20 dengan didapat hasil akurasi sebesar 94%, nilai presisi sebesar 84,6%, nilai recall hanya sebesar 52,5%, dan f1-score didapat sebesar 53% untuk model *Random Forest* pada pengujian data yang tidak seimbang kelasnya. Untuk model *Gradient Boosting* didapat hasil akurasi sebesar 93,7%, nilai presisi sebesar 46,9%, nilai recall sebesar 49,9%, dan f1-score sebesar 48,4%. Setelah di-*oversampling* menggunakan SMOTE didapat hasil untuk model *Random Forest* akurasi sebesar 95,5%, nilai presisi 78,8%, nilai recall 93%, dan f1-score sebesar 84%. Sedangkan untuk model *Gradient Boosting* didapat hasil akurasi sebesar 88%, presisi 87,6%, recall 87,5%, dan f1-score 87,5%. Jika penelitian di atas menggunakan pembagian data sekitar 80:20 dan menganilisis pengaruh SMOTE standar di beberapa algoritma, penelitian yang akan dilakukan data dibagi sekitar 70:30 dan hanya menggunakan algoritma *Random Forest* dan Borderline-SMOTE.

Pada penelitian lain yang dilakukan oleh Desti, dkk (Mualfah dkk., 2022) yang berjudul "Teknik SMOTE untuk mengatasi imbalance data pada deteksi penyakit stroke menggunakan algoritma *Random Forest*" didapati bahwa algoritma *Random Forest* efektif dalam mendeteksi penyakit stroke dan terbukti dapat

mengatasi data *imbalanced*. Pada penelitian ini dilakukan dua kali pengujian data, yaitu pengujian data tanpa SMOTE dan pengujian data menggunaan SMOTE. Untuk pengujian data tanpa SMOTE atau data yang masih tidak seimbang kelasnya didapatkan hasil akurasi sebesar 98%, nilai presisi sebesar 69%, nilai recall sebesar 51%, f1-score sebesar 51%. Sedangkan pada pengujian data dengan SMOTE atau data yang telah diseimbangkan kelasnya didapatkan hasil akurasi sebesar 91%, nilai presisi sebesar 92%, nilai recall 91%, dan f1-score sebesar 91%. Meskipun hasil akurasi yang didapat untuk pengujian data dengan SMOTE lebih kecil daripada hasil akurasi saat pengujian data tanpa SMOTE, namun untuk nilai performa lain seperti presisi, recall, dan f1-score didapatkan hasil yang lebih baik. Berbeda dengan penelitian di atas yang hanya berfokus menggunakan teknik SMOTE standar, penelitian yang akan dilakukan berfokus menggunakan variasi teknik SMOTE, yaitu Borderline-SMOTE.

Pada penelitian lain yang dilakukan oleh Yufis, dkk (Azhar dkk., 2022) yaitu membandingkan beberapa algoritma klasifikasi untuk memprediksi penyakit stroke. Algoritma yang dibandingkan pada penelitian ini antara lain, Decision Tree C4.5, *Logistic Regression*, *Random Forest*, *Support Vector Machine* (*SVM*), *K-Nearest Neighbours* (*KNN*), dan *Naive Bayes*. Penelitian ini mengambil data langsung dari kagle yang mana data ini berisi kumpulan informasi data pasien seperti jenis kelamin, usia, hipertensi, penyakit jantung, dan lain sebagainya yang berisi 12 kolom data. Hasil eksperimen menunjukkan pada data tidak seimbang didapat akurasi untuk Logistic Regresion, *Random Forest*, *SVM*, dan *KNN* sebesar 98,63%, *Decission Tree* didapat sebesar 97,28%, dan *Naive Bayes* didapat sebesar

72,4%. Sedangkan untuk hasil eksperimen data yang telah diseimbangkan didapat hasil akurasi untuk Logistic Regresion 73,48%, *Decission Tree* 68,70%, *Random Forest* 72,17%, *SVM* 76,52%, *KNN* 72,61%, dan *Naive Bayes* 72,04%. Jika penelitian yang dilakukan oleh Yufis, dkk. membandingkan beberapa algoritma, penelitian yang akan dilakukan ini hanya menggunakan algoritma *Random Forest*.

Ary, dkk. (Siregar, Ary Prandika dkk., 2023) melakukan implementasi algoritma *Random Forest* untuk klasifikasi diagnosis penyakit stroke. Pada penelitian ini dilakukan pembagian data 80% untuk data latih dan 20% untuk data uji. Penelitian menghasilkan akurasi sebesar 95% dan untuk AUC-ROC dihasilkan sebesar 0,80 sehingga dengan nilai AUC-ROC tersebut model dikatakan ke dalam klasifikasi baik. Berbeda dengan penelitian yang akan dilakukan, penelitian oleh Ary dkk. ini hanya berfokus pada pemodelan *Random Forest* tanpa melakukan teknik *balancing*, sedangkan penelitian yang akan dilakukan tidak hanya berfokus pada pemodelan *Random Forest* saja tapi juga melakukan *balancing* data menggunakan Borderline-SMOTE.

Fitri, dkk. (Airi dkk., 2023) melakukan komparasi beberapa algoritma metode klasifikasi untuk prediksi penyakit stroke. Penelitian ini menggunakan algoritma *Random Forest*, K-Nearest Neighbour (*KNN*), *Naive Bayes*. Hasil akurasi dari algoritma *Random Forest* didapat sebesar 92,5%, algoritma *KNN* didapat hasil akurasi sebesar 73,6%, dan algoritma *Naive Bayes* hasil akurasi didapat sebesar 71,9%. Jika penelitian oleh Fitri, dkk. membandingkan beberapa algoritma metode klasifikasi dan tanpa melakukan *balancing* data, berbeda penelitian yang akan

dilakukan hanya menggunakan algoritma *Random Forest* dan melakukan teknik *balancing* data menggunakan Borderline-SMOTE.

Tabel 2.1 Penelitian Terdahulu

No	Nama Peneliti	Judul Penelitian	Metode
1.	(Fadmadika et al., 2024)	Pengaruh SMOTE Terhadap Performa Algoritma *Random Forest* dan Algoritma *Gradient Boosting* dalam Memprediksi Penyakit Stroke	*Random Forest* dan *Gradient Boosting*
2.	(Mualfah et al., 2022)	Teknik SMOTE untuk Mengatasi Imbalance Data pada Deteksi Penyakit Stroke menggunakan Algoritma *Random Forest*	*Random Forest*
3.	(Azhar et al., 2022)	Perbandingan Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Stroke	*Decision Tree C4.5*, *Logistic Regression*, *Random Forest*, *Support Vector Machine* (*SVM*), *K-Nearest Neighbours* (*KNN*), dan *Naive Bayes*
4.	(Ary Prandika Siregar dkk., 2023)	Implementasi Algoritma *Random Forest* Dalam Klasifikasi Diagnosis Penyakit Stroke	*Random Forest*
5.	(Airi dkk., 2023)	Komparasi Metode Klasifikasi Data Mining Untuk Prediksi Penyakit Stroke	*Random Forest*, *KNN*, *Naïve Bayes*
6.	Peneliti	Klasifikasi Risiko Stroke Berdasarkan Faktor Medis dan Gaya Hidup Menggunakan *Random Forest* dan SMOTE	*Random Forest*

2.2 Stroke

Stroke merupakan penyakit dengan risiko kematian paling tinggi di nomer 2 setelah penyakit jantung di dunia. Stroke juga merupakan penyakit yang menyebabkan kecacatan nomer 1 di dunia. Secara global, setiap tahunnya terdapat 15 juta orang terserang penyakit stroke dimana 13% meninggal dunia dan selebihnya mengalami cacat permanen (Thalib & Dimara, 2021). Stroke terjadi karena adanya penyumbatan suplai darah pada otak yang disebabkan oleh penggumpalan darah di otak sehingga pasokan oksigen dan nutrisi terganggu dan

menyebabkan terjadinya kerusakan jaringan otak dan fungsi syaraf (Puspitasari, 2020). Stroke sendiri memiliki 2 jenis, yaitu stroke iskemik dan stroke hemoragik.

Stroke iskemik merupakan stroke yang terjadi karena sumbatan dan paling sering terjadi sedangkan stroke hemoragik adalah stroke yang terjadi karena adanya pendarahan atau perpecahan pembuluh darah (Aulyra Familah dkk., 2024). Terdapat 2 macam stroke iskemik, yaitu stroke emboli yang terjadi saat ada pembekuan darah dalam jantung yang kemudian terangkut ke otak dan stroke trombotik yang terjadi saat ada pembekuan darah pada arteri yang menyuplai darah ke otak. Stroke hemoragik juga terdapat 2 macam, yaitu pendarahan intraserebral dan pendarahan subarachnoid. Pendarahan intraserebral terjadi saat pembuluh darah pecah dan darah masuk ke dalam jaringan dan menyebabkan sel-sel pada otak mati dan kerja otak terhenti, sering terjadi karena hipertensi. Pendarahan subarachnoid terjadi saat pembuluh darah pecah di sekitar otak dan bocor di antara otak dan tulang tengkorak, sering terjadi karena pecahnya aneurisma.

2.3 Faktor Medis

Terjadinya stroke tentu memiliki faktor penyebabnya. Faktor medis bisa menjadi salah penyebab terjadinya stroke. Faktor medis sendiri merujuk kepada kondisi fisik dan komplikasi yang bisa mempengaruhi aspek perkembangan dan dapat mempengaruhi perkembangan sosial dan kebutuhan akan layanan medis (Conners & Erhardt, 1998). Faktor medis penyebab terjadinya stroke sendiri mencakup hipertensi, diabetes melitus, riwayat penyakit jantung, tingkat kolesterol yang tinggi, obesitas, usia, dan jenis kelamin.

Hipertensi menjadi salah satu faktor medis yang paling berpengaruh dalam seseorang terserang penyakit stroke. Berdasarkan penelitian yang dilakukan oleh Hendri, dkk. (Budi dkk., 2020) hipertensi menjadi faktor utama penyebab terjadinya stroke dengan nilai *p value* sebesar 0,052 yang mana nilai ini adalah nilai paling tinggi di antara nilai faktor medis lainnya yang diteliti. Selain hipertensi, diabetes melitus juga menjadi salah satu faktor medis yang paling sering menyebabkan stroke karena stroke bisa terjadi karena tingginya kadar gula di tubuh. Penyakit stroke juga bisa menyerang siapa saja dan berbagai kalangan umur baik yang tua maupun muda, baik laki-laki maupun perempuan. Dalam penelitian (Rahayu, 2023) pasien dengan usia di atas 50 tahun dan jenis kelamin laki-laki memiliki tingkat presentase paling tinggi yaitu masing-masing 76% dan 52%.

2.4 Gaya Hidup

Selain faktor medis, risiko terjadinya stroke juga bisa dari gaya hidup orang. Gaya hidup dalam bidang kesehatan sendiri merupakan pola perilaku yang berhubungan dengan kesehatan berdasarkan pilihan hidup seseorang (Brivio dkk., 2023). Gaya hidup sendiri ada 2 yaitu gaya hidup sehat dan gaya hidup tidak sehat. Gaya hidup sehat merupakan gaya hidup yang bebas dari masalah mental dan fisik. Gaya hidup sehat memilik banyak sekali manfaat, seperti hidup menjadi lebih bahagia, meningkatkan energi tubuh, dan mengurangi resiko penyakit (Kemenkes, 2018). Sedangkan gaya hidup tidak sehat merupakan gaya hidup seseorang yang memiliki pola hidup tidak baik sehingga bisa menyebabkan timbulnya penyakit salah satunya yaitu stroke.

Gaya hidup tidak sehat seperti sering mengonsumsi alkohol, merokok, tidak pernah olahraga, dan pola makan yang buruk dapat menjadi penyebab risiko penyakit stroke. Pola makan yang buruk seperti sering makan makanan yang berlemak, tidak pernah makan sayur dan buah, memakan makanan yang tidak jelas kandungan nutrisi nya, sering memakan *junk food*, dan kurang olahraga dapat menyebabkan obesitas yang menjadi salah satu penyebab stroke. Sebagaimana pada penelitian (Budi dkk., 2020) selain hipertensi faktor kurangnya olahraga dan pola makan yang berlemak menjadi faktor utama terserangnya penyakit stroke. Selain itu juga pada penelitian tersebut penderita stroke banyak memiliki riwayat merokok.

2.5 Klasifikasi

Dalam perkembangan teknologi, pengelolahan data yang berjumlah bisa menjadi mudah efesien. Salah satu perkembangan teknologi untuk mengelolah data besar yaitu *data mining*. *Data mining* merupakan proses ekstraksi pengetahuan dari data besar dan kompleks (SLN, 2024). Salah satu metode dalam *data mining* yaitu klasifikasi. Secara umum klasifikasi merupakan pengelompokan benda ke dalam suatu kelompok berdasarkan ciri-ciri yang dimiliki oleh objek klasifikasi. Dalam penerapannya klasifikasi dapat kita temukan dalam kehidupan sehari hari seperti pengelompokan barang-barang yang ada di rak supermarket berdasarkan kategori seperti rak sabun, rak makanan manis, rak makanan gurih, dan lain sebagainya.

Klasifikasi dalam *data mining* merupakan suatu metode yang menentukan faktor dominan yang belum diketahui untuk mengambil sebuah keputusan dari sebuah atribut (x) ke salah satu atribut label (y) yang sudah didefinisikan

sebelumnya (Wanjar dkk., 2020). Dalam *data mining* klasifikasi dapat diproses dengan menggunakan algoritma *machine learning* diantaranya *Decission Tree*, *Naïve Bayes*, *Neural Network*, *Support Vector Machine*, dan *Logistic Regression*.

2.6 *Random Forest*

Random Forest merupakan salah satu algoritma model *machine learning* yang biasa digunakan untuk pengklasifikasi dataset dalam jumlah besar. *Random Forest* merupakan usulan dari Leo Breiman pada tahun 2001 yang menggabungkan beberapa pohon keputusan acak dan prediksi pohon keputusan dengan rata-rata untuk menghasilkan keputusan akhir. Algoritma *Random Forest* ini sering digunakan selain karena penggunaannya yang mudah, *Random Forest* juga diakui kemampuannya dalam menangani data besar terbukti dari hasil akurasi yang baik dari penelitian terdahulu (Santika dkk., 2023).

Statistikawan Leo Breiman yang mengusulkan algoritma *Random Forest* menjelaskan teori *Random Forest* dalam jurnal yang berjudul "*Random Forest*s" pada tahun 2001 (Breiman, 2001). Dalam jurnal tersebut dijelaskan bahwa pengklasifikasian dengan *Random Forest* dihasilkan dari ansambel pohon yang mana ansambel pohon dibuat dengan vektor acak berdasarkan sampel dari data latih dan setiap pohon bisa memilih kelas yang nantinya kelas paling mayoritas dipilih akan dibuat keputusan akhir untuk klasifikasi. Dalam penelitiannya, Breiman menyertakan beberapa rumus penting yang menjelaskan teori dibalik algoritma *Random Forest*. Berikut rumus-rumus penting dalam *Random Forest*:

1. **Margin Function**: rumus ini digunakan untuk mengukur seberapa yakin prediksi ensemble terhadap kelas yang benar diantara kelas lain.

$$mg(X,Y) = \frac{1}{K} + \sum_{k=1}^{K} I(h_k(\mathrm{X}) = Y) - \max_{j \neq Y} \frac{1}{K} + \sum_{k=1}^{K} I(h_k(\mathrm{X}) = j) \quad (2.1)$$

I (·): fungsi indicator

$h_k(\mathrm{X})$: prediksi dari pohon ke-k

K : jumlah pohon dalam hutan

2. **Generalization Error** : probabiitas bahwa margin negatif berarti prediksi salahh.

$$PE^* = P_{X,Y}(\mathrm{mg}(X,Y) < 0) \quad (2.2)$$

3. **Strength of Classifier** : rata-rata margin seluruh data.

$$s = E_{X,Y}[\mathrm{mr}(X,Y)] \quad (2.3)$$

4. **Upper Bound Generalization Error** : menggunakan korelasi antar pohon.

$$PE^* \leq \frac{\rho(1-s^2)}{s^2} \quad (2.4)$$

ρ : korelasi rata-rata pohon

s : kekuatan (strength) dari pohon-pohon

5. **Regression Error Bound** : untuk kasus regresi, error rata-rata dari *Random Forest* lebih kecil dari error rata-rata pohon tunggal dikalikan korelasi residual.

$$PE^*_{\mathrm{forest}} \leq \rho \cdot PE^*_{\mathrm{tree}} \quad (2.4)$$

2.7 SMOTE

Ketidakseimbangan data terjadi saat data memiliki kelas minoritas dan mayoritas. Data dikatakan tidak seimbang apabila terdapat perbandingan kelas 95:5 atau lebih ekstrim lagi. Dalam mengatasi ketidakseimbangan data bisa dilakukan dengan teknik over-sampling yang merupakan teknik menambah sampel data dan under-sampling yang merupakan teknik mengurangi sampel data pada kelas mayoritas. Selain kedua teknik tersebut, terdapat juga teknik lain yaitu SMOTE. SMOTE (*Synthetic Minority Over-sampling Technique*) adalah sebuah teknik untuk mengatasi overfitting akibat ketidakseimbangan kelas oleh dataset.

Teknik SMOTE pertama kali diperkenalkan oleh Nitesh V. Chawla dan timnya yang teridiri dari 3 orang pada tahun 2002 yang ditulis di artikel jurnal berudul "SMOTE: Synthetic Minority Over-sampling Technique". Dalam jurnal yang ditulis oleh Chawla dan timnya teknik SMOTE memadukan teknik under-sampling pada kelas mayoritas dan over-sampling pada kelas minoritas. Pada teknik under-sampling yang menghapus sebagian data bisa menyebabkan hilangnya data yang penting sedangkan teknik over-sampling yang menambah data dengan menduplikasi bisa menyebabkan pemodelan mengalami overfitting. Kelemahan teknik-teknik tersebut menjadikan teknik SMOTE ini digunakan (Chawla dkk., 2002).

a. Borderline-SMOTE

Dalam perkembangannya, SMOTE tidak hanya digunakan dalam bentuk standar, tetapi juga dikembangkan menjadi beberapa variasi untuk meningkatkan performa dalam menangani ketidakseimbangan data. Salah

satu variasi dari SMOTE adalah Borderline-SMOTE. Borderline-SMOTE dikembangkan untuk memperbaiki kelemahan dari SMOTE standar yang cenderung menghasilkan data sintetis secara acak tanpa memperhatikan posisi data dalam ruang fitur. Borderline-SMOTE dikembangkan oleh tim dari salah satu universitas di Beijing yang ditulis pada jurnal berjudul "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning" pada tahun 2005 (H. Han dkk., 2005). Dalam jurnal dijelaskan bahwa Borderline-SMOTE dilakukan dengan memfokuskan penambahan data sintetis pada kelas minoritas yang dekat dengan garis perbatasan keputusan. Borderline-SMOTE sendiri memiliki 2 varian utama:

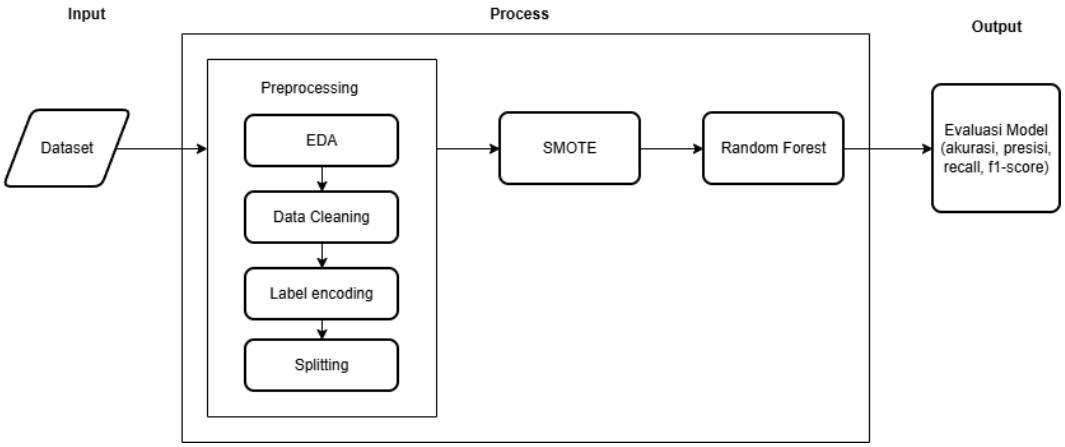
1. **Borderline-SMOTE1**: data sintetis dihasilkan dengan mengambil contoh dari data minoritas yang berada di dekat kelas mayoritas atau bisa disebut zona bahaya dan tetangga terdekatnya.
2. **Borderline-SMOTE2**: pada variasi ini selain mengambil contoh dari data minoritas dan tetangga terdekat, variasi ini juga mengahasilkan data sintetis dengan mempertimbangkan kelas mayoritas dengan tujuan memperkuat contoh minoritas yang ada di dekat perbatasan.

BAB III

METODOLOGI PENELITIAN

3.1 Desain

Penelitian dilakukan dengan memasukkan dataset stroke yang telah diambil kemudian dilakukan preproses data untuk merapikan data sebelum diolah. Setelah dilakukan preproses data dilakukan SMOTE untuk menyeimbangkan kelas dataset yang tidak seimbang dan setelah dataset rapi serta seimbang dilakukan proses utama yaitu proses klasifikasi dengan algoritma *Random Forest* dan menghasilkan output hasil evaluasi model seperti akurasi, presisi, recall, dan f1-score. Untuk desain sistem penelitian ini terdapat pada gambar berikut.



Gambar 3.1 Desain Penelitian

3.2 Dataset

Penelitian ini menggunakan data dari data publik yang diambil dari Kaggle yang berjudul “Stroke Prediction Dataset” oleh Fedesoriano yang memiliki 12 kolom dan 5110 baris, 5 kolom berisi data kategori dan 7 kolom lainnya berisi data numerik . Dataset ini berisi informasi umum pasien seperti id, jenis kelamin, umur,

pekerjaan, status nikah, dan status merokok. Selain informasi umum, dataset ini juga berisi informasi medis pasien seperti hipertensi, riwayat penyakit jantung, kadar glukosa dalam darah, *bmi*, dan status stroke.

Tabel 3.1 Variabel Dataset

Kolom	**Penjelasan**
id	Data unik
gender	Terdiri dari jenis kelamin “male”, “female”, dan “other”
age	Berisi data umur pasien
hypertension	Data “0” apabila pasien tidak memiliki hipertensi, dan “1” apabila pasien memiliki hipertensi
heart_dieses	Data “0” apabila pasien tidak memiliki penyakit jantung, dan “1” apabila pasien memiliki penyakit jantung
ever_married	Berisi data “No” apabila pasien belum menikah, dan “yes” apabila pasien telah/pernah menikah
work_type	Berisi data pekerjaan pasien "children", "Govt_jov", "Never_worked", "Private" or "Self-employed"
residence	Berisi data tempat tinggal pasien "Rural" or "Urban"
avg_glucose_level	Berisi data kadar rata-rata glukosa yang ada dalam darah pasien
bmi	Berisi data *body mass index* pasien
smoking_status	Berisi data status merokok pasien "formerly smoked", "never smoked", "smokes" or "Unknown"
stroke	Data “0” jika pasien tidak stroke, dan “1” jika pasien stroke

Tabel 3.2 Contoh Dataset

Id	**gender**	**age**	**hypertensi**	**heart_dies**	**ever_mar ried**	**work_typ e**	**residence**	**avg_gluco se_level**	**bmi**	**smoking_s tatus**	**stroke**
9046	Male	67	0	1	Yes	Private	Urban	228.69	36.6	formerly smoked	1
51676	Female	61	0	0	Yes	Self-employed	Rural	202.21	N/A	never smoked	1
31112	Male	80	0	1	Yes	Private	Rural	105.92	32.5	never smoked	1
60182	Female	49	0	0	Yes	Private	Urban	171.23	34.4	smokes	1
……	……..	…	…	…	…	…….	……	…….	…	……	…
……	……..	…	…	…	…	…….	……	…….	…	……	…
44873	Female	81	0	0	Yes	Self-employed	Urban	125.2	40	never smoked	0
19723	Female	35	0	0	Yes	Self-employed	Rural	82.99	30.6	never smoked	0
37544	Male	51	0	0	Yes	Private	Rural	166.29	25.6	formerly smoked	0

3.3 Preprocessing Data

Data yang telah didapat masih belum bisa langsung diklasifikasi karena data yang diambil masih data mentah sehingga dapat memengaruhi hasil akhir. Agar hasil akhir klasifikasi mendapatkan hasil yang bagus diperlukan langkah sebelum memulai proses klasifikasi, yaitu langkah preprocessing data.

Preprocessing data pada data mining biasanya dilakukan dengan membersihkan data dari data null/kosong atau membersihkan data dari kata yang tidak perlu, memahami pola data, dan juga mengubah isi data yang berupa kategori ke numerik (García dkk., 2015). Pada penelitian ini preprocessing data dilakukan dengan beberapa langkah sebagai berikut.

3.3.1 EDA

EDA (*Exploration Data Analysis*) merupakan salah satu bagian dari preproses data yang dilakukan dengan menganalisis dan mengeksplorasi data sebelum dilakukan proses implementasi model. EDA dilakukan guna memahami karakteristik dataset dengan melakukan analisis struktur data, mengecek missing value, mengecek distribusi data, dan analisis korelasi antar fitur data. EDA pada penelitian ini dilakukan dengan:

1. Mengecek jumlah kolom, baris, dan tipe data. Pengecekan ini dilakukan untuk mengetahui jumlah baris dan kolom serta mengetahui apa saja tipe datanya. Pada gambar 3.2 kolom data berjumlah 12 dan baris data berjumlah 5110, dan untuk tipe data ada pada gambar 3.3

```
(5110, 12)
```

Gambar 3.2 Jumlah Baris dan Kolom

```
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

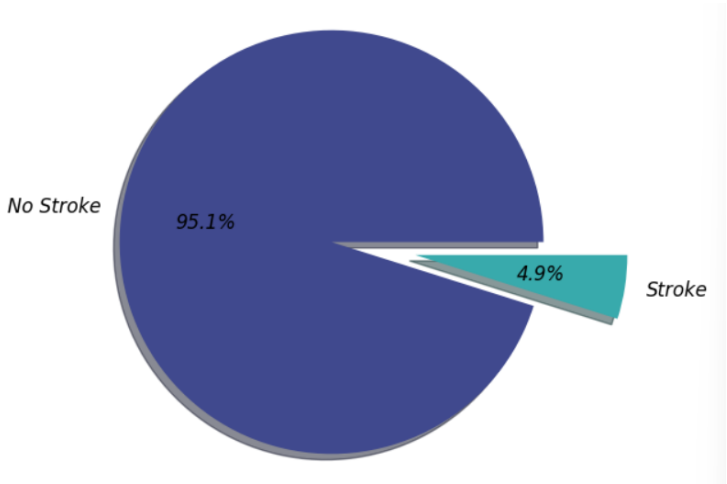
Gambar 3.3 Tipe Data

2. Mengecek deskripsi statistik. Penegecekan deskripsi statistik dilakukan untuk mengetahui karakteristik umum data dengan menghitung mean, median, nilai min dan max, dan standar deviasi pada data numerik. Gambar 3.4 merupakan hasil perhitungan deskripsi statistik data numerik.

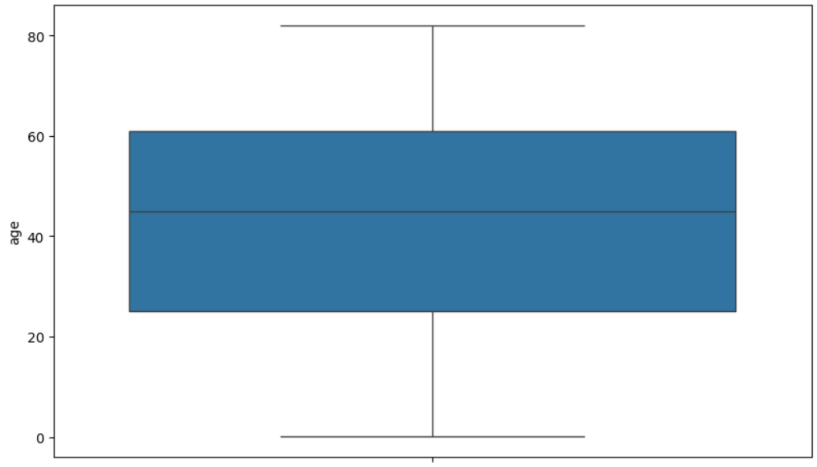
	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke
count	5110.000000	5110.000000	5110.000000	5110.000000	4909.000000	5110.000000
mean	43.226614	0.097456	0.054012	106.147677	28.893237	0.048728
std	22.612647	0.296607	0.226063	45.283560	7.854067	0.215320
min	0.080000	0.000000	0.000000	55.120000	10.300000	0.000000
25%	25.000000	0.000000	0.000000	77.245000	23.500000	0.000000
50%	45.000000	0.000000	0.000000	91.885000	28.100000	0.000000
75%	61.000000	0.000000	0.000000	114.090000	33.100000	0.000000
max	82.000000	1.000000	1.000000	271.740000	97.600000	1.000000

Gambar 3.4 Deksripsi Statistik

3. Memvisualisasikan distribusi data. Visualisasi dilakukan guna memahami data secara visual. Gambar 3.5 merupakan gambar visualisasi kelas stroke dan gambar 3.6 merupakan gambar visualisasi kelas age.



Gambar 3.5 Visualisasi Kelas Stroke



Gambar 3.6 Visualisasi Kelas Age

4. Mengecek data duplikat dan *missing value*. Pengecekan ini dilakukan untuk mengetahui apakah terdapat data duplikat dan *missing value* agar dapat dilakukan data cleaning apabila terdapat data duplikat atau data *missing value*. Pada dataset tidak terdapat data duplikat dan terdapat 201 *missing value* di kolom bmi.

3.3.2 Data Cleaning

Pembersihan data (*data cleaning*) merupakan salah satu langkah dalam preproses data yang dilakukan dengan menghapus atau mengisi data yang hilang

(*missing values*), menghapus duplikasi data, dan mengoreksi inkonsistensi data. Data stroke yang telah diambil dan telah dilakukan EDA, kemudian dilakukan cleaning data dengan:

1. Menghapus kolom id. Kolom id dihapus karena kolom ini tidak diperlukan dan tidak berpengaruh untuk hasil akhir klasfikasi.
2. Mengisi *missing value*. Kolom bmi yang terdapat 201 *missing value* diiisi dengan dilakukan perhitungan mean.
3. Menghapus data *other* di kelas gender. Pada kelas gender data *other* dihapus karena tidak diperlukan.

3.3.3 Data Encoding

Data encoding merupakan suatu proses dimana data yang berupa data kategori diubah menjadi data numerik. Encoding terdapat beberapa macam tergantung bagaimana cara data tersebut diubah salah satunya yaitu label encoding. Label encoding dilakukan dengan mengubah data kategori menjadi data numerik secara langsung. Pada penelitian ini dilakukan label encoding dengan menggunakan library labelencoder() pada fitur yang berupa data kategori seperti fitur gender, smoking_status, ever_merried, work_type, dan residence.

Tabel 3.3 Data setelah Label Encoding

gender	age	hypertension	heart_dies	ever_mar	work_typ	residence	avg_glucose_level	bmi	smoking_s	stroke
1	67.0	0	1	1	2	1	228.69	36.600000	1	1
0	61.0	0	0	1	3	0	202.21	28.893237	2	1
1	80.0	0	1	1	2	0	105.92	32.500000	2	1
0	49.0	0	0	1	2	1	171.23	34.400000	3	1
0	79.0	1	0	1	3	0	174.12	24.000000	2	1

3.3.4 Splitting Data

Splitting data dilakukan dengan memisahkan fitur (x) dan target (y) sebelum dilakukan pemodelan. Pada proses ini dilakukan pemisahan kolom data dengan kolom stroke sebagai target dan kolom lainnya sebagai fitur. Selain itu pada proses ini juga dilakukan pembagian dataset menjadi data latih dan data uji. Pembagian data uji dan data latih menggunakan pembagian data *Hold-out* dibagi sebesar 70:30, 80:20, 90:10, dan juga menggunakan *K-Fold cross validation*.

3.4 SMOTE

Pada penelitian ini dataset yang digunakan memiliki kelas yang tidak seimbang. Terdapat 4861 kelas tidak stroke dan 249 kelas stroke. Pada saat dataset memiliki kelas yang tidak seimbang maka akan terjadi overfitting saat dilakukan pemodelan sehingga diperlukan suatu teknik untuk menyeimbangkan kelas dataset. Salah satu teknik yang paling sering digunakan untuk menyeimbangkan kelas yaitu teknik oversampling dengan SMOTE (*Synthetic Minority Over-sampling Technique*). Pada penelitian ini menggunakan variasi dari perkembangan teknik SMOTE, yaitu Borderline-SMOTE. Borderline-SMOTE memiliki 2 variasi yaitu Borderline-SMOTE1 dan Borderline-SMOTE*2.*

Algoritma Borderline-SMOTE1 bekerja dengan:

1. Mengidentifikasi data minoritas dan mayoritas;
2. Menghitung tetangga terdekat menggunakan *KNN* di setiap sampel yang termasuk data minoritas;
3. Menentukan sampel minoritas yang berada di area bahaya atau data minoritas yang berada di dekat data mayoritas;

4. Membuat sampel sintetis hanya untuk sampel yang berada di area bahaya.

Sedangkan algoritma Borderline-SMOTE*2* bekerja dengan:

1. Mengidentifikasi data minoritas dan mayoritas;
2. Menghitung tetangga terdekat menggunakan *KNN* di setiap sampel yang termasuk data minoritas;
3. Menentukan sampel minoritas yang berada di area bahaya atau data minoritas yang berada di dekat data mayoritas;
4. Membuat sampel sintetis bisa dari data minoritas maupun mayoritas.

3.5 Implementasi *Random Forest*

Pada implementasi model klasifikasi digunakan algoritma *Random Forest* sebagai model klasifikasi. Alasan mengapa memilih algoritma ini adalah karena *Random Forest* mampu menangani data dengan jumlah yang sangat besar terbukti dari beberapa penelitian. *Random Forest* merupakan salah satu algoritma data mining yang digunakan dalam proses klasifikasi. Dalam melakukan klasifikasi *Random Forest* menggabungkan beberapa pohon keputusan (*decision tree*) untuk meningkatkan akurasi dengan mengambil rata rata dari beberapa pohon.

Algoritma *Random Forest* menggunakan hyperparameter utama seperti *n-estimators, max_depth* untuk menentukan jumlah pohon dan kedalaman maksimum dan *random_state* untuk memastikan hasil yang sama setiap kode dijalankan. Dalam pengklasfikasian *Random Forest* menggunakan kumpulan pohon berstruktur $\{h(x,\Theta_k), k=1,\dots\}$ di mana $\{\Theta_k\}$ merupakan vector acak yang mengontrol pertumbuhan setiap pohon dan x merupakan input data. Proses

pengambilan keputusan akhir *Random Forest* dilakukan dengan beberapa langkah sebagai berikut:

1. Membangun pohon dengan menggunakan sampel acak dari data latih (*bagging*) dikenal sebagai *boostraping*;
2. Pemilihan fitur acak, setiap pohon memilih dan menggunakan subset acak dari fitur;
3. Pembentukan pohon menggunakan algoritma CART (*classification and Regression Tree*);
4. Pemilihan kelas oleh setiap pohon yang telah terbentuk dan kelas yang mayoritas dipilih akan digunakan untuk prediksi akhir.

3.6 Evaluasi Model

Evaluasi model dilakukan setelah proses pemodelan selesai. Evaluasi model dilakukan dengan matix evaluasi yang umum digunakan untuk penelitian seperti akurasi, presisi, recall, dan f1-score.

1. **Akurasi** dilakukan dengan mengukur presentase prediksi yang benar dari data yang telah dilakukan proses dengan model *Random Forest*. Pengukuran presentase dihitung dengan formula berikut.

$$akurasi = \frac{\text{jumlah prediksi benar}}{\text{total data}} \tag{3.1}$$

2. **Presisi** dilakukan untuk mengukur seberapa tepat model memprediksi data kelas yang positif stroke dengan formula sebagai berikut.

$$presisi = \frac{\text{true positive}}{\text{true positive} + \text{false positive}} \tag{3.2}$$

3. **Recall** dilakukan untuk mengukur sebarapa baik model menangkap semua data positif stroke dengan formula sebagai berikut.

$$recall = \frac{\text{true positive}}{\text{true positive} + \text{false negative}} \tag{3.3}$$

4. **F1-Score** merupakan kombinasi dari presisi dan recall. Formula untuk menghitung f1-score sebagai berikut.

$$f1 - score = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.4}$$

5. **Confussion Matrix** merupakan metrik yang menunjukkan hasil prediksi atau klasifikasi model yang terdiri dari prediksi model dan kebenarannya untuk dibandingkan.

3.7 Skenario Pengujian

Pengujian data stroke akan dilakukan di *google colab* yang diawali dengan penginput dataset stroke yang digunakan yang kemudian melewati tahap *preprocessing* dimana data akan dibersihkan dari data yang tidak perlu dan mengubah data kategorial menjadi numerikal yang kemudian data dibagi menjadi data uji dan data latih. Pembagian data latih dan data uji menjadi 70:30, 80:20, dan 90:10. Selain pembagian data dengan membagi data latih dan data uji, penelitian ini juga dilakukan pembagian data dengan menggunakan *Kfold*. Setelah itu tahap penggunaan SMOTE untuk data latih yang telah dibagi pada saat tahap preporocessing. Tahap SMOTE selesai kemudian tahap implementasi model untuk klasifikasi menggunakan *Random Forest* dengan tuning parameter menggunakan *GridSearchCV* dan *RandomizedSearchCV* pada data yang telah dilakukan SMOTE. SMOTE yang digunakan pada penelitian ini merupakan variasi dari teknik SMOTE

yaitu, Borderline-SMOTE1 dan Borderline-SMOTE*2* dan kemudian terakhir dilakukan evaluasi model *Random Forest* pada data sesudah SMOTE. Implementasi *Random Forest* juga dilakukan tanpa *balancing* data. Tuning parameter menggunakan *GridSearchCV* dilakukan dengan mencari nilai optimal dari:

1. *n_estimators* antara 100, 200, 300.
2. *max_depth* antara 10, 13, 15.

Dan untuk tuning parameter menggunakan *RandomizedSearchCV* dilakukan dengan mencari:

1. *n_estimators* dari 100 sampai 500
2. *max_depth* dari 10 sampai 20

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pengujian

Pengujian dilakukan dengan menguji data stroke yang tidak seimbang dari 5109 data dibagi menjadi 70:30, 80:20, dan 90:10. Pembagian data 70:30 dibagi data latih sebesar 3576 dan untuk data uji sebesar 1533. Pembagian data 80: 20, data latih sebesar 4087 dan data uji sebesar 1022. Pada pembagian data 90:10, data latih sebesar 4589 dan data uji sebesar 511. Dan untuk pembagian data menggunakan *KFold* jumlah k yang digunakan Adalah k=5 dan k=10. Penelitian dilakukan dengan mengimplementasi model tanpa teknik penyeimbangan data dan dengan teknik penyeimbangan data. Sebeleum implementasi model, dilakukan tuning parameter menggunakan *GridSearchCV* dan *RandomizedSearchCV.* Penyeimbangan data dilakukan dengan menggunakan teknik Borderline-SMOTE1 dan Borderline-SMOTE*2* pada data latih saja agar data uji tetap data asli tanpa tambahan data sintetis*,* dan setelah dilakukan teknik *balancing* tersebut dilakukan juga tuning parameter menggunakan *GridSearchCV* dan *RandomizedSearchCV* sebelum menggunakan model *Random Forest* untuk klasifikasi. Selain itu untuk mengetahui bagaimana performa model *Random Forest* dilakukan evaluasi model menggunakan *classification report* dan *confusion matrix*.

4.2 Hasil Pengujian *Random Forest* tanpa SMOTE

Pengujian ini dilakukan klasifikasi menggunakan *Random Forest* tanpa menggunakan teknik *balancing* data SMOTE. Setelah pembagian data dilakukan tuning parameter menggunakan *GridSearchCV dan RandomizedSearchCV*.

4.2.1 Hasil Pengujian 70:30

Pengujian ini data dibagi menjadi 70:30 dan tuning parameter *GridSearchCV* didapat nilai n_estimators 100 dan max_depth sebesar 15. Dan hasil klasifikasi model didapat nilai akurasi sebesar 94%, presisi sebesar 47%, recall sebesar 50%, dan f1-score sebesar 49%. *Confusion matrix* didapat TP (True Positive) sebesar 0, FP (False Positive) didapat 0, TN (True Negative) didapat 1444, dan FN (False Negative) didapat 89. Hasil dari klasifikasi model baik *classification report* dan *confusion matrix* dapat dilihat pada gambar 4. 1 dan tabel 4. 1 berikut.

	precision	recall	f1-score	support
0	0.94	1.00	0.97	1444
1	0.00	0.00	0.00	89
accuracy			0.94	1533
macro avg	0.47	0.50	0.49	1533
weighted avg	0.89	0.94	0.91	1533

Gambar 4. 1 *Classification Report Random Forest* 70:30, *GridSearchCV* tanpa SMOTE

Tabel 4. 1 *Confusion matrix Random Forest* 70:30, *GridSearchCV* tanpa SMOTE

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	1444	0
True: Stroke	89	0

Pada pengujian ini juga dilakuka tuning parameter menggunakan *RandomizedSearchCV* dan mendapatkan nilai terbaik n_estimators 498 dan max_depth 17. Hasil *classification report* dari implementasi *Random Forest* untuk mengklasifikasi didapat akurasi sebesar 94%, presisi sebesar 47%, recall sebesar

50%, dan f1-score sebesar 49%. *Confusion matrix* yang didapatkan setelah klasifikasi yaitu TP (True Positive) sebesar 0, FP (False Positive) didapat 0, TN (True Negative) didapat 1444, dan FN (False Negative) didapat 89. Hasil dari *classification report* dan *confusion matrix* dapat dilihat pada gambar 4. 2 dan tabel 4. 2.

	precision	recall	f1-score	support
0	0.94	1.00	0.97	1444
1	0.00	0.00	0.00	89
accuracy			0.94	1533
macro avg	0.47	0.50	0.49	1533
weighted avg	0.89	0.94	0.91	1533

Gambar 4. 2 *Classification Report Random Forest* 70:30, *RandomizedSearchCV* tanpa SMOTE

Tabel 4. 2 *Confusion matrix Random Forest* 70:30, *RandomizedSearchCV* tanpa SMOTE

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	1444	0
True: Stroke	89	0

4.2.2 Hasil Pengujian 80:20

Pada pembagian data 80:20, tuning parameter menggunakan *GridSearchCV* didapat n_estimators sebesar 100 dan max_depth sebesar 15. Hasil dari implementasi model *Random Forest* pada pengujian ini mendapat hasil akurasi sebesar 94%, presisi sebesar 64%, recall sebesar 51%, dan f1-score sebesar 50% sebagaimana yang terdapat pada gambar 4. 3. Dan untuk *confusion matrix*, pengujian ini mendapat TP (True Positive): 1, FP (False Positive): 2, TN (True Negative): 958, dan FN (False Negative): 61, dapat dilihat pada tabel 4. 3.

	precision	recall	f1-score	support
0	0.94	1.00	0.97	960
1	0.33	0.02	0.03	62
accuracy			0.94	1022
macro avg	0.64	0.51	0.50	1022
weighted avg	0.90	0.94	0.91	1022

Gambar 4. 3 *Classification Report Random Forest* 80:20, *GridSearchCV* tanpa SMOTE

Tabel 4. 3 *Confusion matrix Random Forest* 80:20, *GridSearchCV* tanpa SMOTE

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	958	2
True: Stroke	61	1

Pada pengujian dengan menggunakan tuning parameter *RandomizedSearchCV* didapat n_esimators 262 dan max_depth 17. Setelah implementasi *Random Forest* digunakan untuk klasifikasi didapat hasil *classification report* akurasi sebesar 94%, presisi sebesar 47%, recall sebesar 50%, dan f1-score sebesar 48% dapat dilihat pada gambar 4. 4. *Confusion matrix* didapat TP (True Positive): 0, FP (False Positive): 1, TN (True Negative): 959, dan FN (False Negative): 62 dapat dilihat pada tabel 4. 4.

	precision	recall	f1-score	support
0	0.94	1.00	0.97	960
1	0.00	0.00	0.00	62
accuracy			0.94	1022
macro avg	0.47	0.50	0.48	1022
weighted avg	0.88	0.94	0.91	1022

Gambar 4. 4 *Classification Report Random Forest* 80:20, *RandomizedSearchCV* tanpa SMOTE

Tabel 4. 4 *Confusion matrix Random Forest* 80:20, *RandomizedSearchCV* tanpa SMOTE

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	959	1
True: Stroke	62	0

4.2.3 Hasil Pengujian 90:10

Pada pembagian data ini, tuning parameter menggunakan *GridSearchCV* mendapat nilai n_estimators 200 dan max_depth 15. Hasil pengujian *Random Forest* pada pengujian ini mendapat nilai akurasi sebesar 94%, presisi sebesar 47%, recall sebesar 50%, dan f1-score sebesar 48%. *Confusion matrix* pada pengujian ini mendapat TP (True Positive): 0, FP (False Positive): 2, TN (True Negative): 481, dan FN (False Negative): 28. Hasil dari *classification report* dan *confusion matrix* dapat dilihat pada tabel 4. 5 dan gambar 4. 5.

	precision	recall	f1-score	support
0	0.94	1.00	0.97	483
1	0.00	0.00	0.00	28
accuracy			0.94	511
macro avg	0.47	0.50	0.48	511
weighted avg	0.89	0.94	0.92	511

Gambar 4. 5 *Classification Report Random Forest* 90:10, *GridSearchCV* tanpa SMOTE

Tabel 4. 5 *Confusion matrix Random Forest* 90:10, *GridSearchCV* tanpa SMOTE

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	481	2
True: Stroke	28	0

Tuning parameter menggunakan *RandomizedSearchCV* mendapatkan niai n_estimators 191 dan nilai max_depth 17. Kemudian klasifikasi menggunakan *Random Forest* dilakukan dan menghasilkan nilai akurasi 95%, presisi 47%, recall 50%, dan f1-score 49%. Untuk *confusion matrix*-nya didapat TP (True Positive): 0, FP (False Positive): 0, TN (True Negative): 483, dan FN (False Negative): 28. Berikut gambar 4. 6 dan tabel 4. 6 yang menunjukkan hasil *classification report* dan *confusion matrix*.

	precision	recall	f1-score	support
0	0.95	1.00	0.97	483
1	0.00	0.00	0.00	28
accuracy			0.95	511
macro avg	0.47	0.50	0.49	511
weighted avg	0.89	0.95	0.92	511

Gambar 4. 6 *Classification report Random Forest* 90:10, *RandomizedSearchCV* tanpa SMOTE

Tabel 4. 6 *Confusion matrix Random Forest* 90:10, *RandomizedSearchCV* tanpa SMOTE

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	481	2
True: Stroke	28	0

4.2.4 Hasil Pengujian k=5

Pada pengujian ini menggunakan Kfold untuk membagi data. Jumlah k yang dilakukan pada penelitian ini sebanyak 5. Pada pengujian ini juga dilakukan tuning parameter menggunakan *GridSearchCV* dan *RandomizedSearchCV*. Pada tuning parameter *GridSearchCV* didapatkan nilai terbaik dari n_estimators dan max_depth pada setiap k dan terdapat pada tabel 4. 7 berikut.

Tabel 4. 7 Hasil Tuning Parameter *GridSearchCV* pada k=5, tanpa SMOTE

k	**N_estimators**	**Max_depth**
k ke-1	100	10
k ke-2	100	10
k ke-3	100	10
k ke-4	100	10
k ke-5	100	10

Dan hasil akurasi dari implementasi *Random Forest* dengan tuning parameter *GridSearchCV* yang didapat pada setiap k dan rata-rata akurasi dari seluruh k terdapat pada tabel 4. 8 berikut.

Tabel 4. 8 Akurasi k=5 tuning parameter *GridSearchCV*, tanpa SMOTE

k	**Akurasi**
k ke-1	100%
k ke-2	97%
k ke-3	93%
k ke-4	93%
k ke-5	97%
Rata-rata seluruh k	96%

Tuning parameter menggunakan *RandomizedSearchCV* pada setiap k mendapatkan nilai n_estimators dan max_depth terbaik sebagaimana yang terdapat tabel 4. 9 berikut.

Tabel 4. 9 Hasil Tuning Parameter *RandomizedSearchCV* pada k=5, tanpa SMOTE

k	**N_estimators**	**Max_depth**
k ke-1	450	11
k ke-2	106	19
k ke-3	344	10
k ke-4	408	16
k ke-5	338	12

Dan setelah dilakukan tuning, dilanjut dengan mengklasifikasi menggunakan *Random Forest* dan menghasilkan nilai akurasi pada setiap k yang mana nilai akurasi pada tiap k dihitung rata-rata. Tabel 4. 10 berikut menunjukkan hasil akurasi tiap k dan hasil rata-rata seluruh akurasi k.

Tabel 4. 10 Akurasi k=5 tuning parameter *RandomizedSearchCV*, tanpa SMOTE

K	**Akurasi**
k ke-1	100%
k ke-2	96%
k ke-3	93%
k ke-4	93%
k ke-5	97%
Rata-rata seluruh k	96%

4.2.5 Hasil Pengujian k=10

Pada pengujian ini penggunaan jumlah Kfold sebanyak k=10. Sama seperti pengujian dengan k=5, pengujian ini juga dilakukan tuning parameter

menggunakan *GridSearchCV* dan *RandomizedSearchCV* sebelum implementasi *Random Forest* untuk klasifikasi. Berikut hasil dari tuning parameter *GridSearchCV* terdapat pada tabel 4. 11.

Tabel 4. 11 Hasil Tuning Parameter *GridSearchCV* pada k=10, tanpa SMOTE

K	**N_estimators**	**Max_depth**
k ke-1	100	10
k ke-2	100	10
k ke-3	300	10
k ke-4	100	10
k ke-5	200	10
k ke-6	100	10
k ke-7	100	10
k ke-8	100	10
k ke-9	100	10
k ke-10	100	10

Dan hasil akurasi tiap k setelah dilakukan implementasi *Random Forest* dan hasil perhitungan rata-rata tiap k dapat dilihat pada tabel 4. 12 berikut.

Tabel 4. 12 Akurasi k=10 tuning parameter *GridSearchCV*, tanpa SMOTE

k	**Akurasi**
k ke-1	100%
k ke-2	100%
k ke-3	100%
k ke-4	93%
k ke-5	100%
k ke-6	87%
k ke-7	100%
k ke-8	100%
k ke-9	100%
k ke-10	93%
Rata-rata seluruh k	96%

Tuning parameter menggunakan *RandomizedSearchCV* menghasilkan nilai n_estimators dan max_depth sebagaimana yang terdapat pada tabel 4. 13 berikut.

Tabel 4. 13 Hasil Tuning Parameter *RandomizedSearchCV* pada k=10, tanpa SMOTE

k	**N_estimators**	**Max_depth**
k ke-1	310	15
k ke-2	298	16
k ke-3	364	12

k	N_estimators	Max_depth
k ke-4	439	19
k ke-5	314	16
k ke-6	278	14
k ke-7	434	14
k ke-8	284	11
k ke-9	456	14
k ke-10	141	19

Setelah tuning parameter dilakukan, implementasi *Random Forest* untuk klasifikasi dilakukan dan menghasilkan akurasi dari tiap k dan setelah didapat hasil akurasi tiap k, dilakukan perhitungan rata-rata untuk nilai akurasi seluruh k. Berikut tabel 4. 14 hasil akurasi tiap k dan rata-rata nya.

Tabel 4. 14 Akurasi k=10 tuning parameter *RandomizedSearchCV*, tanpa SMOTE

k	Akurasi
k ke-1	100%
k ke-2	100%
k ke-3	100%
k ke-4	93%
k ke-5	100%
k ke-6	87%
k ke-7	87%
k ke-8	100%
k ke-9	100%
k ke-10	93%
Rata-rata seluruh k	96%

4.3 Hasil Pengujian *Random Forest* dengan Borderline-SMOTE1

Pada pengujian ini menggunakan teknik Borderline-SMOTE1 untuk menyeimbangkan data yang dilakukan dengan hanya menambahkan data sintetis pada data minoritas yang berada di area bahaya dekat data mayoritas dengan mengambil sampel dari data minoritas saja.

4.3.1 Hasil Pengujian 70:30

Pada pengujian dengan pembagian data 70:30 dilakukan teknik Borderline-SMOTE1 pada data latih dan menghasilkan data sintetis pada data minoritas data latih, perbedaan jumlah data latih dapat dilihat pada tabel 4. 15 berikut.

Tabel 4. 15 Hasil penambahan data sintetis 70:30, Borderline-SMOTE1

Kelas	**Sebelum**	**Sesudah**
0 (mayoritas)	3416	3416
1 (minoritas)	160	3416

Hasil tuning parameter *GridSearchCV* pencarian *n_estimators* didapat 100 dan *max_depth* didapat 15. Penggunaan *Random Forest* pada pengujian didapat hasil *confussion matrix* TP (True Positive) didapat 14, FP (False Positive) didapat 70, TN (True Negative) didapat 1374, dan FN (False Negative) didapat 75. Hasil performa model *Random Forest* didapat akurasi sebesar 91%, presisi 56%, *recall* 55%, *f*1-*score* 56%. Hasil dari penggunaan model *Random Forest* terdapat pada gambar 4. 7 dan tabel 4. 16 yang menunjukkan hasil dari *confussion matrix* dan *classification report* dari implementasi *Random Forest* dengan teknik Borderline-SMOTE1 dan tuning parameter menggunakan *GridSearchCV* pada pengujian 70:30.

	precision	recall	f1-score	support
0	0.95	0.95	0.95	1444
1	0.17	0.16	0.16	89
accuracy			0.91	1533
macro avg	0.56	0.55	0.56	1533
weighted avg	0.90	0.91	0.90	1533

Gambar 4. 7 *Classification Report Random Forest* 70:30, *GridSearchCV* Borderline-SMOTE1

Tabel 4. 16 *Confusion matrix Random Forest* 70:30, *GridSearchCV* Borderline-SMOTE1

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	1374	70
True: Stroke	75	14

Sedangkan hasil klasifikasi setelah tuning parameter dengan *RandomizedSearchCV* pencarian nilai *n_estimators* didapat 402 dan nilai *max_depth* didapat 19. Hasil dari implementasi model didapat *confussion matrix* TP (True Positive): 12, FP (False Positive) : 59, TN (True Negative) : 1385, dan FN (False Negative) : 77. Hasil performa model didapat akurasi sebesar 91%, presisi sebesar 56%, *recall* sebesar 55%, dan *f*1-*score* sebesar 55%. Hasil dari *confussion matrix* dan *classification report* dari klasifikasi dengan tuning parameter menggunakan *RandomizedSearchCV* dan teknik Borderline-SMOTE1 pada pengujian 70:30 dapat dilihat pada gambar 4. 8 dan tabel 4. 17.

	precision	recall	f1-score	support
0	0.95	0.96	0.95	1444
1	0.17	0.13	0.15	89
accuracy			0.91	1533
macro avg	0.56	0.55	0.55	1533
weighted avg	0.90	0.91	0.91	1533

Gambar 4. 8 *Classification Report Random Forest* 70:30, *RandomizedSearchCV* Borderline-SMOTE1

Tabel 4. 17 *Confusion matrix Random Forest* 70:30, *RandomizedSearchCV* Borderline-SMOTE1

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	1385	59
True: Stroke	77	12

4.3.2 Hasil Pengujian 80:20

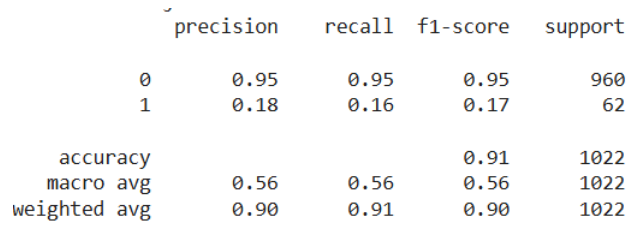
Pada pengujian ini, data minoritas yang ada pada data latih ditambahkan dengan data sintetis yang dibuat dengan teknik Borderline-SMOTE1. Berikut tabel 4. 18. yang merupakan perbedaan jumlah data minoritas sebelum dan sesudah penggunaan teknik Borderline-SMOTE1.

Tabel 4. 18 Hasil penambahan data sintetis 80:20, Borderline-SMOTE1

Kelas	**Sebelum**	**Sesudah**
0 (mayoritas)	3900	3900

1 (minoritas)	187	3900

Tuning parameter dengan menggunakan *GridSearchCV* didapat nilai n_estimators 100 dan nilai max_depth 10. Hasil implementasi model *Random Forest* pada pengujian ini didapat sebagaimana pada gambar 4. 9 dan tabel 4. 19. Hasil dari confussion matrix didapat TP (True Positive): 10, FP (False Positive) : 45, TN (True Negative) : 915, dan FN (False Negative) : 52. Untuk hasil performa model didapat akurasi sebesar 91%, presisi sebesar 56%, *recall* sebesar 56%, dan *f*1-*score* sebesar 56%.



	precision	recall	f1-score	support
0	0.95	0.95	0.95	960
1	0.18	0.16	0.17	62
accuracy			0.91	1022
macro avg	0.56	0.56	0.56	1022
weighted avg	0.90	0.91	0.90	1022

Gambar 4. 9 *Classification Report Random Forest* 80:20, *GridSearchCV* Borderline-SMOTE1

Tabel 4. 19 *Confusion matrix Random Forest* 80:20, *GridSearchCV* Borderline-SMOTE1

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	915	45
True: Stroke	52	10

Pada tuning parameter menggunakan *RandomizedSearchCV* nilai n_estimators dan max_depth yang didapat masing-masing sebesar 352 dan 18. Setelah didapat nilai n_estimators dan max_depth, implementasi model *Random Forest* dilakukan dan mendapatkan hasil *classification report* dan confussion matrix yang terdapat pada gambar 4.10 dan tabel 4. 20 dimana hasil confussion matrix didapat TP (True Positive): 9, FP (False Positive) : 34, TN (True Negative) : 926, dan FN (False Negative) : 53. Sedangkan hasil performa didapat akurasi sebesar 91%, presisi sebesar 58%, *recall* sebesar 55%, dan *f*1-*score* sebesar 56%.

	precision	recall	f1-score	support
0	0.95	0.96	0.96	960
1	0.21	0.15	0.17	62
accuracy			0.91	1022
macro avg	0.58	0.55	0.56	1022
weighted avg	0.90	0.91	0.91	1022

Gambar 4. 10 *Classification Report Random Forest* 80:20, *RandomizedSearchCV* Borderline-SMOTE1

Tabel 4. 20 *Confusion matrix Random Forest* 80:20, *RandomizedSearchCV* Borderline-SMOTE1

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	926	34
True: Stroke	53	9

4.3.3 Hasil Pengujian 90:10

Pengujian ini menambahkan data sintetis pada data minoritas yang ada pada data latih dengan teknik Borderline-SMOTE1. Berikut tabel 4. 21 yang menunjukkan perbedaan jumlah sebelum dan sesudah Borderline-SMOTE1.

Tabel 4. 21 Hasil penambahan data sintetis 90:10, Borderline-SMOTE1

Kelas	**Sebelum**	**Sesudah**
0 (mayoritas)	4377	4377
1 (minoritas)	221	4377

Hasil dari tuning parameter menggunakan *GridSearchCV* mendapatkan n_estimators 300 dan max_depth 15. Implementasi model *Random Forest* mendapatkan hasil confussion matrix TP (True Positive): 4, FP (False Positive) : 22, TN (True Negative) : 461, dan FN (False Negative) : 24. Hasil performa model mendapatkan akurasi sebesar 91%, presisi sebesar 55%, *recall* sebesar 55%, dan *f*1-*score* sebesar 55%. Hasil confussion matrix dan *classification report* pada pengujian 90:10 menggunakan tuning *GridSearchCV* dengan teknik Borderline-SMOTE1 terdapat pada gambar 4. 11 dan tabel 4. 22.

	precision	recall	f1-score	support
0	0.95	0.95	0.95	483
1	0.15	0.14	0.15	28
accuracy			0.91	511
macro avg	0.55	0.55	0.55	511
weighted avg	0.91	0.91	0.91	511

Gambar 4. 11 *Classification Report Random Forest* 90:10, *GridSearchCV* Borderline-SMOTE1

Tabel 4. 22 *Confusion matrix Random Forest* 90:10, *GridSearchCV* Borderline-SMOTE1

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	461	22
True: Stroke	24	4

Hasil dari tuning parameter menggunakan *RandomizedSearchCV* menghasilkan n_estimators 340 dan max_depth 19. Untuk *classification report* didapat hasil akurasi sebesar 91%, presisi sebesar 54%, *recall* sebesar 53%, dan *f*1-*score* sebesar 54%. Confussion matrix yang didapat setelah implementasi model TP (True Positive): 3, FP (False Positive) : 20, TN (True Negative) : 463, dan FN (False Negative) : 25. Dan Hasil dari *classification report* dan confussion matix terdapat pada gambar 4. 12 dan tabel 4. 23.

	precision	recall	f1-score	support
0	0.95	0.96	0.95	483
1	0.13	0.11	0.12	28
accuracy			0.91	511
macro avg	0.54	0.53	0.54	511
weighted avg	0.90	0.91	0.91	511

Gambar 4. 12 *Classification Report Random Forest* 90:10, *RandomizedSearchCV* Borderline-SMOTE1

Tabel 4. 23 *Confusion matrix Random Forest* 90:10, *RandomizedSearchCV* Borderline-SMOTE1

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	463	20
True: Stroke	25	3

4.3.4 Hasil Pengujian k=5

Pengujian ini dilakukan dengan menggunakan Kfold untuk pembagian data. Jumlah k yang dilakukan pada pengujian ini adalah 5. Pada pengujian ini juga dilakukan teknik balancing data menggunakan teknik Borderline-SMOTE1 dan dilakukan pada data latih sebelum tuning parameter dan implementasi model. Tuning parameter menggunakan *GridSearchCV* pada pengujian ini mendapatkan nilai terbaik n_estimators dan max_depth sebagaimana yang terdapat pada tabel 4. 24 berikut.

Tabel 4. 24 Hasil Tuning Parameter *GridSearchCV* pada k=5, Borderline-SMOTE1

k	**N_estimators**	**Max_depth**
k ke-1	300	15
k ke-2	200	13
k ke-3	300	10
k ke-4	100	13
k ke-5	300	13

Setelah melakukan tuning parameter dengan *GridSearchCV*, *Random Forest* diimplementasikan untuk melakukan klasifikasi pada dara stroke ini dan mendapatkan hasil akurasi pada setiap k dan rata-rata dari seluruh k yang terdapat pada tabel 4. 25 berikut.

Tabel 4. 25 Akurasi k=5 tuning parameter *GridSearchCV*, Borderline-SMOTE1

k	**Akurasi**
k ke-1	96%
k ke-2	95%
k ke-3	93%
k ke-4	95%
k ke-5	96%
Rata-rata seluruh k	95%

Tuning parameter *RandomizedSearchCV* juga dilakukan setelah teknik balancing Borderline-SMOTE1 dilakukan dan menghasilkan nilai terbaik dari

n_estimators dan max_depth pada setiap k. Berikut tabel 4. 26 hasil tuning parameter *RandomizedSearchCV* setiap k.

Tabel 4. 26 Hasil Tuning Parameter *RandomizedSearchCV* pada k=5, Borderline-SMOTE1

k	**N_estimators**	**Max_depth**
k ke-1	434	16
k ke-2	136	19
k ke-3	176	18
k ke-4	216	13
k ke-5	455	18

Setelah tuning parameter dilakukan, klasifikasi menggunakan *Random Forest* dilakukan dan menghasilkan akurasi pada tiap k. Hasil akurasi dari seluruh k dilakukan perhitungan rata-rata. Berikut hasil akurasi tiap k dan rata-rata-nya terdapat pada tabel 4. 27.

Tabel 4. 27 Akurasi k=5 tuning parameter *RandomizedSearchCV*, Borderline-SMOTE1

k	**Akurasi**
k ke-1	97%
k ke-2	95%
k ke-3	94%
k ke-4	95%
k ke-5	95%
Rata-rata seluruh k	96%

4.3.5 Hasil Pengujian k=10

Pada pengujian ini juga dilakukan pembagian data menggunakan Kfold dengan jumlah k=10. Sebelum pengimplementasian model *Random Forest*, dilakukan juga teknik balancing data pada data latih menggunakan Borderline-SMOTE1 dan tuning parameter menggunakan *GridSearchCV* dan *RandomizedSearchCV*. Hasil dari tuning parameter menggunakan *GridSearchCV* dapat dilihat pada tabel 4. 28 berikut.

Tabel 4. 28 Hasil Tuning Parameter *GridSearchCV* pada k=10, Borderline-SMOTE1

K	**N_estimators**	**Max_depth**
k ke-1	100	15
k ke-2	100	15
k ke-3	200	15
k ke-4	300	13
k ke-5	100	13
k ke-6	200	13
k ke-7	300	15
k ke-8	300	15
k ke-9	300	15
k ke-10	300	15

Setelah balancing data dan tuning parameter *GridSearchCV* dilakukan, implementasi *Random Forest* untuk mengklasifiaksi data dilakukan dan menghasilkan nilai akurasi untuk setiap k dan dilakukan perhitungan rata-rata nilai akurasi seluruh k. Berikut hasil akurasi dan rata-rata nya yang terdapat pada tabel 4. 29.

Tabel 4. 29 Akurasi k=10 tuning parameter *GridSearchCV*, Borderline-SMOTE1

k	**Akurasi**
k ke-1	96%
k ke-2	97%
k ke-3	96%
k ke-4	98%
k ke-5	97%
k ke-6	94%
k ke-7	96%
k ke-8	96%
k ke-9	97%
k ke-10	94%
Rata-rata seluruh k	96%

Setelah balancing data menggunakan Borderline-SMOTE1, dilakukan tuning parameter menggunakan *RandomizedSearchCV* dan menghasilkan nilai n_estimators dan max_depth sebagaimana yang terdapat pada tabel 4. 30.

Tabel 4. 30 Hasil Tuning Parameter *RandomizedSearchCV* pada k=10, Borderline-SMOTE1

k	**N_estimators**	**Max_depth**
k ke-1	114	15
k ke-2	294	14

k	N_estimators	Max_depth
k ke-3	431	12
k ke-4	355	19
k ke-5	214	18
k ke-6	379	13
k ke-7	295	19
k ke-8	374	14
k ke-9	437	17
k ke-10	441	14

Balancing data dan tuning parameter selesai, kemudian mengimplementasikan *Random Forest* untuk klasifikasi dan menghasilkan nilai akurasi untuk tiap k. Nilai akurasi tiap k dihitung rata-rata. Dan tabel 4.31 berikut merupakan hasil akurasi tiap k dan nilai rata-ratanya.

Tabel 4. 31 Hasil Tuning Parameter *RandomizedSearchCV* pada k=10, Borderline-SMOTE1

k	Akurasi
k ke-1	96%
k ke-2	97%
k ke-3	95%
k ke-4	98%
k ke-5	96%
k ke-6	95%
k ke-7	96%
k ke-8	97%
k ke-9	97%
k ke-10	94%
Rata-rata seluruh k	96%

4.4 Hasil Pengujian *Random Forest* dengan Borderline-SMOTE*2*

Pada pengujian ini menggunakan teknik Borderline-SMOTE*2* untuk menyeimbangkan data yang hanya menambahkan data sintetis pada data minoritas namun pengambilan sampel untuk data sintetisnya tidak hanya mengambil sampel data minoritas tapi juga data mayoritas.

4.4.1 Hasil Pengujian 70:30

Data minoritas pada pengujian ini ditambahkan dengan menggunakan teknik Borderline-SMOTE2, perbedaan sebelum dan sesudah teknik Borderline-SMOTE2 dapat dilihat pada tabel 4. 32 berikut.

Tabel 4. 32 Hasil penambahan data sintetis 70:30, Borderline-SMOTE2

Kelas	**Sebelum**	**Sesudah**
0 (mayoritas)	3416	3416
1 (minoritas)	160	3416

Parameter tuning menggunakan *GridSearchCV* menghasilkan nilai *n_estimators* 200 dan *max_depth* 15. Penggunaan *Random Forest* pada pengujian didapat hasil *confussion matrix* TP (True Positive) didapat 12, FP (False Positive) didapat 77, TN (True Negative) didapat 1385, dan FN (False Negative) didapat 59 sebagaimana pada tabel 4. 33. Hasil performa model *Random Forest* didapat akurasi sebesar 91%, presisi 56%, *recall* 55%, *f*1-*score* 55% sebagaimana pada gambar 4. 13 yang menunjukkan hasil dari *classification report*.

	precision	recall	f1-score	support
0	0.95	0.96	0.95	1444
1	0.17	0.13	0.15	89
accuracy			0.91	1533
macro avg	0.56	0.55	0.55	1533
weighted avg	0.90	0.91	0.91	1533

Gambar 4. 13 *Classification Report Random Forest* 70:30, *GridSearchCV* Borderline-SMOTE2

Tabel 4. 33 *Confusion matrix Random Forest* 70:30, *GridSearchCV* Borderline-SMOTE2

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	1374	70
True: Stroke	75	14

Untuk pengujian dengan tuning parameter menggunakan *RandomizedSearchCV*, n_estimators didapat sebesar 187 dan max_depth sebesar

18. Hasil implementasi model *Random Forest* didapat confussion matrix TP (True Positive): 11, FP (False Positive) : 43, TN (True Negative) : 1401, dan FN (False Negative) : 43. Untuk performa model didapatkan hasil akurasi sebesar 92%, presisi 58%, *recall* 55%, dan *f*1-*score* 56%. Hasil *classification report* dan confussion matrix terdapat pada gambar 4. 14 dan tabel 4. 34 berikut.

	precision	recall	f1-score	support
0	0.95	0.97	0.96	1444
1	0.20	0.12	0.15	89
accuracy			0.92	1533
macro avg	0.58	0.55	0.56	1533
weighted avg	0.90	0.92	0.91	1533

Gambar 4. 14 *Classification Report Random Forest* 70:30, *RandomizedSearchCV* Borderline-SMOTE2

Tabel 4. 34 *Confusion matrix Random Forest* 70:30, *RandomizedSearchCV* Borderline-SMOTE2

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	1401	43
True: Stroke	78	11

4.4.2 Hasil Pengujian 80:20

Pengujian *Random Forest* dengan pembagian data 80:20 yang meggunakan teknik Borderline-SMOTE2 ditambahkan data sintetis pada data minoritas yang ada pada data latih. Tabel 4. berikut merupakan perbedaan jumlah data sebelum dan sesudah menggunakan teknik Borderline-SMOTE2.

Tabel 4. 35 Hasil penambahan data sintetis 80:20, Borderline-SMOTE2

Kelas	**Sebelum**	**Sesudah**
0 (mayoritas)	3900	3900
1 (minoritas)	187	3900

Tuning parameter dengan *GridSearchCV* mendapatkan nilai n_estimators terbaik 200 dan nilai max_depth terbaik 15. Setelah *Random Forest* diimplementasikan, didapatkan hasil confussion matrix TP (True Positive): 9, FP

(False Positive) : 38, TN (True Negative) : 922, dan FN (False Negative) : 38. Performa model dari *Random Forest* yang diimplementasikan pada pengujian ini mendapatkan hasil akurasi sebesar 91%, presisi 57%, *recall* 55%, dan *f*1-*score* 56%. Hasil untuk pengujian 80:20 menggunakan tuning *GridSearchCV* dapat dilihat pada gambar 4. 15 dan tabel 4. 36.

	precision	recall	f1-score	support
0	0.95	0.96	0.95	960
1	0.19	0.15	0.17	62
accuracy			0.91	1022
macro avg	0.57	0.55	0.56	1022
weighted avg	0.90	0.91	0.91	1022

Gambar 4. 15 *Classification Report Random Forest* 80:20, *GridSearchCV* Borderline-SMOTE2

Tabel 4. 36 *Confusion matrix Random Forest* 80:20, *GridSearchCV* Borderline-SMOTE2

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	922	38
True: Stroke	53	9

Sedangkan pengujian 80:20 dengan tuning parameter *RandomizedSearchCV* mendapatkan nilai terbaik n_estimators sebesar 449 dan max_depth sebesar 17. Untuk hasil confussion matrix dan performa model setelah implementasi *Random Forest* terdapat pada gambar 4. 16 dan tabel 4. 37, dimana *classification report* mendapatkan hasil akurasi sebesar 92%, presisi 58%, *recall* 55%, dan *f*1-*score* 56%. Dan confussion matrix didapat hasil TP (True Positive): 8, FP (False Positive) : 28, TN (True Negative) : 932, dan FN (False Negative) : 54.

	precision	recall	f1-score	support
0	0.95	0.97	0.96	960
1	0.22	0.13	0.16	62
accuracy			0.92	1022
macro avg	0.58	0.55	0.56	1022
weighted avg	0.90	0.92	0.91	1022

Gambar 4. 16 *Classification Report Random Forest* 80:20, *RandomizedSearchCV* Borderline-SMOTE2

Tabel 4. 37 *Confusion matrix Random Forest* 80:20, *RandomizedSearchCV* Borderline-SMOTE2

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	932	28
True: Stroke	54	8

4.4.3 Hasil Pengujian 90:10

Pengujian pembagian data 90:10, data minoritas yang ada pada data latih ditambahkan data sintetis menggunakan teknik Borderline-SMOTE2. Pada tabel 4. 38 dapat terlihat perbedaan jumlah data yang belum dan yang sudah ditambahkan data sintetis.

Tabel 4. 38 Hasil penambahan data sintetis 90:10, Borderline-SMOTE2

Kelas	**Sebelum**	**Sesudah**
0 (mayoritas)	4377	4377
1 (minoritas)	221	4377

Tuning parameter menggunakan *GridSearchCV* mendapatkan nilai terbaik n_estimators 300 dan max_depth 15. Dan untuk hasil confussion matrix dan performa model setelah implementasi *Random Forest* terdapat pada gambar 4. 17 dan tabel 4. 39 di bawah. Confussion matrix yang didapat TP (True Positive): 4, FP (False Positive) : 18, TN (True Negative) : 465, dan FN (False Negative) : 24. Performa model menghasilkan akurasi sebesar 92%, presisi 57%, *recall* 55%, dan *f*1-*score* 56%.

	precision	recall	f1-score	support
0	0.95	0.96	0.96	483
1	0.18	0.14	0.16	28
accuracy			0.92	511
macro avg	0.57	0.55	0.56	511
weighted avg	0.91	0.92	0.91	511

Gambar 4. 17 *Classification Report Random Forest* 90:10, *GridSearchCV* Borderline-SMOTE2

Tabel 4. 39 *Confusion matrix Random Forest* 90:10, *GridSearchCV* Borderline-SMOTE2

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	465	18
True: Stroke	24	4

Dan untuk pengujian dengan tuning parameter menggunakan RandomizedSaerchCV mendapatkan nilai terbaik untuk n_estimators 236 dan max_depth 17. Setelah implementasi *Random Forest* dilakukan, dihasilkan akurasi sebesar 92%, presisi 57%, *recall* 55%, dan *f*1-*score* 56%. confussion matrix TP (True Positive): 4, FP (False Positive) : 17, TN (True Negative) : 466, dan FN (False Negative) : 24. Hasil dari *classification report* dan confussion matrix terdapat pada gambar 4. 18 dan tabel 4. 40 berikut.

	precision	recall	f1-score	support
0	0.95	0.96	0.96	483
1	0.19	0.14	0.16	28
accuracy			0.92	511
macro avg	0.57	0.55	0.56	511
weighted avg	0.91	0.92	0.91	511

Gambar 4. 18 *Classification Report Random Forest* 90:10, *RandomizedSearchCV* Borderline-SMOTE2

Tabel 4. 40 *Confusion matrix Random Forest* 90:10, *RandomizedSearchCV* Borderline-SMOTE2

	Pred: No Stroke	**Pred: Stroke**
True: No Stroke	466	17
True: Stroke	24	4

4.4.4 Hasil Pengujian k=5

Pengujian ini dilakukan dengan membagi data dengan menggunakan Kfold dengan jumlah k sebanyak 5. Sebelum implementasi *Random Forest* dilakukan, dilakukan teknik balancing data pada data latih dengan menggunakan Borderline-SMOTE2 dan dilakukan tuning parameter menggunakan *GridSearchCV*. Berikut tabel 4. 41 yang menunjukkan hasil tuning parameter pada setiap k.

Tabel 4. 41 Hasil Tuning Parameter *GridSearchCV* pada k=5, Borderline-SMOTE2

k	**N_estimators**	**Max_depth**
k ke-1	300	13
k ke-2	200	13
k ke-3	100	13
k ke-4	200	13
k ke-5	200	15

Dan hasil dari implementasi *Random Forest* pada pengujian ini menghasilkan nilai akurasi untuk setiap k dan nilai dari akurasi setiap k dihitung rata-rata. Hasil akurasi setiap k dan rata-rata akurasi seluruh k dapat dilihat pada tabel 4. 42 berikut.

Tabel 4. 42 Akurasi k=5 tuning parameter *GridSearchCV*, Borderline-SMOTE2

K	**Akurasi**
k ke-1	97%
k ke-2	96%
k ke-3	93%
k ke-4	94%
k ke-5	95%
Rata-rata seluruh k	95%

Sebelum tuning parameter *RandomizedSearchCV* dilakukan, balancing data dengan teknik Borderline-SMOTE2 juga dilakukan. Dan tuning parameter *RandomizedSearchCV* menghasilkan nilai n_estimators dan max_depth pada setiap k. Berikut tabel 4. 43 hasil tuning parameter *RandomizedSearchCV*.

Tabel 4. 43 Hasil Tuning Parameter *RandomizedSearchCV* pada k=5, Borderline-SMOTE2

K	**N_estimators**	**Max_depth**
k ke-1	225	19
k ke-2	308	11
k ke-3	408	16
k ke-4	371	12
k ke-5	493	15

Setelahnya dilakukan klasifikasi *Random Forest* dan setiap k menghasilkan nilai akurasi dan dilakukan perhitungan rata-rata seluruh nilai akurasi k. Berikut hasil akurasi tiap k dan hasil rata-rata seluruh nilai akurasi k terdapat pada tabel 4. 44.

Tabel 4. 44 Akurasi k=5 tuning parameter *RandomizedSearchCV*, Borderline-SMOTE2

K	**Akurasi**
k ke-1	96%
k ke-2	96%
k ke-3	93%
k ke-4	95%
k ke-5	95%
Rata-rata seluruh k	95%

4.4.5 Hasil Pengujian k=10

Pada pengujian ini juga dilakukan dengan membagi data menggunakan Kfold dengan jumlah k=10. Dilakukan juga balancing data menggunakan teknik Borderline-SMOTE2 dan tuning parameter menggunakan *GridSearchCV* juga *RandomizedSearchCV* sebelum melakukan klasifikasi dengan *Random Forest*. Hasil dari tuning parameter *GridSearchCV* setiap k terdapat pada tabel 4. 45 berikut.

Tabel 4. 45 Hasil Tuning Parameter *GridSearchCV* pada k=10, Borderline-SMOTE2

K	**N_estimators**	**Max_depth**
k ke-1	200	10
k ke-2	200	15
k ke-3	300	15
k ke-4	100	15
k ke-5	300	13
k ke-6	200	13

K	N_estimators	Max_depth
k ke-7	200	15
k ke-8	300	10
k ke-9	200	15
k ke-10	300	13

Hasil akurasi setelah dilakukan klasifikasi menggunakan *Random Forest* setelah melakukan balancing data dan tuning parameter dapat dilihat pada tabel 4. 46 berikut.

Tabel 4. 46 Hasil Tuning Parameter *GridSearchCV* pada k=10, Borderline-SMOTE2

K	Akurasi
k ke-1	96%
k ke-2	98%
k ke-3	96%
k ke-4	98%
k ke-5	94%
k ke-6	94%
k ke-7	97%
k ke-8	95%
k ke-9	98%
k ke-10	92%
Rata-rata seluruh k	96%

Selanjutnya tuning parameter menggunakan *RandomizedSearchCV* untuk mencari nilai n_estimators dan max_depth setelah balancing data dengan teknik Borderline-SMOTE2. Hasil dari tuning parameter *RandomizedSearchCV* sendiri dapat dilihat pada tabel 4. 47.

Tabel 4. 47 Hasil Tuning Parameter *RandomizedSearchCV* pada k=10, Borderline-SMOTE2

K	N_estimators	Max_depth
k ke-1	288	19
1151	151	14
k ke-3	100	16
k ke-4	280	19
k ke-5	410	17
k ke-6	344	15
k ke-7	300	19
k ke-8	299	25
k ke-9	492	19
k ke-10	301	18

Kemudian *Random Forest* diimplementasikan untuk klasifikasi dan menghasilkan nilai akurasi untuk setiap k. Dan seluruh nilai akurasi k akan dihitung rata-rata. Berikut hasil akurasi dari setiap k dan rata-rata seluruh k terdapat pada tabel 4. 48.

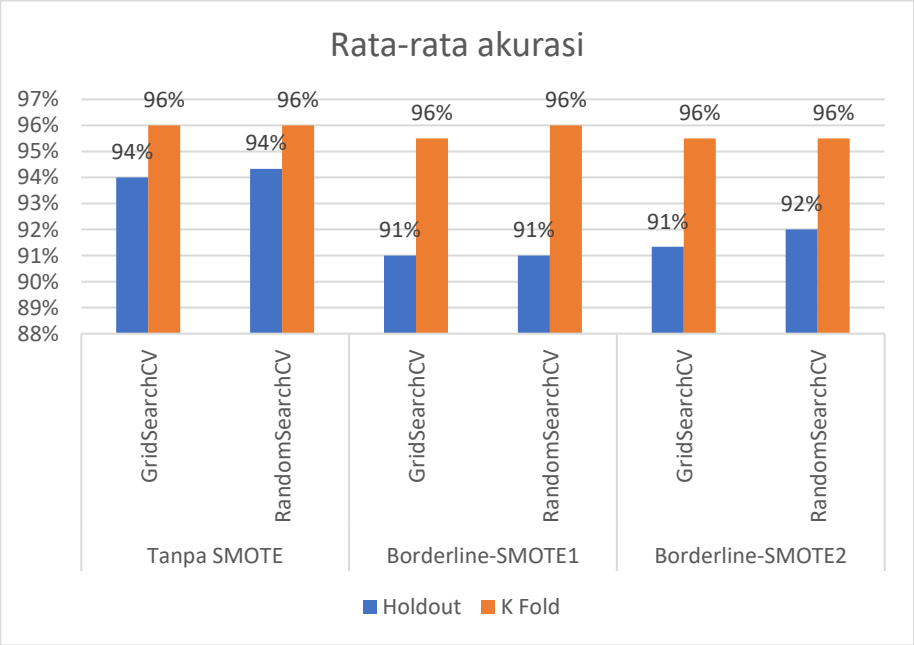
Tabel 4. 48 Hasil Tuning Parameter *RandomizedSearchCV* pada k=10, Borderline-SMOTE2

K	**Akurasi**
k ke-1	96%
k ke-2	98%
k ke-3	96%
k ke-4	98%
k ke-5	94%
k ke-6	95%
k ke-7	97%
k ke-8	95%
k ke-9	97%
k ke-10	92%
Rata-rata seluruh k	96%

4.5 Analisa Hasil Pengujian

Setelah semua pengujian klasifikasi menggunakan *Random Forest* dilakukan, baik tanpa teknik balancing, menggunakan teknik balancing Borderline-SMOTE1, dan menggunakan teknik balancing Borderline-SMOTE2 dilakukan analisa hasil pengujian.

Setelah didapat hasil akurasi pada setiap pembagian data, hasil akurasi pengujian dengan pembagian data *Hold-out* dihitung rata-rata sendiri dan hasil akurasi pengujian dengan pembagian data menggunakan *K-Fold cross validation* juga dihitung rata-rata sendiri. Sehingga hasil rata-rata akurasi terdapat sebagaimana pada gambar grafik berikut.



Gambar 4. 19 Grafik Rata-Rata Hasil Akurasi

Pada Gambar 4. 19 terlihat bahwa model mendapatkan hasil akurasi yang cukup baik. Namun pada pengujian *Random Forest* tanpa menggunakan teknik balancing meskipun hasil rata-rata akurasi yang didapat cukup baik, hasil *classification report* seperti presisi, recall, dan juga f1-score mendapatkan hasil yang kurang baik. *Confusion matrix* yang dihasilkan juga kurang baik, model kurang mampu dalam mengklasifikasi data yang tidak stroke (0) dan data stroke (1). Hal ini disebabkan karena data yang tidak seimbang, sehingga model *Random Forest* kurang mampu mengklasifikasi data yang tidak seimbang tanpa teknik balancing. Sedangkan pada pengujian menggunakan teknik balancing Borderline-SMOTE1 dan Borderline-SMOTE2, meskipun hasil rata-rata akurasi yang didapat lebih ada yang lebih rendah dari pengujian tanpa teknik balancing, namun hasil *classification report* yang lain dan hasil *confusion matrix* yang didapat lebih baik. Dengan menggunakan teknik balancing data, data yang tadinya tidak seimbang

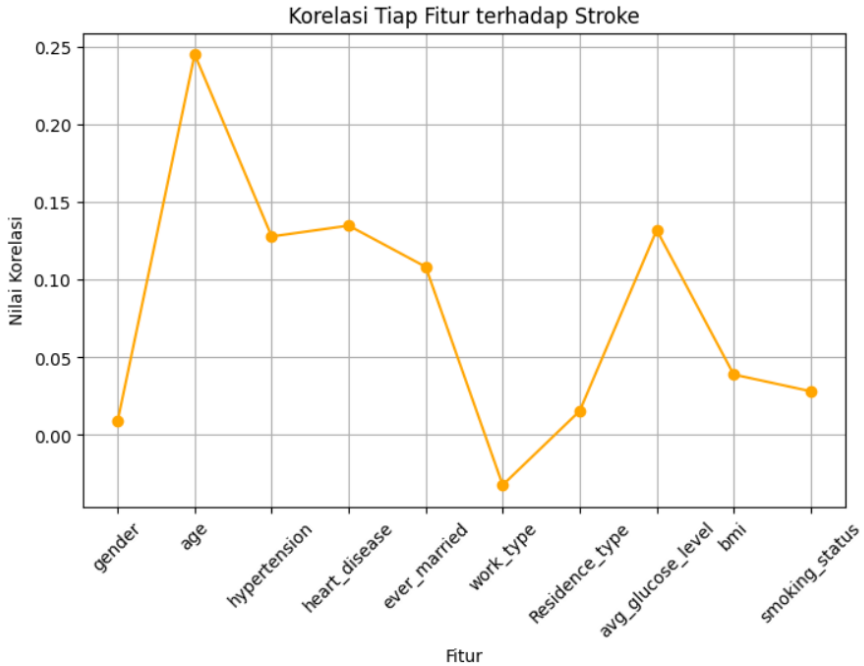
menjadi seimbang, sehingga model *Random Forest* mampu mengklasifikasi kelas tidak stroke (0) dan kelas stroke (1) lebih baik dari pengujian tanpa balancing data sebelumnya.

Selain itu juga pada pengujian Random Forest tanpa balancing data dengan pembagian data k=10, terdapat hasil akurasi yang paling rendah yaitu 87% pada k ke-6 baik menggunakan tuning parameter GridSearchCV maupun RandomizedSearchCV. Setelah dianalisa terdapat data outlier pada dataset, yaitu pada kolom bmi dan kolom avg_glucose_level, pada fold ke-6 jumlah data outlier lebih banyak dari fold lainnya sehingga hasil akurasi pada k ke-6 lebih rendah. Dan pada k ke-7 dengan percobaan tuning parameter RandomizedSearchCV juga mendapatkan hasil akurasi yang rendah yaitu 87%. Hal ini disebabkan karena pada fold ke-7 juga terdapat outlier pada kolom avg_glucose_level dan kolom bmi dan juga tuning parameter menggunakan RandomizedSearchCV kurang optimal dalam mencari nilai-nilai parameter yang terbaik daripada tuning parameter dengan GridSearchCV.

Pada grafik hasil akurasi pada Gambar 4. di atas terlihat bahwa kedua metode penyeimbangan data memberikan akurasi model yang cukup baik dalam mengklasifikasikan antara stroke dan tidak stroke pada dataset yang digunakan. Teknik Borderline-SMOTE1 menghasilkan kinerja model yang sedikit lebih baik dibandingkan Borderline-SMOTE2, yang menunjukkan bahwa model mampu mengidentifikasi kelas minoritas dengan lebih efektif. Hal ini disebabkan karena Borderline-SMOTE1 hanya mengambil sampel dari kelas minoritas, sedangkan

Borderline-SMOTE2 menciptakan data sintetis yang lebih agresif sehingga bisa terjadi overlap dan hasil performa model menjadi kurang (H. Han dkk., 2005).

Sebelumnya telah dilakukan preproses data sebelum melakukan pengujian. Pada pengujian ini kolom yang digunakan adalah kolom gender, age, hypertension, heart_disease, avg_glucose_level, bmi, dan stroke karena termasuk faktor medis. Selain itu kolom work_type, residence, ever_married, dan smoking_status juga digunakan karena termasuk sebagai faktor gaya hidup. Kolom id yang terdapat pada dataset tidak digunakan karena bukan termasuk salah satu faktor medis maupun faktor gaya hidup, kolom id hanya sebagai identitas pasien dan tidak berpengaruh pada klasifikasi. Gambar 4. 20 menunjukkan korelasi antar variabel baik kolom faktor medis maupun gaya hidup dengan kolom stroke dimana kolom stroke merupakan target klasifikasi.



Gambar 4. 20 Korelasi dengan Stroke

Sebagaimana pada gambar 4. 20 faktor yang paling memiliki korelasi paling tinggi terdapat pada faktor medis yaitu age atau umur dengan nilai 0,25 semakin tua umur, semakin tinggi risiko stroke dan untuk faktor medis lainnya hypertension, heart_disease, avg_glucose_level, bmi memiliki nilai yang hampir sama, sedangkan gender memiliki korelasi paling rendah karena stroke tidak memandang jenis kelamin. Pada gaya hidup yang memiliki korelasi paling tinggi yaitu ever_married sedangkan faktor gaya hidup lainnya rata-rata memiliki korelasi yang tidak terlalu tinggi.

Penelitian oleh Luh Ayu Martini, dkk (Martini, Luh Ayu dkk., 2025) menganalisis pengaruh oversampling pada model SVM untuk klasifikasi stroke dengan menggunakan dataset stroke yang sama dengan penelitian penulis. Penelitian ini menguji 4 teknik oversampling salah satunya yaitu Borderline-SMOTE, seleksi fitur, dan menggunakan model SVM dengan 3 kernel. Pembagian data dilakukan dengan menggunakan pembagian data *Hold-out*. Dan mengahasilkan kombinasi terbaik yaitu Borderline-SMOTE dan SVM dengan kernel RBF. Hasil akurasi didapat sebesar 96,86%. Hasil akurasi tersebut lebih besar jika dibandingkan dengan penelitian skripsi ini, namun meskipun begitu eksekusi data besar oleh model Random Forest lebih cepat karena proses pembentukan pohon lebih cepat daripada eksekusi dengan menggunakan model SVM yang memerlukan perhitungan kernel yang digunakan oleh penelitian sebelumnya.

Penelitian yang lain yang dilakukan oleh Nur Diana Saputri, dkk (Saputri dkk., 2022) juga dengan menggunakan dataset stroke yang sama namun dengan

model yang berbeda, mengkomparasi dua metode yaitu bagging dan adaboost pada algoritma C4.5 untuk data yang tidak seimbang. Dengan melakukan pembagian data menggunakan *K-Fold cross validation*, hasil akurasi yang didapat penelitiannya sebesar 92,87%, algoritma C4.5 dengan bagging sebesar 95,02%, dan algoritma C4.5 dengan adaboost sebesar 94,63%. Dan jika dibandingkan dengan algoritma C4.5 baik dengan bagging dan adaboost, model Random Forest yang digunakan pada peneltian skripsi ini mendapatkan hasil yang lebih tinggi.

4.6 Integrasi Islam

Dalam Islam, membedakan yang baik dan salah adalah suatu keharusan. Pernyataan tersebut selaras dengan penelitian ini yang melakukan klasifikasi dengan membedakan data antara benar stroke dan tidak stroke berdasarkan kategori yang ada. Sebagaimana dalam Al-Qur'an surat Al-Anfal ayat 37 yang berbunyi:

لِيَمِيزَ ٱللَّهُ ٱلۡخَبِيثَ مِنَ ٱلطَّيِّبِ وَيَجۡعَلَ ٱلۡخَبِيثَ بَعۡضَهُۥ عَلَىٰ بَعۡضٖ فَيَرۡكُمَهُۥ جَمِيعٗا فَيَجۡعَلَهُۥ فِي جَهَنَّمَۚ أُوْلَٰٓئِكَ هُمُ ٱلۡخَٰسِرُونَ

"Supaya Allah memisahkan (golongan) yang buruk dari yang baik dan menjadikan (golongan) yang buruk itu sebagiannya di atas sebagian yang lain, lalu kesemuanya ditumpukkan-Nya, dan dimasukkan-Nya ke dalam neraka Jahannam. Mereka itulah orang-orang yang merugi."(QS. Al anfal(8):37)

Menurut Zubdatut Tafsir Min Fathil Qadir, penggalan ayat لِيَمِيزَ ٱللَّهُ di atas menjelaskan bahwa Allah memisahkan golongan yang baik dan yang buruk, dimana golongan yang baik merupakan orang yang beriman dan golongan yang buruk adalah orang kafir. Sebagaimana Allah memisahkan dan mengelompokkan manusia menjadi baik dan buruk berdasarkan keimanan, data stroke juga dapat terkelompokkan menjadi stroke dan tidak stroke berdasarkan kategori-kategori

yang ada pada data tersebut dengan menggunakan metode klasifikasi, yang dapat menjadi pedoman dalam pengambilan keputusan medis.

Untuk mendukung klasifikasi yang tepat dan akurat, metode SMOTE digunakan dalam penelitian ini guna menyeimbangkan jumlah data antara kelas mayoritas dan minoritas. Dalam Islam, keseimbangan adalah salah satu prinsip fundamental dalam ciptaan Allah, sebagaimana disebutkan dalam QS. Ar-Rahman ayat 9 yang berbunyi:

وَأَقِيمُواْ ٱلۡوَزۡنَ بِٱلۡقِسۡطِ وَلَا تُخۡسِرُواْ ٱلۡمِيزَانَ

"Dan tegakkanlah timbangan itu dengan adil dan janganlah kamu mengurangi neraca itu." (QS. Ar-Rahman(55):9)

Menurut Tafsir As-Sa'di, penggalan ayat وَأَقِيمُواْ ٱلۡوَزۡنَ بِٱلۡقِسۡطِ yang memiliki arti

"Dan tegakkanlah timbangan itu dengan adil" menjelaskan bahwa Allah memerintahkan kita untuk selalu menegakkan keadilan. Pernyataan ini relevan dengan konsep SMOTE yang mencerminkan upaya menegakkan keadilan dalam data, agar model tidak cenderung "berat sebelah" terhadap kelompok mayoritas dan mengabaikan minoritas. Ini sejalan dengan semangat Islam dalam memperhatikan kelompok yang tertindas atau terpinggirkan.

Dalam implementasi klasifikasi, penelitian ini menggunakan algoritma *Random Forest*, sebuah bentuk teknologi kecerdasan buatan yang meniru prinsip musyawarah dalam Islam, di mana banyak pohon keputusan (*decision trees*) bekerja bersama untuk menghasilkan keputusan yang lebih akurat. Prinsip ini tercermin dalam QS. Asy-Syura ayat 38:

وَٱلَّذِينَ ٱسْتَجَابُواْ لِرَبِّهِمْ وَأَقَامُواْ ٱلصَّلَوٰةَ وَأَمْرُهُمْ شُورَىٰ بَيْنَهُمْ وَمِمَّا رَزَقْنَٰهُمْ يُنفِقُونَ

"Dan (bagi) orang-orang yang menerima (mematuhi) seruan Tuhannya dan mendirikan shalat, sedang urusan mereka (diputuskan) dengan musyawarat antara mereka; dan mereka menafkahkan sebagian dari rezeki yang Kami berikan kepada mereka." (QS. Asy-Syura(42): 38).

Pada penggalan ayat وَأَمْرُهُمْ شُورَىٰ بَيْنَهُم yang artinya "sedang urusan mereka (diputuskan) dengan musyawarat antara mereka" menurut Tafsir Al-Muyassar, Allah memerintahkan kita untuk melakukan musyawarah sebelum memutuskan atau melakukan sesuatu. Hal ini relevan dengan cara kerja *Random Forest* yang memutuskan hasil akhir dengan melakukan musyawarah atau voting dari pohon-pohon keputusan.

Adapun dalam mengevaluasi hasil klasifikasi, Islam sangat menjunjung tinggi keakuratan dan kesungguhan (itqan) dalam setiap pekerjaan. Rasulullah ﷺ bersabda:

وَسَلَّمَ: إِنّ اللَّهَ تَعَالى يُحِبّ إِذَا عَمِلَ أَحَدُكُمْ عَمَلاً أَنْ يُتْقِنَهُ (رواه الطبرني والبيهقي)

"*Dari Aisyah r.a., sesungguhnya Rasulullah s.a.w. bersabda: "Sesungguhnya Allah mencintai seseorang yang apabila bekerja, mengerjakannya secara itqan (professional)". (HR. Thabrani)*

Dalam penelitian ini, akurasi bukan sekadar angka, melainkan cerminan tanggung jawab ilmiah dan amanah terhadap kebenaran informasi. Semakin seimbang data (melalui SMOTE), maka potensi untuk mencapai akurasi tinggi juga meningkat, karena model tidak lagi bias terhadap satu kelompok. Hal ini menunjukkan bahwa keseimbangan (adl) dalam data merupakan jalan menuju

keakuratan (haq) dalam hasil, dua nilai utama yang sangat dijunjung tinggi dalam Islam.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pengujian klasifikasi dengan menggunakan model *Random Forest* pada stroke dengan menggunakan teknik balancing SMOTE dan tanpa SMOTE didapat kesimpulan berikut:

1. Pengujian model *Random Forest* tanpa dan dengan teknik SMOTE mendapatkan hasil akurasi yang baik. Hasil akurasi yang didapat baik tanpa maupun dengan teknik SMOTE didapat 96%.
2. Meskipun hasil akurasi tanpa teknik SMOTE baik, namun model Random Forest kurang mampu mengklasifikasi risiko stroke berdasakan faktor medis dan gaya hidup dengan baik data tidak seimbang karena hasil evaluasi model lain seperti presisi, recall, dan f1-score mendapatkan hasil yang tidak lebih tinggi daripada hasil evaluasi model dengan teknik SMOTE dan kurang mampu mengklasifikasi dengan baik kelas minoritas (1) dan kelas mayoritas (0).
3. Pada pengujian k=10 Random Forest tanpa SMOTE terdapat hasil akurasi terendah pada fold ke-6 dengan tuning parameter RandomizedSearchCV dan GridSearchCV karena terdapat data aneh atau outlier. Pada tuning parameter RandomizedSearchCV fold ke-7 juga mendapatkan akurasi rendah karena juga terdapat outlier dan

tuning parameter RandomizedSearchCV kurang optimal mencari nilai yang terbaik daripada menggunakan tuning parameter GridSearchCV.

4. Faktor yang memiliki korelasi paling tinggi dengan stroke yaitu faktor medis. Age memiliki korelasi paling tinggi dibanding faktor medis lainnya seperti hypertension, heart_disease, avg_glucose_level, bmi memiliki korelasi rata-rata. Sedangkan gender memiliki korelasi paling rendah.
5. Faktor gaya hidup kurang berkorelasi dengan stroke dibandingkan dengan faktor medis. Faktor gaya hidup yang memiliki korelasi paling tinggi yaitu ever_married, sedangkan faktor gaya hidup yang memiliki korelasi paling rendah yaitu work_type.
6. Perbandingan dengan dua penelitian terdahulu dengan dataset yang sama namun metode yang berbeda, penelitian ini mendapatkan hasil akurasi lebih tinggi dari penelitian terdahulu untuk pengujian model *Random Forest* pada pembagian data *K-Fold cross validation* baik dengan dan tanpa SMOTE. Pada pengujian model *Random Forest* pembagian data *Hold-out* didapat hasil akurasi yang lebih rendah dari penelitian terdahulu baik dengan maupun tanpa SMOTE.
7. Dibandingkan dengan model yang dilakukan dengan penelitian sebelumnya, model Random Forest yang digunakan pada penelitian skripsi ini mengeksekusi data dengan lebih cepat dari model penelitian sebelumnya.

5.2 Saran

Meskipun hasil akurasi pengujian model *Random Forest* dengan teknik Borderline-SMOTE1 dan Borderline-SMOTE2 mendapatkan hasil yang baik, namun performa model lain masih kurang baik. Sebagai tindak lanjut dari hasil penelitian yang dilakukan, penulis menyarankan beberapa hal yang dapat bermanfaat bagi penelitian selanjutnya. Adapun saran yang diajukan sebagai berikut:

1. Karena terdapat outlier pada data yang memengaruhi hasil akurasi, disarankan kepada penelitian selanjutnya bisa diatasi data outlier sebelum dilakukan implementasi model.
2. Disarankan kepada penelitian selanjutnya bisa menggunakan teknik balancing data lain seperti SMOTE-ENN, SMOTE Tomek Links, ADASYN, dan lain-lain untuk hasil performa model yang lebih baik.
3. Disarankan kepada penelitian selanjutnya bisa menggunakan model klasifikasi lain seperti KNN, Naive Bayes, XGBoost, dan lain-lain untuk hasil performa model yang lebih baik.

DAFTAR PUSTAKA

Airi, F. A. H., Suprapti, T., & Bahtiar, A. (2023). Komparasi Metode Klasifikasi Data Mining Untuk Prediksi Penyakit Stroke. *E-Link: Jurnal Teknik Elektro dan Informatika*, 1*8*(1), 73. https://doi.org/10.30587/e-link.v18i1.5271

Andriani, C., Herliani, O., Indahsari, N. K., & Masfufatun, M. (2024). Edukasi Pencegahan Stroke dan Penyakit Jantung Melalui Pemeriksaan Darah di Dupak Surabaya. *Jurnal Abdidas*, *5*(1), 39–46. https://doi.org/10.31004/abdidas.v5i1.881

Arifiyanti, A. A., & Wahyuni, E. D. (2020). SMOTE: Metode Penyeimbang Kelas Pada Klasifikasi Data Mining. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, 1*5*(1), 34–39. https://doi.org/10.33005/scan.v15i1.1850

Aulyra Familah, Arina Fathiyyah Arifin, Achmad Harun Muchsin, Mochammad Erwin Rachman, & Dahliah. (2024). Karakteristik Penderita Stroke Iskemik dan Stroke Hemoragik. *Fakumi Medical Journal: Jurnal Mahasiswa Kedokteran*, *4*(6), 456–463. https://doi.org/10.33096/fmj.v4i6.468

Azahra, R. C., Defitrika, F., & Ardaninggar, A. (2025). Pengaruh pola Konsumsi Cepat Saji terhadap Kesehatan Kardiovaskular pada Remaja. *Sulawesi Tenggara Educational Journal*, *5*(1), 291–298. https://doi.org/10.54297/seduj.v5i1.1110

Azhar, Y., Firdausy, A. K., & Amelia, P. J. (2022). Perbandingan Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Stroke. *SINTECH (Science and Information Technology) Journal*, *5*(2), 191–197. https://doi.org/10.31598/sintechjournal.v5i2.1222

Balatif, R., & Sukma, A. A. M. (2021). Memahami Kaitan Gaya Hidup dengan Kanker: Sebagai Langkah Awal Pencegahan Kanker. *SCRIPTA SCORE Scientific Medical Journal*, *3*(1), 40–50. https://doi.org/10.32734/scripta.v3i1.4506

Breiman, L. (2001). *Random Forest*s. *Machine Learning*, *45*, 5–32.https://doi.org/10.1023/A:1010933404324

Brivio, F., Viganò, A., Paterna, A., Palena, N., & Greco, A. (2023). Narrative Review and Analysis of the Use of “Lifestyle” in Health Psychology. *International Journal of Environmental Research and Public Health*, *20*(5), 4427. https://doi.org/10.3390/ijerph20054427

Budi, H., Bahar, I., & Sasmita, H. (2020). Faktor Risiko Stroke Pada Usia Produktif Di Rumah Sakit Stroke Nasional (rssn) Bukit Tinggi. *Jurnal Persatuan Perawat Nasional Indonesia (JPPNI)*, *3*(3), 129. https://doi.org/10.32419/jppni.v3i3.163

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Conners, C. K., & Erhardt, D. (1998). Attention-deficit Hyperactivity Disorder in Children and Adolescents. Dalam *Comprehensive Clinical Psychology* (hlm. 487–525). Elsevier. https://doi.org/10.1016/B0080-4270(73)00133-4

Fadmadika, F., Handayani, H. H., Mudzakir, T. A., & Indra, J. (2024). Pengaruh SMOTE Terhadap Performa Algoritma *Random Forest* Dan Algoritma Gradient Boosting Dalam Memprediksi Penyakit Stroke. *Jurnal Teknik Informasi dan Komputer (Tekinkom)*, *7*(2), 837. https://doi.org/10.37600/tekinkom.v7i2.1575

Feigin, V. L., Brainin, M., Norrving, B., Martins, S., Sacco, R. L., Hacke, W., Fisher, M., Pandian, J., & Lindsay, P. (2022). World Stroke Organization (WSO): Global Stroke Fact Sheet 2022. *International Journal of Stroke*, 17(1), 18–29. https://doi.org/10.1177/17474930211065917

Fitri Handayani & Reny Medikawati Taufiq. (2024). Komparasi Algoritma Menggunakan Teknik SMOTE Dalam Melakukan Klasifikasi Penyakit Stroke Otak. *Jurnal CoSciTech (Computer Science and Information Technology)*, *5*(2), 367–372. https://doi.org/10.37859/coscitech.v5i2.7439

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining* (Vol. 72). Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. Dalam D.-S. Huang, X.-P. Zhang, & G.-B. Huang (Ed.), *Advances in Intelligent Computing* (Vol. 3644, hlm. 878–887). Springer Berlin Heidelberg. https://doi.org/10.1007/11538059_91

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques (Third Edition)* (3 ed.). Morgan Kaufmann. https://www.sciencedirect.com/book/9780123814791/data-mining-concepts-and-techniques#book-description

Junaidi, I. (2011). *Stroke, waspadai ancamannya*. Penerbit Andi.

Kemenkes. (2018, Januari 1). *Hidup Sehat* [Post]. Ayo Sehat. https://ayosehat.kemkes.go.id/hidup-sehat

Kemenkes. (2023, Oktober 6). Kenali Stroke dan Penyebabnya. *ayosehat*. https://ayosehat.kemkes.go.id/kenali-stroke-dan-penyebabnya

Markaz Tafsir Riyadh. (1996). *Tafsir Al-Mukhtashar* (Vol. 6). Dar al-Ma'arif.

Martini, Luh Ayu, Pradipta, G. A., & Huizen, R. R. (2025). Analysis of the Impact of Data Oversampling on the Support Vector Machine Method for Stroke Disease Classification. *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, *7*(2), 404–421. https://doi.org/10.35882/jeeemi.v7i2.698

Mualfah, D., Fadila, W., & Firdaus, R. (2022). Teknik SMOTE untuk Mengatasi Imbalance Data pada Deteksi Penyakit Stroke Menggunakan Algoritma *Random Forest*. *Jurnal CoSciTech (Computer Science and Information Technology)*, *3*(2), 107–113. https://doi.org/10.37859/coscitech.v3i2.3912

Muhamad Malik Matin, I. (2023). Hyperparameter Tuning Menggunakan *GridSearchCV* pada *Random Forest* untuk Deteksi Malware. *MULTINETICS*, *9*(1), 43–50. https://doi.org/10.32722/multinetics.v9i1.5578

Mutmainah, S. (2021). Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi*, 1(1), 10–16. https://doi.org/10.20885/snati.v1i1.2

Oktafiani, R., Hermawan, A., & Avianto, D. (2023). Pengaruh Komposisi Split data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning. *Jurnal Sains dan Informatika*, 19–28. https://doi.org/10.34128/jsi.v9i1.622

Puspitasari, P. N. (2020). Hubungan Hipertensi Terhadap Kejadian Stroke. *Jurnal Ilmiah Kesehatan Sandi Husada*, 1*2*(2), 922–926. https://doi.org/10.35816/jiskh.v12i2.435

Putri, F. A. E. (2024). *Pengaruh Penanganan Ketidakseimbangan Kelas Pada Dataset Penyakit Stroke Terhadap Performa Algoritma Random Forest* [Skripsi]. Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Rahayu, T. G. (2023). Analisis Faktor Risiko Terjadinya Stroke Serta Tipe Stroke. *Faletehan Health Journal*, 1*0*(01), 48–53. https://doi.org/10.33746/fhj.v10i01.410

Rahmadania, S. R. (2024, September 17). Kasus Kematian Akibat Stroke di Indonesia Tinggi, Capai 300 Ribu Orang Per Tahun. *detikhealth*. https://health.detik.com/berita-detikhealth/d-7543633/kasus-kematian-akibat-stroke-di-indonesia-tinggi-capai-300-ribu-orang-per-tahun

Saif, Z. B., Sakib, N., Adnan, M., Hasan, Md. T., Nishat, M. M., Faisal, F., Shafiullah, A., & Ali, S. (2023). Sensorimotor Activity Patterns using Machine Learning: Assessing the Impact of Auditory Timing Perception and Comparing Different Algorithms. *2023 IEEE 8th International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, 1–7. https://doi.org/10.1109/ICRAIE59459.2023.10468111

Santika, A. A., Saragih, T. H., & Muliadi, M. (2023). Penerapan Skala Likert pada Klasifikasi Tingkat Kepuasan Pelanggan Agen Brilink Menggunakan *Random Forest*. *Jurnal Sistem dan Teknologi Informasi (JustIN)*, 11(3), 405. https://doi.org/10.26418/justin.v11i3.62086

Saputri, N. D., Khalid, K., & Rolliawati, D. (2022). Comparison of Bagging and Adaboost Methods on C4.5 Algorithm for Stroke Prediction. *SISTEMASI*, 11(3), 567. https://doi.org/10.32520/stmsi.v11i3.1684

Setiawan, A., Nasution, Z. H., Khairi, Z., Rahmaddeni, & Efrizoni, L. (2024). Klasifikasi Tingkat Risiko Diabetes Menggunakan Algoritma *Random Forest*. *Jurnal Informatika dan Rekayasa Elektronik*, *7*(2), 263–271. https://doi.org/10.36595/jire.v7i2.1259

Sharfina, N., & Ramadhan, N. G. (2023). Analisis SMOTE Pada Klasifikasi Hepatitis C Berbasis *Random Forest* dan Naïve Bayes. *JOINTECS (Journal of Information Technology and Computer Science)*, *8*(1), 33. https://doi.org/10.31328/jointecs.v8i1.4456

Siregar, Ary Prandika, Dwi Priyadi Purba, Jojor Putri Pasaribu, & Khairul Reza Bakara. (2023). Implementasi Algoritma *Random Forest* Dalam Klasifikasi Diagnosis Penyakit Stroke. *Jurnal Penelitian Rumpun Ilmu Teknik*, *2*(4), 155–164. https://doi.org/10.55606/juprit.v2i4.3039

SLN, F. (2024). *Buku Dasar Data Mining from A to Z*. https://www.researchgate.net/publication/377018853_Buku_Dasar_Data_Mining_from_A_to_Z_-_Feri_SLN_Free

Thalib, A. H. S., & Dimara, H. (2021). Efektifitas Mirror Therapy Terhadap Peningkatan Kekuatan Otot Pada Pasien Post Stroke: Literature Review. *IMJ (Indonesian Midwifery Journal)*, *5*(1), 11. https://doi.org/10.31000/imj.v5i1.6007

Wanjar, A., Siregar, M. N. H., Windarto, A. P., Hartama, D., Ginantara, N. L. W. S. R., Napitupulu, D., Negara, E. S., Lubis, M. R., Dewi, S. V., & Prianto, C. (2020). *Data Mining: Algoritma dan Implementasi* (Vol. 1). Yayasan Kita menulis. https://kitamenulis.id/2020/04/27/data-mining-algoritma-dan-implementasi/