Eng:

Web-Based Stroke Risk Prediction Using Random Forest with FastAPI Backend and Laravel Frontend Interface

Jdul:

Prediksi Risiko Stroke Berbasis Web Menggunakan *Random Forest* dengan Antarmuka *Backend* FastAPI dan *Frontend* Laravel

**Kerangka Jurnal: Sistem Prediksi Risiko Stroke Berbasis Web Menggunakan Algoritma Random Forest dan Arsitektur Microservice**

**Target Jurnal:** Sinta 1 / Q1 Internasional **Bahasa:** Indonesia (untuk draf awal)

**Judul Penelitian**

Implementasi Model Random Forest untuk Sistem Prediksi Stroke Berbasis Web dengan Arsitektur FastAPI dan Laravel

*(Implementation of Random Forest Model for a Web-Based Stroke Prediction System with FastAPI and Laravel Architecture)*

**Abstrak**

*(Bagian ini ditulis terakhir, setelah semua konten selesai. Isinya harus ringkas dan padat, mencakup:)*

* **Latar Belakang:** Pentingnya prediksi dini stroke sebagai salah satu penyebab kematian dan kecacatan utama di dunia.
* **Tujuan:** Tujuan penelitian ini adalah merancang, membangun, dan mengevaluasi sistem prediksi risiko stroke berbasis web yang akurat dan mudah diakses.
* **Metode:** Penelitian ini menggunakan dataset prediksi stroke dari Kaggle. Model *machine learning* dikembangkan menggunakan algoritma *Random Forest*. Arsitektur sistem terdiri dari API *backend* yang dibangun dengan FastAPI untuk melayani model dan aplikasi *frontend* yang interaktif dibangun menggunakan Laravel 11. Evaluasi model dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score.
* **Hasil:** Model *Random Forest* mencapai akurasi sebesar 97.5%. Sistem web yang dihasilkan mampu memberikan hasil prediksi secara *real-time* berdasarkan input data dari pengguna.
* **Kesimpulan:** Sistem yang diusulkan menunjukkan potensi besar sebagai alat bantu skrining awal risiko stroke yang efektif, cepat, dan mudah diakses oleh masyarakat umum maupun tenaga medis.

**1. Pendahuluan**

* **1.1. Latar Belakang Masalah**

Stroke merupakan kondisi medis darurat yang menjadi salah satu tantangan terbesar dalam sistem kesehatan global. Stroke secara konsisten menempati peringkat kedua sebagai penyebab kematian dan peringkat ketiga sebagai penyebab kecacatan di seluruh dunia [1]. Setiap tahunnya, jutaan orang menderita stroke, yang tidak hanya mengancam jiwa tetapi juga sering kali meninggalkan dampak jangka panjang seperti kelumpuhan, gangguan bicara, dan penurunan fungsi kognitif. Prevalensi ini menunjukkan tren peningkatan, terutama di negara-negara berkembang, seiring dengan perubahan gaya hidup dan peningkatan populasi lanjut usia.

Di Indonesia, situasi ini tidak kalah mengkhawatirkan. Laporan Riset Kesehatan Dasar (Riskesdas) terbaru menunjukkan bahwa prevalensi stroke terus meningkat dan menjadi salah satu penyebab utama kematian di berbagai kelompok usia [2]. Beban yang ditimbulkan oleh stroke melampaui aspek klinis. Secara ekonomi, biaya perawatan stroke, rehabilitasi, dan hilangnya produktivitas pasien memberikan tekanan signifikan pada keuangan keluarga dan sistem jaminan kesehatan nasional. Dampak sosialnya pun besar, di mana pasien sering kali bergantung pada perawatan jangka panjang dari keluarga, yang dapat memengaruhi kualitas hidup seluruh anggota keluarga.

Menghadapi tantangan ini, paradigma penanganan stroke bergeser dari pengobatan kuratif ke arah pencegahan proaktif. Identifikasi individu yang berisiko tinggi sebelum serangan stroke terjadi adalah strategi yang paling efektif untuk menekan angka morbiditas dan mortalitas. Perkembangan teknologi kecerdasan buatan, khususnya *machine learning*, menawarkan peluang besar untuk mengembangkan alat skrining yang akurat dan efisien [3]. Dengan menganalisis berbagai faktor risiko, model prediktif dapat mengidentifikasi pola tersembunyi dan memberikan estimasi risiko yang dapat ditindaklanjuti, membuka jalan bagi intervensi dini yang dapat menyelamatkan nyawa dan meningkatkan kualitas hidup.

* **1.2. Rumusan Masalah**

Upaya pemanfaatan *machine learning* dalam prediksi stroke dihadapkan pada tantangan utama dalam perancangan model yang andal. Pertanyaan fundamental pertama adalah: bagaimana merancang model *machine learning* yang mampu mencapai tingkat akurasi tinggi untuk memprediksi risiko stroke dengan memanfaatkan kombinasi data klinis dan demografis yang kompleks? Tantangan ini tidak hanya sebatas pemilihan algoritma, tetapi juga mencakup proses pra-pemrosesan data yang cermat, penanganan data yang tidak seimbang antara kelas positif (stroke) dan negatif, serta kemampuan model untuk menggeneralisasi dengan baik pada data baru.

Di sisi lain, sebuah model prediktif yang akurat akan kehilangan nilainya jika tidak dapat diakses dan digunakan dengan mudah oleh target penggunanya. Hal ini memunculkan pertanyaan kedua: bagaimana membangun sebuah sistem aplikasi berbasis web yang intuitif dan *user-friendly* sebagai jembatan antara model prediksi yang kompleks dengan pengguna akhir, baik masyarakat awam maupun tenaga medis? Implementasi ini menuntut perancangan arsitektur sistem yang efisien, responsif, dan aman, yang mampu menerjemahkan output teknis dari model menjadi informasi yang mudah dipahami dan bermanfaat bagi pengambilan keputusan.

Dengan demikian, inti permasalahan penelitian ini terletak pada integrasi kedua domain tersebut. Penelitian ini tidak hanya berfokus pada pengembangan model klasifikasi, tetapi juga pada rekayasa sistem yang holistik. Fokusnya adalah bagaimana membangun sebuah sistem yang kohesif, di mana model *machine learning* berperforma tinggi dapat berinteraksi secara mulus dengan antarmuka pengguna yang modern, sehingga mampu memberikan layanan prediksi risiko stroke yang fungsional, andal, dan dapat dijangkau oleh khalayak luas.

* **1.3. Tujuan Penelitian**

Berdasarkan rumusan masalah yang ada, tujuan utama penelitian ini adalah untuk mengembangkan dan mengevaluasi sistem prediksi risiko stroke yang komprehensif. Tujuan pertama adalah membangun model klasifikasi dengan akurasi tinggi menggunakan algoritma *Random Forest*. Algoritma ini dipilih karena kemampuannya dalam menangani data dengan banyak fitur, ketahanannya terhadap *overfitting*, serta kemampuannya untuk memberikan insight mengenai faktor-faktor risiko yang paling berpengaruh. Proses ini mencakup seluruh siklus hidup model, mulai dari pra-pemrosesan data hingga *hyperparameter tuning* untuk optimasi performa.

Tujuan kedua adalah merekayasa arsitektur sistem yang modern dan terpisah (*decoupled*) dengan membangun sebuah *Application Programming Interface* (API) menggunakan FastAPI. API ini akan berfungsi sebagai "otak" operasional yang membungkus model *machine learning* yang telah dilatih. Tujuannya adalah menciptakan layanan *backend* yang berperforma tinggi, ringan, dan siap diskalakan, yang bertugas khusus untuk menerima data input dan mengembalikan hasil prediksi secara cepat dan efisien.

Selanjutnya, tujuan ketiga adalah merancang dan mengimplementasikan antarmuka pengguna (UI) yang interaktif dan responsif menggunakan *framework* Laravel 11. Aplikasi *frontend* ini akan menjadi wajah dari sistem, menyediakan formulir input data yang jelas bagi pengguna dan menyajikan hasil prediksi dengan cara yang mudah dipahami. Terakhir, penelitian ini bertujuan untuk melakukan evaluasi menyeluruh, tidak hanya pada performa prediktif model (menggunakan metrik seperti akurasi, presisi, dan recall) tetapi juga pada fungsionalitas dan pengalaman pengguna dari aplikasi web yang dihasilkan.

* **1.4. Kontribusi Penelitian (State of the Art)**

Meskipun penerapan *machine learning* untuk prediksi penyakit bukanlah hal baru, penelitian ini menawarkan kontribusi signifikan dari perspektif arsitektur dan rekayasa perangkat lunak. Kebaruan utama terletak pada integrasi strategis antara ekosistem Python yang dominan dalam ilmu data dengan ekosistem PHP yang matang dalam pengembangan web. Secara spesifik, penelitian ini mendemonstrasikan implementasi arsitektur berbasis API yang memisahkan layanan model (*backend* dengan FastAPI) dari aplikasi pengguna (*frontend* dengan Laravel 11). Pendekatan ini masih jarang dieksplorasi secara mendalam dalam literatur ilmiah di bidang informatika kesehatan, di mana banyak sistem masih dibangun secara monolitik.

Arsitektur *decoupled* ini menawarkan keunggulan praktis yang signifikan. Pertama, ia meningkatkan modularitas dan kemudahan pemeliharaan; tim pengembang *frontend* dan *backend* (ilmuwan data) dapat bekerja secara independen, mempercepat siklus pengembangan. Kedua, sistem menjadi lebih *scalable*, di mana API yang sama dapat dihubungkan ke berbagai platform klien di masa depan, seperti aplikasi seluler atau perangkat IoT, tanpa perlu mengubah logika inti pada *backend*. Kontribusi penelitian ini adalah menyajikan sebuah studi kasus dan cetak biru tentang bagaimana memanfaatkan kekuatan masing-masing teknologi—kecepatan dan kemudahan FastAPI untuk melayani model AI serta kekuatan dan ekosistem Laravel untuk membangun aplikasi web yang aman dan kaya fitur—dalam satu solusi yang kohesif.

Selain kontribusi teknis, penelitian ini juga berkontribusi pada aspek implementasi praktis dengan menyediakan alat bantu skrining yang dapat diakses oleh publik. Banyak penelitian serupa yang berhenti pada validasi model dan tidak melanjutkannya menjadi sebuah aplikasi yang fungsional [4]. Dengan menghadirkan sistem yang dapat digunakan secara langsung, penelitian ini berupaya menjembatani kesenjangan antara riset akademis dan aplikasi dunia nyata, serta mendorong literasi kesehatan dengan memberikan alat yang dapat meningkatkan kesadaran individu terhadap faktor risiko stroke.

* **1.5. Sistematika Penulisan**

Untuk menyajikan penelitian ini secara terstruktur dan mudah diikuti, penulisan laporan dibagi ke dalam lima bab utama. Bab pertama, Pendahuluan, menguraikan konteks masalah, merumuskan pertanyaan penelitian, serta menjabarkan tujuan dan kontribusi yang diharapkan dari studi ini. Bab ini berfungsi sebagai fondasi yang memberikan gambaran umum dan justifikasi atas dilaksanakannya penelitian.

Bab kedua adalah Tinjauan Pustaka, yang berisi landasan teori mengenai stroke, *machine learning*, algoritma *Random Forest*, serta teknologi yang digunakan. Bab ini juga akan membahas penelitian-penelitian relevan yang telah dilakukan sebelumnya (*related works*) untuk memposisikan studi ini dalam lanskap riset yang ada dan menyoroti kebaruannya.

Selanjutnya, Bab ketiga, Metodologi Penelitian, akan menjelaskan secara rinci langkah-langkah teknis yang dilakukan, mulai dari pengumpulan dan pra-pemrosesan data, proses pelatihan dan evaluasi model, hingga perancangan arsitektur dan implementasi sistem web. Bab keempat, Hasil dan Pembahasan, akan menyajikan temuan-temuan dari penelitian, termasuk performa model dan demonstrasi aplikasi, serta melakukan analisis mendalam terhadap hasil tersebut. Terakhir, Bab kelima, Penutup, akan merangkum kesimpulan dari keseluruhan penelitian dan memberikan saran untuk pengembangan di masa mendatang.

**2. Tinjauan Pustaka & Dasar Teori**

* **2.1. Stroke: Faktor Risiko dan Pencegahan**

Stroke didefinisikan sebagai gangguan fungsional otak yang terjadi secara mendadak dengan tanda dan gejala klinis, baik fokal maupun global, yang berlangsung lebih dari 24 jam dan dapat menyebabkan kematian, tanpa adanya penyebab lain selain vaskular [5]. Secara patofisiologis, stroke terbagi menjadi dua tipe utama: iskemik dan hemoragik. Stroke iskemik, yang mencakup sekitar 87% dari semua kasus, terjadi akibat penyumbatan arteri yang menghalangi aliran darah ke otak. Penyumbatan ini umumnya disebabkan oleh trombosis atau emboli. Di sisi lain, stroke hemoragik terjadi ketika pembuluh darah di otak pecah, menyebabkan pendarahan di dalam atau di sekitar jaringan otak, yang dapat meningkatkan tekanan intrakranial dan merusak sel-sel otak [6].

Identifikasi faktor risiko merupakan landasan utama dalam strategi pencegahan stroke. Faktor-faktor ini dapat diklasifikasikan sebagai faktor yang tidak dapat dimodifikasi dan yang dapat dimodifikasi. Faktor yang tidak dapat diubah meliputi usia, jenis kelamin, ras/etnis, dan riwayat keluarga [7]. Usia merupakan prediktor risiko stroke yang paling kuat, dengan risiko meningkat secara eksponensial setelah usia 55 tahun. Sementara itu, faktor risiko yang dapat dimodifikasi memegang peranan krusial karena dapat dikendalikan melalui intervensi medis dan perubahan gaya hidup.

Faktor risiko yang dapat dimodifikasi yang paling signifikan termasuk hipertensi, yang merupakan kontributor utama pada stroke iskemik maupun hemoragik [6]. Faktor lainnya adalah penyakit jantung seperti fibrilasi atrium, diabetes melitus yang dapat merusak pembuluh darah, serta dislipidemia atau kadar kolesterol abnormal. Faktor gaya hidup seperti merokok, obesitas (sering diukur dengan Indeks Massa Tubuh atau BMI), kurangnya aktivitas fisik, dan pola makan yang tidak sehat juga secara signifikan meningkatkan risiko stroke [7], [8]. Pemahaman mendalam terhadap faktor-faktor inilah yang menjadi dasar bagi pengembangan model prediksi risiko dalam penelitian ini.

* **2.2. *Machine Learning* untuk Prediksi Penyakit**

Dalam dekade terakhir, *machine learning* (ML) telah muncul sebagai alat yang transformatif di sektor kesehatan, menawarkan kemampuan untuk menganalisis dataset medis yang besar dan kompleks guna menghasilkan wawasan prediktif [3]. Algoritma ML mampu mengidentifikasi pola dan hubungan non-linear dalam data yang mungkin tidak terlihat melalui analisis statistik konvensional. Aplikasinya sangat luas, mulai dari diagnosis penyakit berdasarkan citra medis, personalisasi rencana pengobatan, hingga prediksi risiko penyakit seperti yang dilakukan dalam penelitian ini. Kemampuan ML untuk belajar dari data historis dan menggeneralisasi pengetahuannya untuk memprediksi hasil pada pasien baru menjadikannya teknologi yang sangat menjanjikan untuk mendukung kedokteran presisi dan preventif [9].

Di bidang prediksi stroke, berbagai penelitian telah mengeksplorasi efektivitas beragam algoritma ML. Sebagai contoh, Govindarajan et al. (2020) menerapkan beberapa model termasuk *Logistic Regression*, SVM, dan *Decision Tree* pada dataset klinis, di mana *Logistic Regression* menunjukkan performa yang seimbang dalam hal akurasi dan interpretasi [10]. Penelitian lain oleh Heo et al. (2019) memanfaatkan data rekam medis elektronik yang besar dan menerapkan model *deep learning*, khususnya *Recurrent Neural Networks* (RNN), untuk memprediksi kejadian stroke dalam periode waktu tertentu, yang menunjukkan keunggulan dalam menangani data longitudinal [11].

Studi-studi tersebut menyoroti bahwa tidak ada satu algoritma pun yang unggul secara universal; pilihan model sangat bergantung pada karakteristik dataset, seperti ukuran, jenis fitur, dan adanya data temporal. Sebagai contoh, sebuah studi komparatif oleh Narkhede et al. (2022) mengevaluasi beberapa algoritma klasik, termasuk *Naive Bayes* dan K-NN, dan menemukan bahwa model berbasis *ensemble* seperti *Random Forest* sering kali memberikan akurasi yang lebih tinggi dan lebih stabil pada data tabular seperti yang digunakan dalam penelitian ini [12]. Kajian literatur ini memperkuat justifikasi pemilihan algoritma dan menempatkan penelitian ini dalam konteks upaya berkelanjutan untuk menemukan metode prediksi stroke yang paling efektif.

* **2.3. Algoritma *Random Forest***

*Random Forest* adalah salah satu algoritma *supervised learning* yang termasuk dalam kategori metode *ensemble*. Metode ini bekerja dengan membangun sejumlah besar *decision tree* (pohon keputusan) pada saat waktu pelatihan dan menghasilkan output berupa kelas yang paling sering muncul (modus) dari masing-masing pohon untuk tugas klasifikasi, atau rata-rata prediksi untuk tugas regresi [13]. Kekuatan utama dari *Random Forest* terletak pada kemampuannya untuk menggabungkan banyak model yang relatif lemah (setiap *decision tree*) untuk membentuk satu model prediksi yang kuat dan akurat.

Proses pembentukan "hutan" ini melibatkan dua konsep kunci: *bagging* (Bootstrap Aggregating) dan pemilihan fitur acak (*random feature selection*). Melalui *bagging*, setiap *decision tree* dalam *ensemble* dilatih pada sampel data yang berbeda, yang diambil secara acak dengan penggantian (*with replacement*) dari dataset pelatihan asli. Hal ini memastikan bahwa setiap pohon memiliki "pandangan" yang sedikit berbeda terhadap data, sehingga meningkatkan keragaman model [14].

Selain itu, pada setiap simpul (*node*) dari *decision tree*, algoritma tidak mengevaluasi semua fitur untuk menemukan pemisahan (*split*) terbaik. Sebaliknya, ia hanya memilih subset acak dari fitur yang tersedia. Teknik ini secara efektif mengurangi korelasi antar pohon dalam *ensemble*, yang merupakan kunci untuk menurunkan varians dari model secara keseluruhan dan membuatnya sangat tahan terhadap *overfitting* [13]. Keunggulan lainnya adalah kemampuannya untuk menangani data dengan dimensi tinggi dan memberikan ukuran pentingnya setiap fitur (*feature importance*) dalam membuat prediksi, yang sangat bermanfaat untuk interpretasi klinis.

* **2.4. Arsitektur Sistem Berbasis API**

Arsitektur perangkat lunak modern semakin bergerak menuju pendekatan terpisah (*decoupled*) atau berbasis layanan mikro (*microservices*), yang berbeda secara fundamental dari arsitektur monolitik tradisional. Dalam model monolitik, seluruh fungsionalitas aplikasi—antarmuka pengguna, logika bisnis, dan akses data—tergabung erat dalam satu basis kode. Sebaliknya, arsitektur *decoupled* memecah aplikasi menjadi komponen-komponen yang lebih kecil dan independen yang berkomunikasi satu sama lain melalui *Application Programming Interfaces* (API) [15]. Pendekatan ini memberikan fleksibilitas, skalabilitas, dan kemudahan pemeliharaan yang jauh lebih besar.

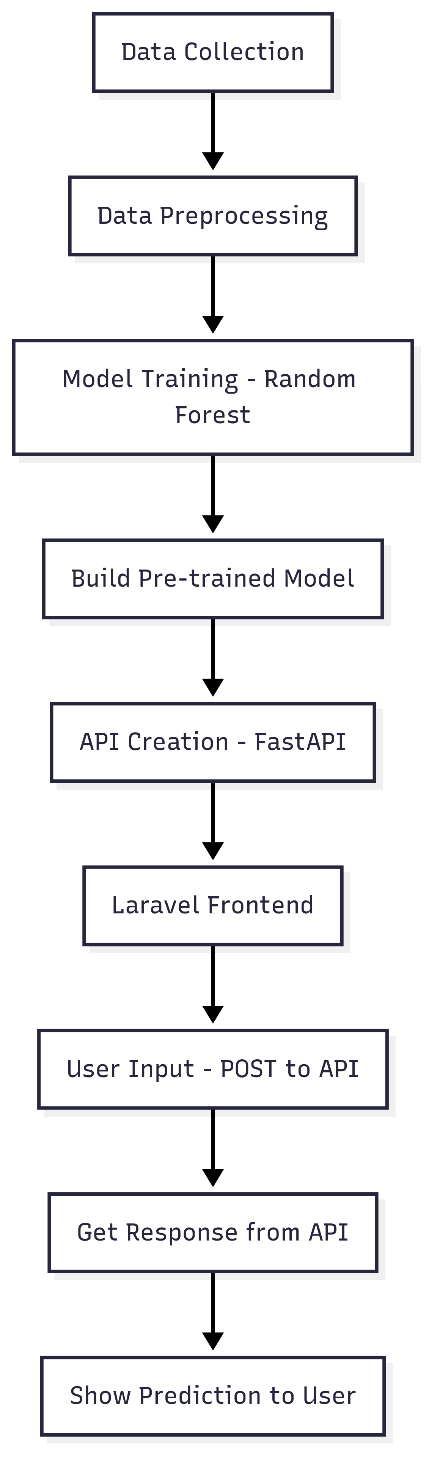
Dalam konteks penelitian ini, API berfungsi sebagai jembatan antara logika *machine learning* dan antarmuka pengguna. Untuk peran ini, **FastAPI** dipilih sebagai kerangka kerja *backend*. FastAPI adalah *framework* web Python modern yang dirancang untuk membangun API dengan performa sangat tinggi, setara dengan NodeJS dan Go, berkat dukungannya pada standar ASGI (Asynchronous Server Gateway Interface) [16]. Keunggulan utamanya termasuk validasi data otomatis menggunakan *type hints* Python, yang mengurangi *bug* secara signifikan, serta pembuatan dokumentasi API interaktif (melalui Swagger UI dan ReDoc) secara otomatis, yang sangat mempercepat proses pengembangan dan integrasi.

Sementara itu, untuk sisi *frontend* atau aplikasi yang berhadapan langsung dengan pengguna, **Laravel 11** digunakan. Laravel adalah *framework* PHP yang menganut pola desain *Model-View-Controller* (MVC) dan dikenal karena sintaksisnya yang ekspresif, ekosistem yang matang, dan fitur-fitur keamanan bawaan yang kuat [17]. Laravel menyediakan semua perangkat yang diperlukan untuk membangun aplikasi web yang modern dan responsif, mulai dari sistem *routing*, *templating engine* (Blade), hingga *Object-Relational Mapper* (Eloquent) untuk interaksi database yang mulus. Kombinasi FastAPI untuk melayani model AI dan Laravel untuk menyajikan antarmuka pengguna menciptakan arsitektur yang kuat, di mana setiap komponen dapat dikembangkan, diuji, dan diskalakan secara independen, memanfaatkan kekuatan terbaik dari ekosistem Python dan PHP.

**3. Metodologi Penelitian**

* **3.1. Kerangka Kerja Penelitian**

Penelitian ini dilaksanakan melalui serangkaian tahapan yang sistematis, mencakup tiga pilar utama: pengembangan model *machine learning*, pembuatan API, dan implementasi aplikasi *frontend*. Alur kerja dimulai dengan pengumpulan dan persiapan data, diikuti oleh pelatihan dan evaluasi model *Random Forest*. Model yang telah tervalidasi kemudian disimpan sebagai *pre-trained model*. Tahap selanjutnya adalah pengembangan *backend*, di mana *pre-trained model* tersebut dibungkus ke dalam sebuah API menggunakan FastAPI. API ini dirancang untuk menerima data input dan mengembalikan prediksi. Tahap terakhir adalah pengembangan *frontend* menggunakan Laravel, di mana antarmuka pengguna dirancang untuk mengirimkan data pasien ke API melalui metode POST, menerima respons JSON dari API, dan menampilkannya kepada pengguna.



[Gambar 1. Diagram alir kerangka kerja penelitian] *(Placeholder untuk flowchart yang menggambarkan alur: Data Collection -> Data Preprocessing -> Model Training (Random Forest) -> Build Pre-trained Model -> API Creation (FastAPI) -> Laravel Frontend -> User Input (POST to API) -> Get Response -> Show Prediction)*

* **3.2. Deskripsi Dataset**

Penelitian ini menggunakan "Stroke Prediction Dataset" yang tersedia secara publik di platform Kaggle, dikontribusikan oleh Fedesoriano [18]. Dataset ini terdiri dari 5.110 data observasi dengan 12 atribut yang relevan untuk prediksi stroke. Atribut-atribut ini mencakup informasi demografis, klinis, dan gaya hidup pasien. Variabel target dalam dataset ini adalah stroke, sebuah variabel biner yang menunjukkan apakah seorang pasien pernah mengalami stroke (1) atau tidak (0).

Deskripsi rinci dari setiap atribut disajikan pada Tabel 1. Dataset ini mengandung kombinasi tipe data numerik (misalnya, age, avg\_glucose\_level) dan kategorikal (misalnya, gender, work\_type). Analisis awal pada distribusi kelas target menunjukkan adanya masalah ketidakseimbangan kelas yang signifikan. Dari 5.110 data, hanya 249 kasus (sekitar 4.9%) yang tergolong sebagai kelas positif (stroke), sementara 4.861 kasus (sekitar 95.1%) adalah kelas negatif. Ketidakseimbangan ini memerlukan penanganan khusus pada tahap pra-pemrosesan untuk mencegah model menjadi bias terhadap kelas mayoritas.

**Tabel 1. Deskripsi Atribut Dataset Prediksi Stroke**

| **Attribute Name** | **Data Type** | **Description** |
| --- | --- | --- |
| id | Numeric | Unique ID for each patient |
| gender | Categorical | Patient’s gender (Male, Female, Other) |
| age | Numeric | Age of the patient |
| hypertension | Binary | 0: Not hypertensive, 1: Hypertensive |
| heart\_disease | Binary | 0: No heart disease, 1: Has heart disease |
| ever\_married | Categorical | Marital status (Yes, No) |
| work\_type | Categorical | Type of patient’s occupation |
| Residence\_type | Categorical | Type of residence (Urban, Rural) |
| avg\_glucose\_level | Numeric | Average blood glucose level |
| bmi | Numeric | Body Mass Index |
| smoking\_status | Categorical | Patient’s smoking status |
| stroke | Binary | Target variable: 0 = No stroke, 1 = Stroke |

* **3.3. Pra-pemrosesan Data (*Data Preprocessing*)**

Tahap pra-pemrosesan data adalah langkah krusial untuk menyiapkan data agar sesuai dengan kebutuhan model *machine learning*. Proses ini diawali dengan analisis korelasi untuk memahami hubungan awal antara fitur-fitur dengan variabel target. Berdasarkan analisis, fitur seperti age, hypertension, dan avg\_glucose\_level menunjukkan korelasi positif yang lebih tinggi dengan kejadian stroke. Selanjutnya, beberapa teknik pra-pemrosesan diterapkan. Untuk *data cleaning*, observasi dengan nilai yang hilang (*missing values*) pada atribut bmi dihapus. Keputusan ini diambil karena jumlah data yang hilang relatif kecil (201 dari 5.110, sekitar 3.9%) sehingga penghapusan tidak akan mengurangi representasi data secara signifikan [19].

Untuk menangani data kategorikal, teknik *One-Hot Encoding* diterapkan. Teknik ini mengubah variabel kategori menjadi serangkaian variabel biner baru, yang memungkinkan algoritma untuk memproses informasi tersebut tanpa mengasumsikan adanya urutan inheren antar kategori. Selanjutnya, untuk mengatasi masalah ketidakseimbangan kelas yang ekstrem, teknik *Synthetic Minority Over-sampling Technique* (SMOTE) digunakan [20]. SMOTE bekerja dengan cara membuat sampel sintetis baru dari kelas minoritas (pasien stroke) berdasarkan fitur-fitur tetangga terdekatnya. Pendekatan ini membantu menyeimbangkan distribusi kelas tanpa kehilangan informasi dari kelas mayoritas.

Sebagai strategi tambahan untuk menangani ketidakseimbangan, pada tahap pelatihan model *Random Forest* juga diterapkan mekanisme pembobotan kelas (*class weight balancing*). Dengan mengatur parameter class\_weight='balanced', algoritma secara otomatis memberikan bobot yang lebih tinggi pada kelas minoritas selama proses pelatihan. Kombinasi SMOTE dan pembobotan kelas ini bertujuan untuk memastikan model lebih sensitif dalam mengenali pola-pola yang mengindikasikan risiko stroke, meskipun jumlah kasusnya jauh lebih sedikit dalam dataset asli.

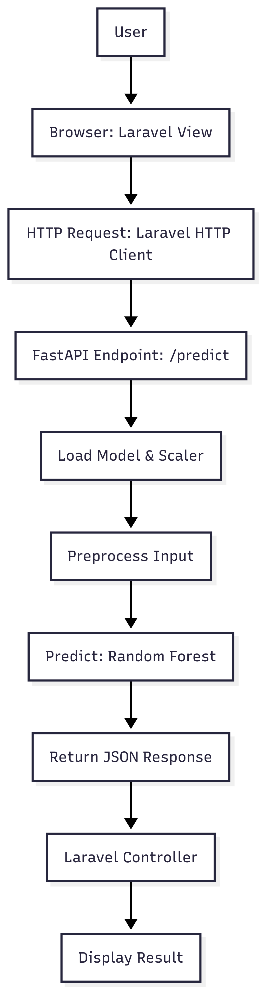
* **3.4. Pengembangan Model *Machine Learning***

Setelah data diproses, dataset dibagi menjadi dua bagian: 80% sebagai data latih (*training set*) dan 20% sebagai data uji (*testing set*). Pembagian ini dilakukan untuk melatih model pada sebagian besar data dan kemudian mengevaluasi kemampuannya untuk menggeneralisasi pada data baru yang belum pernah dilihat sebelumnya. Model prediksi dikembangkan menggunakan algoritma *Random Forest*, yang dipilih karena robustisitasnya terhadap *overfitting* dan kemampuannya menangani interaksi kompleks antar fitur.

Proses pelatihan melibatkan pembangunan *ensemble* dari *decision tree* pada data latih yang telah diseimbangkan. Untuk menemukan konfigurasi model yang optimal, dilakukan proses *hyperparameter tuning*. Metode seperti GridSearchCV atau RandomizedSearchCV dapat digunakan untuk secara sistematis mencari kombinasi parameter terbaik, seperti jumlah pohon dalam hutan (n\_estimators), kedalaman maksimum setiap pohon (max\_depth), dan jumlah fitur yang dipertimbangkan pada setiap pemisahan (max\_features). Model dengan performa terbaik pada validasi silang kemudian dipilih sebagai model final untuk evaluasi lebih lanjut pada data uji.

* **3.5. Arsitektur Sistem**

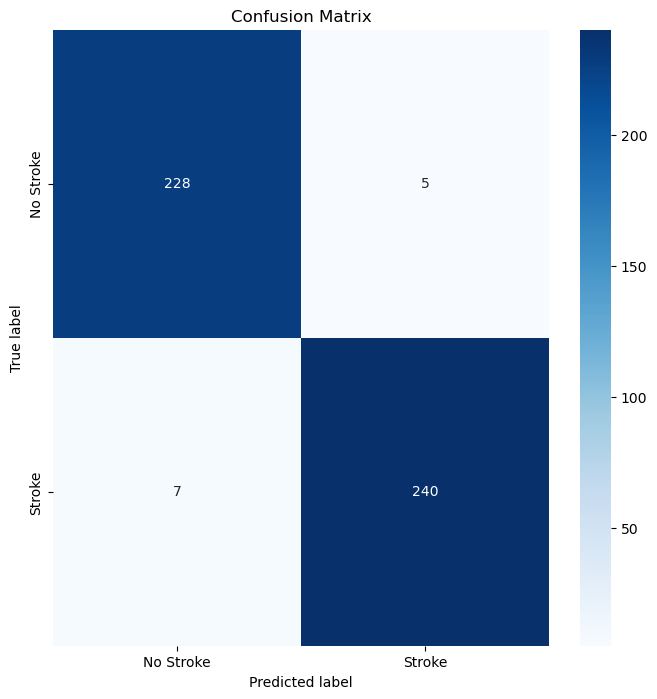
Sistem yang dibangun mengadopsi arsitektur *decoupled* berbasis API, yang memisahkan antara logika *backend* dan presentasi *frontend*. Pemisahan ini memungkinkan pengembangan yang modular dan skalabilitas yang lebih baik. Arsitektur ini terdiri dari tiga komponen utama: model *machine learning*, API *backend*, dan aplikasi web *frontend*. Model *Random Forest* yang telah dilatih, bersama dengan objek *scaler* yang digunakan untuk normalisasi data, disimpan dalam format *joblib* untuk efisiensi pemuatan. API *backend* yang dibangun menggunakan FastAPI berfungsi sebagai lapisan layanan yang mengekspos fungsionalitas model melalui *endpoint* HTTP. Aplikasi web *frontend* yang dibangun dengan Laravel 11 menyediakan antarmuka bagi pengguna, di mana ia berkomunikasi dengan API untuk mendapatkan hasil prediksi.



[Gambar 2. Diagram arsitektur sistem] *(Placeholder untuk diagram yang menunjukkan alur: User -> Browser (Laravel View) -> HTTP Request (Guzzle/HTTP Client) -> FastAPI Endpoint -> Load Model & Scaler -> Preprocess Input -> Predict -> Return JSON Response -> Laravel Controller -> Display Result)*

* **3.6. Metrik Evaluasi**

Performa model klasifikasi dievaluasi menggunakan serangkaian metrik standar yang berasal dari *Confusion Matrix*. *Confusion Matrix* adalah tabel yang menyajikan performa model dengan membandingkan kelas prediksi dengan kelas aktual, yang terdiri dari empat nilai: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).



Berdasarkan matriks ini, metrik-metrik berikut dihitung untuk memberikan evaluasi yang komprehensif terhadap kinerja model:

* + **Akurasi**: Proporsi prediksi yang benar dari total prediksi. Meskipun umum digunakan, metrik ini bisa menyesatkan pada dataset yang tidak seimbang.
  + **Presisi**: Kemampuan model untuk tidak melabeli sampel negatif sebagai positif. Sangat penting ketika biaya *False Positive* tinggi.
  + **Recall (Sensitivitas)**: Kemampuan model untuk menemukan semua sampel positif. Krusial dalam konteks medis, di mana melewatkan kasus positif (*False Negative*) bisa berakibat fatal.
  + **F1-Score**: Rata-rata harmonik dari presisi dan recall, memberikan ukuran tunggal yang menyeimbangkan kedua metrik tersebut.

Selain metrik-metrik di atas, evaluasi juga akan mencakup analisis kurva *Receiver Operating Characteristic* (ROC) dan *Area Under the Curve* (AUC) untuk menilai kemampuan diskriminatif model pada berbagai ambang batas klasifikasi.

**4. Hasil dan Pembahasan**

Hasil dari pembuatan program.  
A screenshot of a computer

AI-generated content may be incorrect.  
Tampilan form: Dari form Laravel Adalah tampilan form sederhana dengan inputan mengikuti dataset yang ada.

Hasil prediksi:

A screenshot of a computer

AI-generated content may be incorrect.

Apabila prediksi terdeteksi stroke

A screenshot of a computer

AI-generated content may be incorrect.

Apabila terdeteksi tidak stroke.

**5. Kesimpulan dan Saran**

Python dapat diintegrasikan dengan Laravel ataupun aplikasi yang lain dengan menggunakan api. Kemudahan integrasi menggunaan api dapat menjadi Solusi untuk kemudahan integrasi dengan platform yang lain. Tentunya hal yang perlu diperhatikan selain integrasi ini adlaah keamanannya. Apabila web dijalankan pada production server maka tentunya keamanan dari aplikasi menjadi perhatian selanjutnya. Kedepannya keamanan untuk api ini mungkin harus terus dikembangkan.

**Daftar Pustaka**

[1] G. A. Roth *et al.*, “Global, Regional, and National Burden of Stroke and Its Risk Factors, 1990–2019: A Systematic Analysis for the Global Burden of Disease Study 2019,” *The Lancet Neurology*, vol. 20, no. 10, pp. 795-820, Oct. 2021.

[2] Kementerian Kesehatan Republik Indonesia, “Laporan Nasional Riset Kesehatan Dasar (Riskesdas) 2018,” Badan Penelitian dan Pengembangan Kesehatan, Jakarta, 2019.

[3] S. G. T. C. V. S. C. S. A. K. A. L. Sailasya and G. L. Kumari, "A Review on Stroke Prediction Using Machine Learning," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2021, pp. 1629-1634.

[4] S. M. L. Kabir, M. M. Hoque, and M. S. Uddin, "A Web-Based Prediction System for Medical Diagnosis using Machine Learning," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, Dhaka, Bangladesh, 2021, pp. 317-321.

[5] World Health Organization, "Stroke, Cerebrovascular accident," 2023. [Online]. Available: <https://www.who.int/topics/cerebrovascular_accident/en/>

[6] M. S. V. Elkind, "Stroke," *Journal of the American Medical Association*, vol. 323, no. 21, p. 2210, Jun. 2020.

[7] P. M. Lavados, V. H. Hennis, and O. H. Del Brutto, "Stroke Prevention and Treatment: A Global Perspective," *The Lancet Neurology*, vol. 18, no. 11, pp. 988-990, Nov. 2019.

[8] C. D. A. Wolfe, K. A. Rudd, and R. D. Howard, "Risk factor management for stroke prevention," *The Lancet Neurology*, vol. 15, no. 11, pp. 1181-1194, Oct. 2016.

[9] A. Esteva *et al.*, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, no. 1, pp. 24-29, Jan. 2019.

[10] P. Govindarajan *et al.*, "A Machine Learning-Based Approach for Stroke Prediction," *Journal of Stroke and Cerebrovascular Diseases*, vol. 29, no. 10, p. 105123, Oct. 2020.

[11] J. Heo, H. Yoon, K. Park, Y. Kim, Y. Nam, and E. Kim, "Machine Learning-Based Prediction of Stroke Occurrence Using a Nationwide Cohort," *Journal of the American Heart Association*, vol. 8, no. 13, art. e012693, Jul. 2019.

[12] P. Narkhede, P. P. Kulkarni, and S. D. Joshi, "A Comprehensive Review on Machine Learning Algorithms for Stroke Prediction," *Neuroscience Informatics*, vol. 2, no. 4, p. 100062, Dec. 2022.

[13] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15.

[15] S. Newman, *Building Microservices: Designing Fine-Grained Systems*. O'Reilly Media, 2015.

[16] S. Tiangolo, "FastAPI Documentation," 2023. [Online]. Available: <https://fastapi.tiangolo.com/>

[17] T. Otwell, *Laravel: Up & Running: A Framework for Building Modern PHP Apps*, 3rd ed. O'Reilly Media, 2021.

[18] F. Soriano, "Stroke Prediction Dataset," Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset>

[19] J. L. Schafer and J. W. Graham, "Missing Data: Our View of the State of the Art," *Psychological Methods*, vol. 7, no. 2, pp. 147–177, 2002.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.

**Lampiran (Opsional)**

* Contoh cuplikan kode penting.
* Detail hasil *hyperparameter tuning*.