PERANCANGAN *ARTIFICIAL INTELEGENT* UNTUK MENGHITUNG PREDIKSI NILAI *LOW-DENSITY LIPOPROTEIN* (LDL) MENGGUNAKAN MODEL RANDOM FOREST

KERJA PRAKTIK

Program Studi
S1 Teknik Komputer

Oleh:
ZEFANYA SEPTIANUS VIDIANTO
22410200022

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2025

PERANCANGAN *ARTIFICIAL INTELEGENT* UNTUK MENGHITUNG PREDIKSI NILAI *LOW-DENSITY LIPOPROTEIN* (LDL) MENGGUNAKAN MODEL RANDOM FOREST

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Strata Satu (S1)

Disusun Oleh :

Nama	**:**	**Zefanya Septianus Vidianto**
Nim	**:**	**22410200022**
Program	**:**	**S1 (Strata Satu)**
Jurusan	**:**	**Teknik Komputer**

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA

2025

"Kegagalan bukanlah akhir, namun awal perjalanan."

LEMBAR PENGESAHAN

Perancangan *Artificial Intelegent* Untuk Menghitung Prediksi Nilai *Low-Density Lipoprotein* (LDL) Menggunakan Model Random Forest

Laporan Kerja Praktik oleh

Zefanya Septianus Vidianto

NIM: 22410200022

Telah diperiksa, diuji, dan disetujui

Surabaya, 16 Juli 2025

Disetujui:

Pembimbing	Penyelia
Heri Pratikno, M.T.	**Sofan Wahyudi**
NIDN. 0716117302	

Mengetahui,
Ketua Program Studi Teknik Komputer

Pauladie Susanto, S.Kom., M.T.
NIDN. 0729047501

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama	: **Zefanya Septianus Vidianto**
NIM	: **22410200022**
Program Studi	: **S1 Teknik Komputer**
Fakultas	: **Fakultas Teknologi dan Informatika**
Jenis Karya	: **Laporan Kerja Praktik**
Judul Karya	: **PERANCANGAN *ARTIFICIAL INTELEGENT* UNTUK MENGHITUNG PREDIKSI NILAI *LOW-DENSITY LIPOPROTEIN* (LDL) MENGGUNAKAN MODEL RANDOM FOREST**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (***Non-Exclusive Royalty Free Right***) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (***database***) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikan surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 16 Juli 2025

Zefanya Septianus Vidianto
NIM : 22410200022

ABSTRAK

Kerja Praktik ini dilakukan di PT. Wahana Meditek Indonesia (AdamLabs) dan berfokus pada pengembangan sistem prediksi nilai LDL (*Low-Density Lipoprotein*) menggunakan algoritma *machine learning* Random Forest. Pengolahan data ini memanfaatkan parameter laboratorium seperti total kolesterol, HDL, dan trigliserida sebagai *input*. Tujuan dari proyek ini adalah menciptakan solusi berbasis AI yang dapat meningkatkan akurasi prediksi nilai LDL dibandingkan perhitungan manual seperti rumus Friedewald. Hasil implementasi menunjukkan bahwa model Random Forest mampu memberikan hasil prediksi yang mendekati nilai aktual, dengan nilai koefisien determinasi (R2) sebesar 0.9965211560161251 setelah dilakukan *tuning hyperparameter*. Proyek ini mendemonstrasikan bahwa teknologi *machine learning* dapat diintegrasikan secara efektif dalam sistem informasi laboratorium untuk mendukung pengambilan keputusan klinis.

Kata Kunci: LDL, *Machine Learning*, Random Forest, Laboratorium, Kesehatan Digital

KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat-Nya, sehingga laporan Kerja Praktik di PT. Wahana Meditek Indonesia – AdamLabs ini dapat diselesaikan dengan baik. Laporan ini menjadi dokumentasi pengalaman dan tanggung jawab saya selama magang sebagai *Datalogger*, yang membuka wawasan lebih dalam mengenai dunia kerja, khususnya di bidang teknologi informasi kesehatan dan penerapan AI dalam pengolahan data laboratorium.Penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa yang memberikan rahmat, hidayah, dan kesempatan yang diberikan sehingga penulis dapat menyelesaikan program Dinamika *Industrial Internship* (DII) dan Kerja Praktik dengan baik.
2. Bapak Heri Pratikno, M.T., selaku dosen pembimbing yang telah memberikan arahan, dukungan, serta masukan selama proses pelaksanaan magang hingga penyusunan laporan ini.
3. Bapak Wigananda Firdaus Putra Aditya, S.Kom., sebagai penyelenggara program Dinamika *Industrial internship* (DII) yang mana dapat pada akhirnya saya melakukan konversi untuk kerja praktik
4. Bapak Sofan Wahyudi, selaku mentor dalam program ini dan bagian R&D dan *Datalogger*, atas bimbingan, kerjasama, dan kesempatan yang diberikan kepada saya.
5. Rekan-rekan mahasiswa serta seluruh pihak yang telah memberikan semangat dan bantuan selama masa magang.

Oleh karena itu, segala kritik dan saran yang membangun sangat saya harapkan demi perbaikan di masa yang akan datang. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pihak-pihak yang berkepentingan.

Surabaya, 18 Juli 2025

Zefanya Septianus Vidianto

DAFTAR ISI

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi di bidang kesehatan, khususnya dalam sistem informasi laboratorium, telah membuka peluang untuk meningkatkan efisiensi dan akurasi layanan medis. Salah satu pendekatan yang mulai banyak diterapkan adalah pemanfaatan kecerdasan buatan (AI) dan *machine learning* untuk analisis data laboratorium. Di tengah tantangan dalam mendeteksi penyakit kardiovaskular secara dini, salah satu indikator penting yang perlu dimonitor adalah kadar LDL (*Low-Density Lipoprotein*), yaitu kolesterol jahat yang memiliki kaitan erat dengan risiko penyumbatan pembuluh darah.

Pengukuran nilai LDL umumnya dilakukan dengan metode kimiawi atau menggunakan rumus estimasi seperti Friedewald. Namun, metode tersebut memiliki keterbatasan, terutama ketika kadar trigliserida sangat tinggi atau data laboratorium tidak lengkap (Friedewald et al., 1972). Oleh karena itu, dibutuhkan pendekatan alternatif yang lebih fleksibel dan adaptif, salah satunya melalui implementasi model *machine learning*, seperti Random Forest, untuk melakukan prediksi nilai LDL berdasarkan parameter masukan seperti Total Kolesterol, HDL, dan Trigliserida.

Melalui program kerja praktik ini, dilakukan implementasi dan evaluasi model Random Forest dalam lingkungan kerja nyata di PT. Wahana Meditek Indonesia (AdamLabs). Perusahaan ini bergerak di bidang teknologi informasi kesehatan dan memiliki fokus dalam pengembangan sistem informasi laboratorium yang terintegrasi dengan AI.

1.2 Rumusan Masalah

Dari latar belakang, maka dapat perumusan masalah sebagai berikut:

1. Bagaimana membangun sistem prediksi nilai LDL menggunakan model Random Forest berdasarkan parameter laboratorium?
2. Bagaimana tingkat akurasi dan performa model AI jika dibandingkan dengan metode manual seperti rumus Friedewald?
3. Bagaimana sistem prediksi AI ini dapat diintegrasikan dengan sistem

backend laboratorium?

1.3 Batasan Masalah

Batasan masalah pada pelaksanaan Kerja Praktik ini adalah sebagai berikut:

1. Data yang digunakan terdiri dari parameter Total Kolesterol, HDL, dan Trigliserida.
2. Model AI yang digunakan adalah Random Forest, dengan perbandingan terhadap perhitungan manual Friedewald.
3. Lingkup sistem terbatas pada pemrosesan data prediktif LDL dan tidak mencakup visualisasi *frontend*.
4. Evaluasi model dilakukan berdasarkan nilai akurasi, R2 score, dan RMSE (*Root Mean Square Error*).

1.4 Tujuan

Tujuan Kerja Praktik di AdamLabs adalah sebagai berikut:

1. Membangun model prediksi nilai LDL berbasis *machine learning* dengan algoritma Random Forest.
2. Mengevaluasi performa model prediksi terhadap hasil perhitungan manual.
3. Mengintegrasikan model prediksi dengan sistem *backend* perusahaan untuk kebutuhan pengolahan data laboratorium.

1.5 Manfaat

Pengembangan ini memiliki berbagai manfaat, baik bagi pengguna maupun lingkungan perusahaan itu sendiri, antara lain:

1. Menyediakan pendekatan alternatif dalam estimasi kadar LDL yang lebih fleksibel dan akurat.
2. Meningkatkan efisiensi proses laboratorium dengan pemanfaatan model prediksi berbasis AI.
3. Memberikan kontribusi dalam pengembangan sistem informasi kesehatan yang berbasis data dan teknologi cerdas.
4. Menambah pengalaman serta wawasan praktis bagi mahasiswa dalam menerapkan *machine learning* di dunia industri.

BAB II
GAMBARAN UMUM PERUSAHAAN

2.1 Sejarah Singkat Perusahaan

PT. Wahana Meditek Indonesia, melalui produk dan layanan dengan merek AdamLabs, merupakan perusahaan rintisan (*startup*) berbasis teknologi informasi kesehatan (*healthtech*) yang berdiri sejak tahun 2018 dan berlokasi di Eastern Park AB, Jl. Raya Sukolilo Mulia No.Ruko B23, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111. AdamLabs dikembangkan untuk menjawab kebutuhan akan sistem informasi laboratorium dan klinik yang efisien, terintegrasi, serta mampu menangani data dalam skala besar secara akurat dan cepat. Sejak awal pendiriannya, perusahaan ini memiliki misi untuk mendigitalisasi proses operasional di bidang kesehatan melalui sistem yang adaptif dan inovatif.

Dalam perjalanannya, AdamLabs telah menghadirkan berbagai solusi digital seperti ADAMLIS (*Laboratory Information System*), ADAMEDS (*Clinic Information System*), ADAMEDSPRO (*Hospital Information System*), hingga ADAMPACS (sistem teleradiologi). Selain itu, AdamLabs juga mengembangkan produk pendukung seperti AdamExpertise dan AdamQue, yang mendukung layanan monitoring dokter dan antrean pasien secara *real-time*. Inovasi-inovasi tersebut menjadikan AdamLabs sebagai salah satu pionir dalam digitalisasi layanan laboratorium dan klinik di Indonesia

2.2 Visi, Misi dan Tujuan Perusahaan

2.2.1 Visi

Dengan dukungan tim yang berpengalaman, kami berupaya memenuhi semua kebutuhan industri kesehatan dengan pendekatan Ekosistem Teknologi kesehatan *Healthtech*, sehingga pelayanan kesehatan menjadi lebih baik. Integrasi adalah kata yang menggambarkan visi kita untuk masa depan.

2.2.2 Misi

Sebagai penyedia layanan Sistem Informasi kesehatan untuk Rumah Sakit, Klinik, Laboratorium, Radiologi, yang terintegrasi, kami berharap dapat bekerja sama dengan Anda untuk meningkatkan kualitas layanan kesehatan di Indonesia

dengan pengembangan yang berkelanjutan.

2.2.3 Tujuan

Visi dan Misi Kami untuk Layanan Kesehatan yang Lebih Baik

2.3 Struktur Perusahaan

Gambar 2. 1 Struktur Perusahaan

Pada gambar diatas merupakan struktur perusahaan yang ada di AdamLabs

2.4 Informasi Kontak Perusahaan

Tempat	: Ruko Eastern Park AB, Jl. Raya Sukolilo Mulia No. B23, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111
Email	: support.marketing@adamlabs.id
Website	: Adamlabs.id
No Telfon	: 0882-0104-53808
Sosial Media	:
Facebook	: Adamlabs
TikTok	: adamlabs.id
Instagram	: @adamlabs.id

BAB III
LANDASAN TEORI

3.1 Konsep LDL dan Perhitungan Manual

Low-Density Lipoprotein (LDL) merupakan jenis kolesterol jahat dalam darah yang berperan besar dalam proses aterosklerosis, yaitu penumpukan plak di dinding arteri yang dapat menyebabkan penyakit jantung koroner. Oleh karena itu, pengukuran nilai LDL menjadi parameter penting dalam diagnosis dan pemantauan kondisi kardiovaskular.

Salah satu metode perhitungan nilai LDL yang paling banyak digunakan adalah rumus Friedewald:

$$\text{LDL} = \text{Total Kolesterol} - \text{HDL} - (\text{Trigliserida} / 5) \quad (3.1)$$

Namun, metode rumus 3.1 ini memiliki keterbatasan, terutama pada kondisi hiperlipidemia dengan kadar trigliserida tinggi atau saat data parameter tidak lengkap (Martin et al., 2013).

3.2 *Machine Learning* dan Random Forest

Machine learning (ML) adalah cabang dari kecerdasan buatan (AI) yang memungkinkan sistem belajar dari data untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit. ML terbagi menjadi beberapa jenis, dan salah satu yang digunakan dalam prediksi nilai LDL adalah *supervised learning* dengan model Random Forest.

Random Forest merupakan *ensemble* model yang terdiri dari sejumlah pohon keputusan (*decision tree*) yang dilatih secara paralel. Hasil akhir dari prediksi adalah rata-rata (untuk regresi) atau voting (untuk klasifikasi) dari semua pohon dalam *ensemble*. Keunggulan Random Forest antara lain: toleransi terhadap *overfitting*, kemampuannya menangani fitur yang banyak dan kompleks, serta akurasi yang tinggi pada berbagai kasus prediksi (Breiman, 2001; Zhang et al., 2019).

Dalam konteks kesehatan, Random Forest telah banyak digunakan untuk

prediksi nilai laboratorium, klasifikasi penyakit, dan analisis risiko pasien. Model ini sangat sesuai untuk sistem prediksi LDL karena mampu menangani data heterogen dan non-linear.

3.3 Dataset dan Fitur Prediktor

Dalam konteks penerapan *machine learning* untuk prediksi nilai *Low-Density Lipoprotein* (LDL), pemilihan dataset dan fitur prediktor memiliki peran yang sangat penting. Fitur prediktor adalah variabel *input* yang digunakan oleh model untuk mempelajari pola dan hubungan dengan variabel target, yaitu kadar LDL. Dalam penelitian ini, fitur yang digunakan mencakup tiga parameter utama dari profil lipid, yakni Total Kolesterol, *High-Density Lipoprotein* (HDL), dan Trigliserida. Ketiganya memiliki dasar ilmiah yang kuat dalam kaitannya dengan metabolisme lipid tubuh dan merupakan komponen penting dalam evaluasi risiko kardiovaskular (Wong et al., 2016).

Total Kolesterol menggambarkan jumlah keseluruhan kolesterol dalam darah dan menjadi indikator umum dalam *screening* penyakit jantung. HDL, yang dikenal sebagai kolesterol baik, membantu mengangkut kolesterol dari jaringan tubuh kembali ke hati untuk diproses dan dikeluarkan, sehingga kadarnya yang tinggi cenderung bersifat protektif terhadap risiko kardiovaskular. Sementara itu, Trigliserida adalah jenis lemak darah yang meningkat akibat konsumsi karbohidrat atau alkohol berlebih dan dapat meningkatkan risiko penyakit jantung koroner jika berada dalam kadar tinggi (Toth et al., 2019). Oleh karena itu, kombinasi ketiga parameter ini digunakan pula dalam rumus Friedewald untuk mengestimasi LDL secara manual.

Namun, dalam pengembangan model *machine learning*, pemrosesan data mentah menjadi sangat krusial. Dataset yang digunakan harus melalui tahap pembersihan (*data cleaning*), penghapusan duplikasi, pengisian nilai kosong (*missing values*), normalisasi nilai, serta transformasi data jika diperlukan. Langkah-langkah *preprocessing* ini dimaksudkan untuk mengurangi *noise* dalam data dan membantu model *machine learning* bekerja lebih optimal (Kuhn & Johnson, 2019).

Setelah proses pembersihan, data dibagi menjadi dua bagian utama, yaitu

data pelatihan (*training set*) dan data pengujian (*test set*). Pembagian ini bertujuan untuk mengevaluasi kemampuan generalisasi model terhadap data baru. Beberapa metode validasi juga diterapkan seperti *K-Fold Cross Validation* untuk memastikan hasil pelatihan tidak bias dan model tidak mengalami *overfitting* (James et al., 2021).

Pemilihan fitur yang tepat, ketersediaan data yang cukup, serta *preprocessing* yang optimal sangat menentukan performa akhir dari model prediksi LDL yang dibangun. Studi oleh Nguyen et al. (2021) menunjukkan bahwa kualitas data *input* memiliki kontribusi signifikan terhadap keberhasilan penerapan *machine learning* dalam konteks klinis. Hal ini juga diperkuat oleh temuan Alzubaidi et al. (2021), yang menekankan pentingnya representasi data yang baik untuk meningkatkan akurasi dalam aplikasi prediksi kesehatan berbasis AI.

3.4 Evaluasi Model

Evaluasi model merupakan komponen krusial dalam pengembangan *machine learning* karena menentukan seberapa andal dan akurat model dalam membuat prediksi. Dalam konteks prediksi nilai LDL, proses evaluasi dilakukan untuk mengetahui sejauh mana model mampu menghasilkan estimasi kadar LDL yang mendekati nilai aktual. Model yang baik harus mampu menunjukkan performa prediksi yang konsisten, akurat, dan dapat digeneralisasikan terhadap data baru dalam lingkungan laboratorium klinis (Chicco & Jurman, 2020).

Beberapa metrik evaluasi yang sering digunakan dalam regresi adalah *R-squared* (R^2), *Root Mean Square Error* (RMSE), dan *Mean Absolute Error* (MAE).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{Y}_I)^2}{\sum_{i=1}^{n}(y_i - \bar{Y}_I)^2} \tag{3.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.4}$$

- yi = nilai aktual (observasi ke-i)
- $\hat{y}_i$ = nilai prediksi dari model
- $\bar{y}_i$ = rata-rata dari semua nilai aktual
- n = jumlah sampel

R^2 merujuk pada metode rumus 3.2 mengukur proporsi variansi dalam data target yang dapat dijelaskan oleh model, dengan nilai mendekati 1 menandakan kecocokan model yang sangat baik (Witten et al., 2016). RMSE dan MAE merujuk pada metode rumus 3.3 dan 3.4 digunakan untuk menghitung rata-rata kesalahan prediksi, RMSE memberikan bobot lebih besar terhadap *error* besar, sehingga lebih sensitif terhadap outlier dibanding MAE (Brownlee, 2020).

Selain metrik tersebut, validasi silang seperti *K-Fold Cross Validation* digunakan untuk menilai kemampuan generalisasi model terhadap data yang tidak terlihat. Teknik ini membagi *dataset* menjadi beberapa subset (folds), kemudian melatih dan menguji model secara bergiliran agar seluruh data berkontribusi dalam pelatihan dan pengujian. Metode ini sangat direkomendasikan dalam studi prediksi klinis karena mampu mengurangi bias dan *overfitting* (Kohavi, 1995; Raschka & Mirjalili, 2019).

Beberapa studi sebelumnya mendukung efektivitas evaluasi ini. Misalnya, dalam penelitian oleh Huang et al. (2020), model Random Forest menunjukkan performa prediksi klinis yang kompetitif pada data kesehatan dengan nilai R^2 yang tinggi dan error rendah setelah dilakukan hyperparameter tuning dan validasi silang. Hal ini diperkuat oleh Somasundaram et al. (2021), yang menunjukkan bahwa Random Forest unggul dalam prediksi nilai parameter laboratorium dibandingkan model linier tradisional.

Selain metrik numerik, analisis residual juga dilakukan untuk menilai distribusi kesalahan prediksi. Residual yang tersebar secara acak di sekitar nol menunjukkan bahwa model telah menangkap pola yang ada di data dengan baik. Jika terdapat pola tertentu dalam residual, seperti tren naik atau turun, maka diperlukan perbaikan fitur atau penyesuaian model (Witten et al., 2016).

Evaluasi menyeluruh ini menjadi dasar pengambilan keputusan apakah model layak untuk diintegrasikan ke dalam sistem *backend* perusahaan. Dengan

pendekatan evaluasi yang sistematis dan terukur, model Random Forest yang dibangun diharapkan mampu mendukung proses otomatisasi analisis data laboratorium secara akurat dan efisien.

3.5 FastAPI

Dalam proses integrasi model *machine learning* ke dalam sistem *backend* yang digunakan oleh AdamLabs, salah satu teknologi penting yang digunakan adalah FastAPI. FastAPI merupakan framework modern berbasis Python yang digunakan untuk membangun layanan web API (*Application Programming Interface*) dengan performa tinggi. *Framework* ini dirancang untuk memudahkan pengembangan API yang cepat, efisien, serta didukung dengan dokumentasi otomatis dan validasi data berbasis tipe (*type hinting*) dari Python (Tiangolo, 2023).

FastAPI dibangun di atas dua komponen utama yaitu Starlette sebagai web *toolkit* dan Pydantic untuk validasi data. Keunggulan ini membuat FastAPI mampu menghasilkan layanan API dengan performa tinggi dan keamanan yang kuat, sangat sesuai dengan kebutuhan sistem prediksi laboratorium berbasis AI yang membutuhkan kecepatan dan keandalan (Tiangolo, 2023).

Dalam penerapannya pada proyek prediksi LDL ini, FastAPI digunakan sebagai jembatan antara model *machine learning* Random Forest dengan sistem *backend* perusahaan. Parameter masukan seperti Total Kolesterol, HDL, dan Trigliserida dikirim melalui *endpoint* HTTP POST dalam format JSON, kemudian diproses oleh model dan dikembalikan sebagai hasil prediksi LDL. Format komunikasi yang seragam dan dokumentasi otomatis dari FastAPI sangat mempermudah proses integrasi ini (Tiangolo, 2023).

Selain itu, FastAPI mendukung *asynchronous programming* yang memungkinkan server tetap responsif meskipun harus menangani banyak permintaan secara bersamaan. Fitur ini penting dalam konteks data laboratorium yang dikirim secara *real-time* dari berbagai alat *analyzer* ke sistem *middleware* (Sebastián Ramírez, 2020).

Dari sisi keamanan, FastAPI telah mendukung protokol OAuth2 dan JSON Web Token (JWT) untuk otentikasi dan otorisasi. Hal ini penting dalam menjaga

privasi data kesehatan pasien, terutama dalam sistem informasi laboratorium dan rumah sakit. Keunggulan-keunggulan tersebut menjadikan FastAPI sebagai pilihan tepat dalam pengembangan API modern berbasis Python di bidang kesehatan.

3.6 Uvicorn

Dalam pengembangan aplikasi berbasis FastAPI, salah satu komponen penting yang digunakan sebagai server *asynchronous* adalah Uvicorn. Uvicorn merupakan server ASGI (*Asynchronous Server Gateway Interface*) yang ringan dan cepat, dibangun di atas pustaka Python seperti uvloop dan httptools. Server ini dirancang untuk menjalankan aplikasi FastAPI dengan efisiensi tinggi dan mendukung pemrosesan *non-blocking*, yang sangat penting dalam lingkungan data intensif seperti sistem laboratorium digital (Uvicorn, 2023).

Sebagai server yang kompatibel dengan spesifikasi ASGI, Uvicorn menjadi pilihan utama untuk menjalankan aplikasi web modern berbasis Python, termasuk FastAPI. Salah satu keunggulan utamanya adalah dukungan terhadap *asynchronous* I/O, yang memungkinkan sistem tetap responsif meskipun harus menangani banyak permintaan secara bersamaan (Tiangolo, 2023). Dalam konteks proyek prediksi LDL di AdamLabs, Uvicorn berperan sebagai server yang menjalankan API FastAPI yang menerima *input* parameter laboratorium dan mengembalikan hasil prediksi LDL secara real-time.

Keunggulan lainnya dari Uvicorn adalah kemampuannya untuk mendukung fitur-fitur modern seperti WebSocket, HTTP/2, dan integrasi dengan *container* Docker. Hal ini memberikan fleksibilitas tinggi dalam proses *deployment* aplikasi ke lingkungan produksi, termasuk integrasi dengan *reverse* proxy seperti Nginx atau sistem orkestrasi seperti Kubernetes (Uvicorn, 2023).

Walaupun belum banyak jurnal ilmiah yang secara spesifik membahas Uvicorn, dokumentasi resmi dan *benchmark* komunitas menunjukkan bahwa Uvicorn merupakan salah satu ASGI server tercepat dan paling stabil dalam pengembangan API modern. Oleh karena itu, penggunaan Uvicorn dalam proyek ini dinilai tepat untuk mendukung sistem yang skalabel dan efisien.

3.7 Pydantic

Pydantic merupakan pustaka Python yang digunakan untuk validasi data dan parsing berbasis *type annotations*, dan menjadi komponen inti dalam pengembangan aplikasi FastAPI. Pydantic bekerja dengan cara memanfaatkan tipe data Python 3.6+ untuk memastikan bahwa data yang diterima dan dikirim melalui API sesuai dengan skema yang diharapkan. Keunggulan utama Pydantic terletak pada kemampuannya melakukan *parsing* data yang cepat serta mendeteksi kesalahan secara otomatis melalui pendekatan deklaratif (Samuel & McKinley, 2021).

Dalam konteks pengembangan sistem informasi laboratorium di AdamLabs, Pydantic digunakan untuk memastikan bahwa data yang dikirim ke model *machine learning* dalam bentuk JSON sudah valid. Sebagai contoh, parameter masukan seperti Total Kolesterol, HDL, dan Trigliserida divalidasi terlebih dahulu oleh Pydantic sebelum diteruskan ke model prediksi LDL. Jika terjadi ketidaksesuaian tipe data, nilai kosong, atau format yang salah, Pydantic akan secara otomatis mengembalikan pesan kesalahan yang dapat digunakan oleh sistem *backend* untuk menangani *error* tersebut secara elegan.

Pydantic juga memiliki performa tinggi karena dibangun di atas pustaka *dataclasses* dan menggunakan Cython untuk optimasi proses parsing. Hal ini membuatnya sangat cocok untuk sistem produksi yang menangani volume data besar dan memerlukan validasi data secara konsisten dan cepat (Samuel & McKinley, 2021).

Integrasi Pydantic dalam FastAPI tidak hanya memberikan keamanan data yang lebih baik, tetapi juga memungkinkan dokumentasi API yang otomatis dan akurat melalui standar OpenAPI. Setiap skema data yang didefinisikan dengan Pydantic akan langsung ditampilkan pada dokumentasi interaktif FastAPI, sehingga memudahkan kolaborasi antar pengembang dan pengguna sistem.

3.8 Numpy

NumPy (*Numerical* Python) merupakan pustaka fundamental dalam ekosistem Python yang dirancang untuk komputasi numerik tingkat tinggi dan efisien. Pustaka ini menyediakan objek array multidimensi yang disebut ndarray,

yang memungkinkan pemrosesan data dalam jumlah besar secara cepat dan optimal. Dalam proyek prediksi LDL berbasis *machine learning*, NumPy menjadi komponen penting untuk mendukung berbagai proses matematis yang terjadi dalam tahapan *preprocessing*, pelatihan, dan evaluasi model (Harris et al., 2020).

Salah satu kekuatan utama dari NumPy adalah kemampuannya dalam menangani operasi vektor dan matriks yang kompleks, seperti transformasi linier, dekomposisi matriks, dan kalkulasi statistik dasar. NumPy menyediakan banyak fungsi matematis seperti mean, median, standar deviasi, dot product, hingga *inverse matrix*, yang dibutuhkan untuk menyiapkan data sebelum dimasukkan ke dalam model pembelajaran mesin. Misalnya, ketika data hasil laboratorium diambil dari format JSON, NumPy digunakan untuk mengubah data tersebut menjadi *array* numerik agar dapat diproses secara efisien oleh model Random Forest.

NumPy juga mendukung konsep *broadcasting* yang memungkinkan operasi antara *array* dengan bentuk berbeda tanpa perlu melakukan iterasi eksplisit. Hal ini secara signifikan mempercepat pemrosesan data dan mengurangi beban komputasi, terutama pada sistem berskala besar seperti sistem informasi laboratorium yang menangani ratusan atau ribuan data uji klinis setiap harinya (Van Der Walt et al., 2011).

Selain mendukung *preprocessing*, NumPy sangat berguna dalam evaluasi performa model. Misalnya, untuk menghitung *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), atau membuat visualisasi distribusi residual, seluruhnya dapat dilakukan dengan operasi vektor yang efisien melalui NumPy. Dalam integrasinya dengan pustaka lain seperti Pandas (untuk manipulasi data), Scikit-learn (untuk algoritma *machine learning*), dan Matplotlib (untuk visualisasi), NumPy berfungsi sebagai tulang punggung pemrosesan numerik dalam keseluruhan *pipeline* analisis data.

Oleh karena itu, NumPy bukan hanya pustaka pendukung, tetapi menjadi elemen krusial dalam seluruh proses pengembangan sistem prediksi LDL berbasis *machine learning* karena kestabilan, kecepatan, dan fleksibilitasnya dalam menangani data numerik besar dan kompleks.

3.9 Scikit-learn

Scikit-learn adalah salah satu pustaka *open-source* paling berpengaruh dalam pengembangan *machine learning* di Python. Pustaka ini dikembangkan pertama kali sebagai bagian dari proyek *Google Summer of Code* dan kini menjadi bagian integral dalam berbagai proyek *data science* dan kecerdasan buatan. Scikit-learn menyediakan berbagai algoritma *machine learning* seperti klasifikasi, regresi, klastering, serta alat bantu untuk *preprocessing* data, seleksi fitur, evaluasi model, dan reduksi dimensi. Keunggulannya terletak pada sintaks yang konsisten, dokumentasi yang sangat lengkap, dan komunitas pengguna yang aktif (Pedregosa et al., 2011).

Dalam proyek prediksi nilai LDL di AdamLabs, Scikit-learn digunakan sebagai kerangka utama dalam pengembangan model Random Forest Regressor. Pustaka ini menyediakan implementasi Random Forest yang telah dioptimasi dengan berbagai parameter seperti `n_estimators`, `max_depth`, dan `random_state`, sehingga memungkinkan eksperimen model yang dapat dikontrol dan direproduksi. Selain model, Scikit-learn menyediakan berbagai fungsi *preprocessing* seperti `StandardScaler` untuk normalisasi data, serta `train_test_split` untuk membagi *dataset* menjadi data latih dan data uji dengan cara yang sistematis dan acak (Buitinck et al., 2013).

Salah satu fitur unggulan Scikit-learn adalah kemampuan untuk membuat *pipeline*, yang memungkinkan integrasi *preprocessing* dan model prediktif dalam satu alur kode yang kompak dan efisien. *Pipeline* ini membantu menjaga konsistensi saat proses pelatihan dan pengujian model, serta sangat memudahkan integrasi ke dalam sistem *backend* berbasis FastAPI. API dari Scikit-learn sangat terstruktur, dengan penggunaan metode `.fit()`, `.predict()`, dan `.score()` pada hampir semua objek estimator, yang memudahkan pengguna baru maupun pengembang lanjutan dalam menggunakan pustaka ini (Varoquaux et al., 2015).

Dalam konteks sistem laboratorium digital, Scikit-learn berperan besar dalam proses validasi model prediksi LDL melalui metrik regresi seperti *R-squared* (R^2), *Mean Absolute Error* (MAE), dan *Root Mean Squared Error* (RMSE). Metrik-metrik ini memungkinkan pengembang untuk menilai sejauh mana model berhasil memetakan data *input* ke *output* yang diharapkan. Selain itu, teknik validasi silang seperti *K-Fold Cross Validation* yang disediakan oleh Scikit-learn menjadi alat

penting dalam menilai generalisasi model terhadap data baru.

Scikit-learn juga terintegrasi secara erat dengan pustaka lain seperti NumPy dan Pandas, sehingga mendukung proses analisis dan manipulasi data yang kompleks. Dalam praktiknya, seluruh *pipeline* prediksi LDL di proyek ini—mulai dari pembacaan data laboratorium, *preprocessing*, pelatihan model, evaluasi performa, hingga pengujian akhir—dibangun dengan landasan Scikit-learn sebagai komponen utama.

3.10 Joblib

Joblib adalah pustaka Python yang sangat berguna dalam konteks pengembangan sistem *machine learning* karena kemampuannya untuk melakukan serialisasi objek Python secara efisien. Serialisasi ini merujuk pada proses menyimpan objek Python—seperti model yang telah dilatih—ke dalam file yang dapat dimuat kembali di lain waktu tanpa perlu melatih ulang. Salah satu keunggulan Joblib dibandingkan pustaka pickle adalah kemampuannya dalam menangani objek berbasis NumPy dengan lebih efisien, terutama pada *array* besar yang menjadi ciri khas data dalam *machine learning* (Varoquaux et al., 2011).

Dalam proyek prediksi LDL di AdamLabs, model Random Forest yang telah dilatih menggunakan pustaka Scikit-learn disimpan menggunakan Joblib dalam format .joblib. Proses ini memungkinkan penghematan waktu komputasi karena model tidak perlu dilatih ulang setiap kali API dipanggil. Saat server FastAPI dijalankan, model dimuat sekali dari file hasil serialisasi, dan dapat langsung digunakan untuk melakukan inferensi terhadap data laboratorium yang masuk. Pendekatan ini meningkatkan efisiensi sistem dan mengurangi beban komputasi harian secara signifikan (Joblib, 2023).

Selain untuk menyimpan model, Joblib juga memiliki fitur *caching* otomatis yang sangat bermanfaat dalam eksperimen *machine learning*. Fitur ini memungkinkan penyimpanan hasil dari fungsi-fungsi komputasi berat, sehingga ketika fungsi yang sama dipanggil kembali dengan parameter yang sama, hasil sebelumnya dapat langsung digunakan tanpa menghitung ulang. Hal ini mempercepat proses *tuning* model dan evaluasi parameter dalam eksperimen dengan *dataset* berukuran besar.

Joblib juga dirancang untuk mendukung paralelisme melalui pustaka *multiprocessing* Python. Dengan menggunakan *Parallel* dan *delayed*, pengembang dapat mengeksekusi proses pelatihan atau evaluasi model secara paralel pada beberapa *core* prosesor, yang sangat berguna dalam lingkungan pengembangan berskala besar seperti sistem informasi laboratorium digital yang digunakan oleh AdamLabs.

Secara keseluruhan, kemampuan Joblib dalam serialisasi cepat, penyimpanan efisien, *caching* otomatis, dan dukungan pemrosesan paralel menjadikannya alat penting dalam *pipeline machine learning* modern, terutama dalam proyek yang mengintegrasikan AI ke dalam sistem *backend* berbasis API.

3.11 Pandas

Pandas adalah pustaka Python yang dirancang khusus untuk manipulasi dan analisis data terstruktur, seperti data tabular dalam bentuk spreadsheet, database, maupun data hasil ekspor dari alat laboratorium. Pustaka ini dikembangkan oleh Wes McKinney dan menjadi salah satu pustaka yang paling esensial dalam ekosistem data *science* karena menyediakan struktur data utama seperti *DataFrame* dan Series. Struktur tersebut memungkinkan pengguna untuk mengelola, menyaring, membersihkan, dan menganalisis data dengan cara yang efisien dan deklaratif (McKinney, 2010).

Dalam proyek prediksi nilai LDL berbasis *machine learning*, Pandas berfungsi sebagai fondasi pada tahap awal *pipeline*, yaitu proses ekstraksi data laboratorium dari file eksternal seperti CSV atau Excel, transformasi data menjadi format terstruktur, hingga tahap loading ke *pipeline* analitik. Fungsi-fungsi seperti read_csv(), dropna(), fillna(), astype(), loc[], dan groupby() sangat membantu dalam membersihkan data, menghapus nilai yang hilang, memperbaiki tipe data, serta melakukan agregasi berdasarkan parameter tertentu. Proses ini sangat krusial dalam memastikan bahwa *dataset* yang masuk ke model prediksi benar-benar bersih dan siap digunakan.

Keunggulan utama Pandas terletak pada fleksibilitasnya dalam menangani data dari berbagai sumber. Ia mampu membaca dan menulis file dari beragam format, termasuk CSV, Excel, JSON, SQL, hingga Parquet. Kemampuannya dalam

interoperabilitas dengan pustaka seperti NumPy, Matplotlib, dan Scikit-learn menjadikan Pandas sangat ideal untuk digunakan dalam *workflow machine learning*. Dalam tahap eksplorasi data, Pandas juga menyediakan fungsi statistik dasar seperti mean(), std(), corr(), dan describe() yang sangat membantu dalam analisis awal fitur-fitur yang akan digunakan untuk pelatihan model (Reback et al., 2020).

Dalam konteks sistem informasi laboratorium digital seperti yang dikembangkan di AdamLabs, data dari berbagai alat analyzer seperti alat kimia klinik, hematologi, maupun imunoserologi biasanya datang dengan struktur dan format yang berbeda-beda. Pandas berperan penting dalam menstandarkan format data tersebut, baik melalui rekayasa ulang kolom, konversi tipe data, hingga penggabungan beberapa sumber data dalam satu dataset konsolidasi. Semua proses tersebut dilakukan dalam memori secara efisien dan berskala besar, menjadikan Pandas sebagai alat utama dalam mempercepat *pipeline* data dari raw file menuju proses prediksi nilai LDL dengan model Random Forest.

Dengan skalabilitas, kecepatan, dan fleksibilitas yang dimilikinya, Pandas terbukti menjadi solusi ideal untuk pengolahan data laboratorium yang heterogen dalam proyek prediksi LDL, serta memastikan integrasi data yang lancar menuju sistem *backend* berbasis FastAPI.

BAB IV
DESKRIPSI PEKERJAAN

4.1 Deskripsi Kerja Praktik

Selama menjalani masa kerja praktik di AdamLabs (PT. Wahana Meditek Indonesia), saya mendapat mandat utama untuk merancang, membangun, serta mengimplementasikan sistem berbasis *machine learning* yang difokuskan untuk melakukan prediksi terhadap nilai *Low-Density Lipoprotein* (LDL) secara otomatis berdasarkan *input* parameter hasil pemeriksaan laboratorium. Sistem ini merupakan bagian dari inovasi pengembangan sistem informasi laboratorium berbasis teknologi *Artificial Intelligence* (AI), yang bertujuan untuk meningkatkan efisiensi dalam pengolahan data, mempercepat proses diagnosis awal, serta meningkatkan akurasi perhitungan nilai LDL tanpa sepenuhnya mengandalkan metode manual konvensional.

Adapun proyek ini dikembangkan dengan fokus khusus pada sisi pengolahan dan pemodelan *machine learning*, bukan pada proses *parsing* data dari alat. Saya secara intensif melakukan serangkaian tahapan mulai dari analisis literatur ilmiah mengenai hubungan biologis antara LDL dengan parameter lain seperti total kolesterol, trigliserida, dan HDL, hingga tahap eksperimen implementasi model regresi prediktif menggunakan Random Forest. Selain itu, dilakukan pula analisis berbagai algoritma *machine learning* lain sebagai perbandingan, namun hasil evaluasi menunjukkan bahwa Random Forest memberikan performa prediktif yang paling optimal pada *dataset* yang tersedia.

Seluruh kegiatan ini saya jalankan dalam ekosistem pengembangan yang telah saya siapkan sendiri menggunakan Python dan pustaka-pustaka pendukung seperti Pandas untuk manipulasi data, NumPy untuk komputasi numerik, Scikit-learn untuk pemodelan *machine learning*, Joblib untuk penyimpanan model, serta FastAPI dan Uvicorn sebagai *backend interface* yang menghubungkan model ke sistem eksternal melalui API. Proyek ini mencakup tahapan mulai dari *preprocessing* data, eksplorasi fitur, pelatihan dan evaluasi model, validasi dengan metode tradisional, serta *deployment* API ke server lokal perusahaan. Hasil akhir dari kegiatan ini berupa sistem layanan prediksi LDL yang telah siap pakai dan

dapat diintegrasikan ke dalam sistem informasi laboratorium AdamLabs.

4.2 Uraian Pekerjaan

Pekerjaan dimulai dengan melakukan studi literatur yang mendalam untuk memahami landasan teoritis yang berkaitan dengan perhitungan nilai *Low-Density Lipoprotein* (LDL). Kajian dilakukan terhadap parameter laboratorium yang relevan seperti Total Kolesterol, Trigliserida, dan *High-Density Lipoprotein* (HDL), serta bagaimana ketiganya saling berinteraksi dan menjadi prediktor dalam formula medis konvensional seperti Friedewald. Literatur ilmiah dan publikasi akademik seperti Harris et al. (2020) untuk NumPy, Pedregosa et al. (2011) untuk Scikit-learn, serta dokumentasi FastAPI dan Pydantic (Ramírez, 2020; Sebastián, 2023) digunakan sebagai dasar dalam perencanaan sistem.

Langkah berikutnya adalah melakukan persiapan lingkungan kerja yang mencakup instalasi Python 3.x serta pustaka pendukung seperti Pandas, NumPy, Scikit-learn, Joblib, FastAPI, Uvicorn, dan Pydantic. Seluruh pustaka ini digunakan sesuai dengan peran fungsionalnya: Pandas untuk pemrosesan data tabular, NumPy untuk manipulasi numerik, Scikit-learn untuk proses pelatihan model dan evaluasi performa, FastAPI untuk membangun REST API, dan Pydantic untuk validasi *input* data.

Struktur proyek disusun secara modular dengan membagi folder menjadi beberapa komponen utama, yaitu: modul untuk pembacaan dan *preprocessing* data, modul untuk pelatihan dan penyimpanan model, serta modul khusus yang menangani permintaan API dan menampilkan hasil prediksi. File `train_ldl_randomforest.py` bertugas untuk memuat data, melakukan *training*, evaluasi, dan menyimpan model dalam format `.joblib`, sedangkan `ldl_predict.py` yang terdapat pada lampiran 5 digunakan untuk memuat model dan melakukan prediksi berbasis *input* JSON melalui *endpoint* API. Untuk melihat implementasi lengkap dari kedua file tersebut, pembaca dapat merujuk pada lampiran 4. Program `train_ldl_randomforest.py` yang disertakan di bagian akhir laporan ini.

Uji coba awal dilakukan secara lokal dengan menggunakan *dataset dummy* untuk memastikan bahwa *pipeline machine learning* dapat berjalan dengan benar

dari awal hingga akhir. Selanjutnya, lingkungan *backend* FastAPI dijalankan melalui Uvicorn agar dapat menangani permintaan HTTP secara *asynchronous* dan ringan. Dengan setup yang terstruktur ini, sistem prediksi LDL dapat dijalankan secara efisien dan siap untuk diintegrasikan lebih lanjut dalam *workflow* laboratorium digital di AdamLabs.

4.3 Preprocessing Data dan Penambahan Fitur

Data mentah yang digunakan pada proyek ini berupa file berformat CSV yang berisi hasil parameter laboratorium pasien, yaitu total kolesterol, trigliserida, HDL, dan nilai target LDL yang akan diprediksi. Proses awal yang dilakukan adalah melakukan pembacaan dan pembersihan data menggunakan pustaka Pandas. Salah satu bagian penting dalam *preprocessing* adalah mengonversi seluruh nilai ke format numerik untuk memastikan kompatibilitas dengan algoritma *machine learning* yang digunakan. Hal ini dilakukan dengan fungsi astype(float) untuk kolom *input* numerik, serta pengecekan apakah ada nilai yang tidak valid.

Langkah berikutnya adalah menangani *missing values*, yaitu data yang kosong atau tidak terisi, yang sering dijumpai dalam data klinis. *Missing values* diatasi dengan menghapus baris yang memiliki nilai kosong secara signifikan menggunakan dropna() untuk menjaga kualitas *dataset*. Dalam file train_ldl_randomforest.py, tahap ini ditandai dengan blok kode pembersihan data sebelum proses pelatihan dilakukan.

Selain itu, saya juga menambahkan beberapa fitur turunan berbasis rasio antara parameter. Misalnya, rasio kolesterol terhadap HDL (ratio_chol_hdl) dan rasio trigliserida terhadap HDL (ratio_trig_hdl). Penambahan fitur ini bertujuan untuk memberikan informasi tambahan kepada model terkait hubungan antar parameter dalam konteks prediksi LDL. Penelitian sebelumnya menunjukkan bahwa rasio-rasio tersebut memiliki korelasi yang cukup signifikan terhadap nilai LDL. Pembuatan fitur ini dilakukan menggunakan ekspresi aritmatika langsung pada Pandas *DataFrame*, seperti df['ratio_chol_hdl'] = df['total_cholesterol'] / df['hdl'], dan hasilnya kemudian disisipkan ke dalam *dataset* sebagai fitur baru.

Hasil akhir dari proses *preprocessing* ini adalah sebuah *DataFrame* bersih dan lengkap yang berisi seluruh parameter *input* serta nilai target LDL, yang siap

digunakan untuk proses pelatihan model. Untuk detail implementasi tahap *preprocessing* ini, pembaca dapat merujuk secara langsung pada kode program yang tercantum dalam lampiran train_ldl_randomforest.py guna memahami lebih lanjut tahapan dan struktur kode yang digunakan.

4.4 Hasil Testing Model

Setelah proses *preprocessing* dan penambahan fitur selesai, langkah selanjutnya dalam sistem prediksi LDL adalah melakukan pelatihan model *machine learning*. Model yang digunakan dalam proyek ini adalah Random Forest Regressor, salah satu algoritma *ensemble learning* dari pustaka Scikit-learn. Random Forest dipilih karena memiliki kemampuan menangani data non-linear, tahan terhadap *overfitting*, dan memberikan hasil prediksi yang cukup stabil dalam berbagai studi kasus kesehatan berbasis data tabular.

Proses pelatihan model dimulai dengan pembagian *dataset* menjadi data latih dan data uji menggunakan fungsi `train_test_split()` dari `sklearn.model_selection`. Dalam file `train_ldl_randomforest.py`, pembagian ini dilakukan dengan rasio 80:20, di mana 80% data digunakan untuk melatih model dan 20% sisanya digunakan untuk menguji performa prediksi.

Setelah data terbagi, dilakukan proses standardisasi fitur dengan menggunakan objek `StandardScaler` untuk memastikan semua fitur memiliki skala yang sebanding. Hal ini penting karena perbedaan skala antar fitur dapat mempengaruhi performa model. Berikut adalah cuplikan kode yang digunakan:

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42)
```

Sebelum proses pelatihan dimulai, fitur yang telah diturunkan dari proses *preprocessing*, seperti `chol_hdl_ratio` dan `tg_hdl_ratio`, juga telah dinormalisasi menggunakan teknik *scaling*. Scaling ini penting untuk menghindari dominasi fitur tertentu terhadap performa model, mengingat Random Forest tetap sensitif terhadap skala ketika digunakan dalam kombinasi dengan teknik evaluasi tertentu. Proses scaling dilakukan dengan fungsi `scale_features()` dari modul `utils.ldl_features`.

```
_, X_scaled = scale_features(X)
```

Setelah data siap, proses pelatihan dilanjutkan dengan menggunakan teknik *Grid Search* untuk menemukan kombinasi *hyperparameter* terbaik. *Grid Search* dikombinasikan dengan strategi *Repeated K-Fold Cross Validation* agar proses pelatihan dapat mengevaluasi performa model pada banyak subset data, sehingga mengurangi risiko *overfitting*. Dalam implementasinya, strategi validasi silang dilakukan menggunakan 4 fold yang diulang sebanyak 4 kali:

```
cv_strategy = RepeatedKFold(n_splits=4, n_repeats=4,
random_state=50)
```

Parameter yang dituning meliputi:

- `n_estimators`: jumlah pohon dalam hutan (misal: 200, 300, 400)
- `max_depth`: kedalaman maksimal pohon keputusan
- `min_samples_split`: jumlah minimal sampel untuk melakukan split
- `min_samples_leaf`: jumlah minimal sampel dalam daun pohon keputusan

Seluruh kombinasi parameter ini diuji menggunakan `GridSearchCV` untuk mencari konfigurasi dengan nilai skor R^2 terbaik.

```
grid = GridSearchCV(
    estimator=model,
    param_grid=PARAM_GRID,
    scoring='r2',
    cv=cv_strategy,
    n_jobs=-1,
    verbose=1
)
grid.fit(X_train, y_train)
```

Setelah proses *tuning* selesai, model terbaik diambil dari hasil pencarian dan digunakan untuk melakukan prediksi terhadap data uji (X_test). Prediksi ini kemudian digunakan untuk evaluasi yang lebih dalam pada bagian berikutnya.

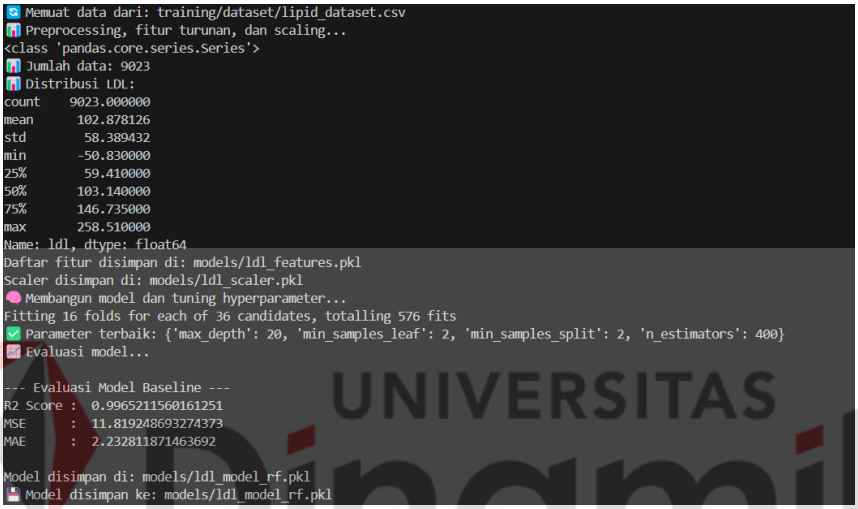
```
best_model = grid.best_estimator_
y_pred = best_model.predict(X_test)
```

Hasil parameter terbaik juga disimpan dalam file JSON (`best_params.json`) untuk keperluan dokumentasi dan analisis performa lebih

lanjut:

```
with open(os.path.join(MODEL_DIR, "best_params.json"), "w") as f:
    import json
    json.dump(best_params, f, indent=2)
```

Berikut dibawah ini merupakan hasil *training* model *machine learning* yang sudah di jalankan guna mendapatkan hasil R2, MSE, dan MAE pada model Random forest



Gambar 4.1 Hasil Training Model Random Forest

Dengan pendekatan pelatihan ini, model Random Forest tidak hanya dilatih dengan data terbaik yaitu memiliki R^2, tetapi juga divalidasi melalui metode statistik yang ketat untuk memastikan performa generalisasi yang baik. Tahapan pelatihan ini menjadi fondasi penting dalam membangun sistem prediksi LDL yang presisi dan dapat diandalkan dalam konteks laboratorium digital seperti AdamLabs.

4.4 Evaluasi Model dan Penyimpanan

Setelah model terbaik diperoleh dari hasil pelatihan dan tuning *hyperparameter*, evaluasi model dilakukan untuk mengukur sejauh mana model mampu melakukan prediksi nilai LDL dengan akurat. Evaluasi ini menggunakan tiga metrik utama, yaitu *R-squared* (R^2)**,** *Mean Squared Error* (MSE)**,** dan *Mean Absolute Error* (MAE)**.** R^2 digunakan untuk mengukur sejauh mana variasi target dapat dijelaskan oleh model. MSE menunjukkan seberapa besar rata-rata kesalahan

kuadrat yang dihasilkan model, sedangkan MAE memberikan informasi tentang rata-rata kesalahan absolut.

Alih-alih melakukan evaluasi langsung di dalam *script* utama, fungsi evaluasi telah disusun secara modular di file ldl_features.py, tepatnya dalam fungsi evaluate_model(). Berikut adalah cuplikan fungsi tersebut:

```
def evaluate_model(y_test, y_pred):
    print("\n--- Evaluasi Model Baseline ---")
    print("R2 Score : ", r2_score(y_test, y_pred))
    print("MSE      : ", mean_squared_error(y_test, y_pred))
    print("MAE      : ", mean_absolute_error(y_test, y_pred))
```

Fungsi diatas dipanggil setelah prediksi dilakukan oleh model:

```
y_pred = best_model.predict(X_test)
evaluate_model(y_test, y_pred)
```

Dengan pemisahan ini, proses evaluasi menjadi lebih rapi dan mudah digunakan kembali untuk model lain. Evaluasi dilakukan tidak hanya berdasarkan satu metrik, tetapi dari tiga perspektif (presisi, kesalahan kuadrat, dan kesalahan absolut), sehingga memberikan gambaran performa model secara lebih menyeluruh.

Setelah evaluasi selesai dan model dianggap valid, proses penyimpanan model (serialisasi) dilakukan menggunakan fungsi `save_model()` yang juga terdapat di `ldl_train_randomforest.py`. Berikut cuplikan fungsi tersebut:

```
os.makedirs(MODEL_DIR, exist_ok=True)
save_model(best_model, MODEL_PATH)
```

Fungsi-fungsi ini membantu dalam memastikan bahwa model yang telah disimpan dapat digunakan secara konsisten di tahap inferensi, karena fitur dan skala yang digunakan saat pelatihan tetap terjaga. Dengan struktur modular yang ditawarkan oleh `ldl_features.py`, proses evaluasi dan penyimpanan model menjadi tidak hanya efisien tetapi juga skalabel. Hal ini memperkuat praktik rekayasa perangkat lunak yang baik dalam pengembangan sistem AI laboratorium seperti yang diterapkan di AdamLabs.

4.5 Hasil Testing Model

Setelah layanan API berhasil dikembangkan dan di-*deploy* menggunakan *framework* **FastAPI** dan server **Uvicorn**, dilakukan pengujian untuk memastikan bahwa sistem dapat memberikan hasil prediksi nilai LDL dengan benar berdasarkan *input* parameter laboratorium. Pengujian ini sekaligus bertujuan untuk mengevaluasi performa respons sistem serta konsistensi hasil prediksi terhadap nilai-nilai yang dikalkulasi secara manual menggunakan rumus Friedewald sebagai acuan.

Pengujian dilakukan dengan mengirimkan permintaan HTTP `POST` ke endpoint `/ldl` yang telah dikembangkan. Permintaan ini memuat *payload* dalam format JSON, berisi nilai parameter laboratorium yaitu total kolesterol, trigliserida, dan HDL. Berikut adalah salah satu contoh permintaan *testing* beserta hasil prediksinya:

```
PS C:\Users\Hype> & "C:/Program Files/Python313/python.exe" "c:/magang adamlabs/test_client.py"
200
{'model': 'randomforest', 'input': [185.0, 98.0, 66.0], 'ldl_predicted': 95.58, 'ldl_friedewald': 99.4, '
accuracy_vs_friedewald': 0.9616, 'data_accepted': False, 'auto_submit_message': '❌ Prediksi tidak akurat
 (akurasi < 99%), tidak disimpan.'}
```

Gambar 4.2 Hasil prediksi model

Dari hasil nilai prediksi yang dimana jika nilai total kolesterol = 185, trigliserida = 98, dan HDL = 66, maka didapati nilai prediksi dari model Random Forest tersebut adalah 95,58. Dari hasil prediksi tersebut maka didapati kalau hasil prediksi tidak dapat dijadikan *dataset* baru karena tingkat akurasi hanya 0,9616 atau 96% sedangkan syarat agar bisa menjadi *dataset* kembali dengan tingkat akurasi dengan rumus friedewald adalah 0.99 atau 99%.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil kerja praktik yang dilaksanakan di PT. Wahana Meditek Indonesia (AdamLabs), dapat disimpulkan bahwa penerapan *machine learning* dengan algoritma Random Forest terbukti efektif dalam membangun sistem prediksi nilai *Low-Density Lipoprotein* (LDL) berdasarkan data parameter laboratorium seperti Total Kolesterol, HDL, dan Trigliserida. Seluruh tahapan mulai dari pengumpulan dan *preprocessing* data, eksplorasi fitur, pelatihan dan tuning model, evaluasi performa, hingga deployment API berhasil dilakukan secara sistematis dan sesuai tujuan awal proyek.

Oleh sebab itu didapati beberapa poin yang dapat diperhatikan dan dapat saya pelajari serta simpulkan antara lain:

1. **Pengembangan Model Prediksi LDL Berbasis Random Forest**
   Sistem prediksi nilai *Low-Density Lipoprotein* (LDL) berhasil dirancang dan diimplementasikan menggunakan algoritma Random Forest berdasarkan tiga parameter laboratorium utama: Total Kolesterol, HDL, dan Trigliserida. Meskipun *input* data terbatas, hasil evaluasi menunjukkan bahwa model mampu menangkap hubungan non-linear antar fitur dan memberikan akurasi prediksi yang sangat baik, dengan nilai koefisien determinasi (R^2) mencapai lebih dari 0.99. Ini membuktikan bahwa model dapat menjadi alternatif perhitungan LDL yang lebih adaptif dan cerdas dibanding metode manual seperti rumus Friedewald.
2. **Evaluasi Akurasi dan Performa Model Terhadap Pendekatan Manual**
   Model dievaluasi menggunakan metrik regresi seperti *R-squared* (R^2), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Error* (MAE). Nilai R^2 yang tinggi dan tingkat *error* yang rendah menunjukkan performa prediktif model yang sangat baik. Dibandingkan dengan perhitungan manual menggunakan rumus Friedewald, model *machine learning* menunjukkan ketahanan dan keandalan yang lebih baik dalam kondisi data yang tidak ideal, seperti kadar trigliserida yang tinggi atau data yang kurang lengkap.

Dengan demikian, pertanyaan mengenai tingkat akurasi dan efektivitas metode prediksi AI berhasil dijawab secara kuantitatif dan empiris.

3. **Integrasi ke Sistem *Backend* Perusahaan**

   Sistem prediksi LDL telah berhasil diintegrasikan ke dalam arsitektur *backend* perusahaan menggunakan FastAPI dan Uvicorn. Pengembangan layanan API *backend* ini memungkinkan sistem untuk menerima *input* data secara *real-time* melalui HTTP *request* dan memberikan hasil prediksi secara langsung. Meskipun belum mencakup sisi *frontend*, integrasi *backend* yang telah dilakukan sudah memenuhi aspek tujuan sistem dan batasan lingkup kerja praktik.

4. **Evaluasi dan Validasi Model Secara Menyeluruh**

   Evaluasi model dilakukan secara sistematis menggunakan teknik *Repeated K-Fold Cross Validation* untuk memastikan bahwa performa model dapat digeneralisasi dengan baik terhadap data baru. Selain itu, dilakukan pula validasi hasil prediksi terhadap nilai yang dihitung menggunakan rumus Friedewald. Keseluruhan tahapan ini memastikan bahwa sistem memenuhi kebutuhan laboratorium digital yang presisi dan efisien.

Dengan demikian, proyek ini telah berhasil menjawab rumusan dan batasan masalah yang telah ditetapkan, sekaligus memberikan kontribusi nyata dalam penerapan teknologi AI untuk sistem informasi laboratorium di dunia nyata.

5.2 Saran

Untuk pengembangan sistem lebih lanjut, berikut beberapa saran yang dapat dijadikan pertimbangan:

1. **Perluasan *Dataset* dan Fitur**

   Untuk meningkatkan akurasi dan generalisasi model, disarankan agar *dataset* diperluas dengan menambahkan jumlah sampel dan parameter lain yang relevan seperti usia, jenis kelamin, tekanan darah, atau riwayat penyakit. Hal ini memungkinkan model menangkap kompleksitas biologis yang lebih kaya dalam prediksi LDL.

2. **Penerapan Validasi Klinis**

Model yang telah dibangun sebaiknya diuji lebih lanjut melalui uji klinis atau dibandingkan dengan hasil aktual dari perangkat medis untuk memastikan validitas dalam lingkungan medis nyata.

3. **Pengembangan Antarmuka Pengguna (UI/UX)**

   Meskipun sistem *backend* telah berjalan optimal, pengembangan antarmuka *frontend* akan memudahkan pengguna non-teknis dalam menggunakan layanan prediksi LDL. Hal ini juga mendukung integrasi ke sistem informasi laboratorium yang telah berjalan di AdamLabs.

4. **Integrasi *Real-Time* ke *Middleware* Laboratorium**

   Diperlukan pengembangan fitur tambahan yang memungkinkan integrasi real-time dengan *middleware* alat laboratorium agar sistem ini dapat langsung menerima *input* dari perangkat dan memberikan output secara otomatis tanpa intervensi manual.

5. **Peningkatan Keamanan dan *Logging System***

   Untuk memastikan keamanan data laboratorium dan mencegah kesalahan, sistem sebaiknya dilengkapi dengan fitur *logging*, validasi tambahan, serta autentikasi berbasis token seperti JWT atau OAuth2, terutama jika sistem akan dikembangkan untuk produksi skala luas.

Dengan mempertimbangkan dan menerapkan saran-saran tersebut, sistem prediksi LDL berbasis AI ini diharapkan dapat dikembangkan lebih jauh dan memberikan manfaat optimal di dunia kesehatan digital.

DAFTAR PUSTAKA

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8*(1), 1–74. https://doi.org/10.1186/s40537-021-00444-8

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brownlee, J. (2020). *How to evaluate machine learning algorithms*. Machine Learning Mastery. https://machinelearningmastery.com

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *arXiv preprint*, arXiv:1309.0238.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics, 21*(1), 1–13. https://doi.org/10.1186/s12864-019-6413-7

Geron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature, 585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2020). Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics & Proteomics, 15*(1), 41–51. https://doi.org/10.21873/cgp.20063

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer. https://doi.org/10.1007/978-1-0716-1418-1

Joblib. (2023). *Joblib documentation*. https://joblib.readthedocs.io/en/latest/

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (pp. 1137–1143).

Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.

Martin, S. S., Blaha, M. J., Elshazly, M. B., Toth, P. P., Kwiterovich, P. O., Blumenthal, R. S., & Jones, S. R. (2013). Friedewald-estimated versus directly measured low-density lipoprotein cholesterol and treatment implications. *Journal of the American College of Cardiology, 62*(8), 732–739. https://doi.org/10.1016/j.jacc.2013.01.100

McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference* (pp. 51–56). https://doi.org/10.25080/Majora-92bf1922-00a

Nguyen, P. A., Tran, K. T., & Pham, H. T. (2021). Predicting laboratory test outcomes using machine learning algorithms: A case study on lipid profile data. *BMC Medical Informatics and Decision Making, 21*(1), 1–11. https://doi.org/10.1186/s12911-021-01442-2

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Ramírez, S. (2020). *FastAPI documentation*. https://fastapi.tiangolo.com

Raschka, S., & Mirjalili, V. (2019). *Python machine learning* (3rd ed.). Packt Publishing.

Reback, J., McKinney, W., jbrockmendel, Van den Bossche, J., Augspurger, T., Cloud, P., ... & Petersen, T. (2020). *pandas-dev/pandas: Pandas 1.0.3*. Zenodo. https://doi.org/10.5281/zenodo.3715232

Samuel, J., & McKinley, C. (2021). Building data validation systems in Python using Pydantic. *Journal of Open Source Software, 6*(67), 3822. https://doi.org/10.21105/joss.03822

Sebastián Ramírez. (2020). Async features in FastAPI. https://fastapi.tiangolo.com/async

Somasundaram, K., Thilagar, L., & Krishnamoorthy, S. (2021). Comparative study of machine learning models for medical data classification. *Journal of Healthcare Engineering, 2021*, 1–9. https://doi.org/10.1155/2021/3868023

Tiangolo. (2023). *FastAPI – FastAPI 0.100 documentation*. https://fastapi.tiangolo.com/

Toth, P. P., Graham, I., Lewanczuk, R. Z., & Bruckert, E. (2019). Triglyceride-rich lipoproteins as a causal factor for cardiovascular disease. *European Heart Journal Supplements, 21*(Suppl B), B31–B36. https://doi.org/10.1093/eurheartj/suz007

Uvicorn. (2023). *Uvicorn documentation*. https://www.uvicorn.org

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Varoquaux, G., Buitinck, L., Louppe, G., & Grisel, O. (2015). Scikit-learn. In *Encyclopedia of Machine Learning and Data Mining*. Springer.

Varoquaux, G., Blondel, M., Louppe, G., Pedregosa, F., & Mueller, A. (2011). Joblib: Running Python functions as pipeline jobs. *scikit-learn developers*. https://joblib.readthedocs.io

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data mining: Practical machine learning tools and techniques* (4th ed.). Morgan Kaufmann.

Wong, N. D., Young, D., & Zhao, Y. (2016). The predictive value of the lipid profile for the incidence of hypertension in a multiethnic population. *Current Hypertension Reports*, *18*(9), 76. https://doi.org/10.1007/s11906-016-0683-1