

## IMPLEMENTASI *BACKEND* DAN *MACHINE LEARNING* UNTUK ANALISIS KADAR LOGAM Fe DAN Cu BERBASIS CITRA DIGITAL PADA APLIKASI TUTOR AI TITRASI FARMASI

Akik Hidayat<sup>1</sup>, Azmi Naifah Iftinah<sup>2</sup>

Prodi Teknik Informatika, Universitas Padjadjaran Bandung

Email : akik@unpad.ac.id<sup>1</sup>, azmi23001@mail.unpad.ac.id<sup>2</sup>

### ABSTRAK

Digitalisasi praktikum kimia farmasi, khususnya dalam analisis kuantitatif kadar logam, merupakan langkah krusial untuk mengatasi keterbatasan instrumen laboratorium yang mahal dan meminimalisir subjektivitas pengamatan visual manusia pada metode konvensional. Analisis logam Besi (Fe) dan Tembaga (Cu) secara konvensional dilakukan menggunakan instrumen seperti *Atomic Absorption Spectrometry* (AAS) dan ICP-MS yang memiliki biaya operasional tinggi dan aksesibilitas terbatas untuk praktikum rutin mahasiswa. Penelitian ini memaparkan perancangan dan implementasi sistem backend yang terintegrasi dengan pemodelan *machine learning* untuk menganalisis konsentrasi logam Fe dan Cu berbasis *Digital Image Colorimetry* (DIC) pada aplikasi Tutor AI Titrasi Farmasi. Arsitektur sistem dibangun menggunakan pendekatan *two-layer backend*, di mana Node.js difungsikan sebagai API Gateway untuk mengelola alur data dan autentikasi, sementara FastAPI difungsikan sebagai layanan mikro (*microservice*) khusus untuk komputasi inferensi. Algoritma Random Forest Regressor diimplementasikan sebagai model prediksi utama dengan parameter  $n\_estimators$  sebesar 350. Berdasarkan karakteristik kimia visual, sistem mengekstraksi 9 fitur kombinasi dari ruang warna RGB untuk analisis sampel Fe, serta menggunakan fitur murni ruang warna HSV (H, S, V) untuk menangani fluktuasi pencahayaan pada sampel Cu. Evaluasi performa model menggunakan metrik *Mean Absolute Error* (MAE) dan koefisien determinasi ( $R^2$ ) menunjukkan bahwa model Fe mencapai MAE sebesar 1,5991 dengan  $R^2$  sebesar 0,9934, sedangkan model Cu menghasilkan MAE sebesar 7,7795 dengan  $R^2$  sebesar 0,7831. Seluruh fungsionalitas sistem telah tervalidasi melalui pengujian *Black Box* menggunakan Postman, mencakup autentikasi JWT, unggah citra, inferensi prediksi, dan penyimpanan riwayat pengujian ke basis data MySQL.

**Keywords**—Machine Learning,  
Pengolahan Citra Digital, Analisis

Logam, Backend, Random Forest,  
Tutor AI

## I. PENDAHULUAN

### A. Latar Belakang

Analisis kadar logam, khususnya Besi (Fe) dan Tembaga (Cu), merupakan salah satu kompetensi fundamental yang harus dikuasai oleh mahasiswa farmasi. Kedua logam tersebut bersifat esensial bagi sistem biologis manusia, namun dapat menjadi toksik apabila konsentrasinya melebihi ambang batas yang ditetapkan.

Secara konvensional, penentuan kadar logam dilakukan melalui metode spektroskopi serapan atom (AAS), ICP-MS, dan spektrofotometri UV-Vis. Meskipun menghasilkan pengukuran yang akurat, metode-metode ini memiliki keterbatasan mendasar: hanya dapat dioperasikan di laboratorium khusus, memerlukan biaya instrumen yang sangat tinggi, prosedur yang kompleks, serta penggunaan reagen kimia yang berpotensi mencemari lingkungan [1].

Seiring berkembangnya teknologi digital, *Digital Image Colorimetry* (DIC) berbasis kamera *smartphone* telah muncul sebagai pendekatan alternatif yang menjanjikan. DIC memanfaatkan kemampuan kamera untuk menangkap spektrum warna sampel secara digital dan mengonversinya menjadi data konsentrasi yang terukur [2].

## II. TINJAUAN PUSTAKA

### A. Kecerdasan Buatan dan *Machine Learning*

Kecerdasan Buatan (*Artificial Intelligence/AI*) merupakan cabang ilmu komputer yang berfokus pada pengembangan sistem yang mampu

mereplikasi fungsi kognitif manusia, seperti pengenalan pola dan pengambilan keputusan berbasis data [4]. Dalam konteks analisis data kuantitatif, *Machine Learning* (ML) memegang peranan penting karena kemampuannya mengidentifikasi pola tersembunyi dari himpunan data tanpa memerlukan instruksi pemrograman yang eksplisit [5].

Dalam penelitian ini, algoritma Random Forest Regressor diimplementasikan sebagai model prediksi utama. Algoritma ini merupakan teknik *ensemble learning* yang beroperasi dengan membangun sekumpulan pohon keputusan (*decision trees*) secara independen selama fase pelatihan. Output prediksi yang dihasilkan merupakan agregasi (rata-rata) dari seluruh pohon tersebut, yang secara signifikan mampu mereduksi variansi (*overfitting*) dan meningkatkan stabilitas prediksi pada data yang memiliki hubungan non-linear [6]. Pemilihan algoritma ini dinilai sangat relevan untuk menangani kompleksitas fitur warna citra kimia yang seringkali tidak memiliki korelasi linear sederhana terhadap nilai konsentrasi logam.

Implementasi algoritma dilakukan menggunakan pustaka Scikit-learn, yaitu *library open source* berbasis Python yang menyediakan berbagai implementasi algoritma *machine learning*, termasuk klasifikasi, regresi, dan pengelompokan [7].

### B. Pengolahan Citra Digital dan *Digital Image Colorimetry*

Pengolahan citra digital adalah serangkaian teknik komputasi untuk memanipulasi dan menganalisis representasi matriks dua dimensi dari sebuah gambar guna mengekstraksi

informasi kuantitatif [8]. Dalam bidang kimia analitik, pengolahan citra menjadi fondasi bagi metode *Digital Image Colorimetry* (DIC).

DIC memanfaatkan perangkat kamera, seperti yang terdapat pada *smartphone*, untuk menangkap spektrum warna sampel fisik dan mentransformasikannya menjadi nilai digital yang terukur secara matematis [2]. Metode ini mengeliminasi bias subjektivitas pengamatan mata manusia dan memberikan alternatif instrumen analitik yang presisi, portabel, dan terjangkau untuk kebutuhan praktikum pendidikan [9].

**C. Ekstraksi Fitur Ruang Warna: RGB dan HSV**

Ekstraksi fitur merupakan proses transformasi informasi visual mentah menjadi sekumpulan nilai representasi numerik yang menjadi input utama bagi model prediksi [10]. Penelitian ini mengimplementasikan dua pendekatan ruang warna yang berbeda berdasarkan karakteristik kimia masing-masing larutan logam.

Model Warna RGB (*Red, Green, Blue*) merepresentasikan warna sebagai kombinasi intensitas tiga warna primer. Meskipun representatif untuk mengukur tingkat kegelapan warna secara umum seperti pada sampel larutan Fe, RGB memiliki kelemahan mendasar berupa sensitivitas tinggi terhadap fluktuasi intensitas cahaya lingkungan [11]. Untuk analisis Fe, sistem mengekstraksi 9 fitur kombinasi yang mencakup nilai rata-rata (*mean R, mean G, mean B*), normalisasi warna (*r\_norm, g\_norm, b\_norm*), serta rasio antar komponen (*r\_g\_ratio, r\_b\_ratio,*

*g\_b\_ratio*). Model Warna HSV (*Hue, Saturation, Value*) memisahkan komponen rona warna dominan (*Hue*) dari saturasi (*Saturation*) dan tingkat kecerahan (*Value*). Pemisahan ini memungkinkan sistem analitik untuk fokus murni pada perubahan rona kimiawi larutan tanpa terdistorsi oleh kondisi pencahayaan laboratorium yang bervariasi [12].

Perbandingan karakteristik kedua ruang warna tersebut pada :

Aspek	RGB	HSV
Komponen	Red, Green, Blue	Hue, Saturation, Value
Sensitivitas Cahaya	Tinggi	Rendah
Jumlah Fitur	9 (mean, normalisasi, rasio)	H, S, V (HSV murni)
Penerapan	Analisis logam Fe	Analisis logam Cu
Keunggulan	Representatif untuk gradasi kegelapan	Stabil terhadap fluktuasi pencahayaan

**Tabel I.** Perbandingan Karakteristik Ruang Warna RGB dan HSV

**D. Arsitektur Backend dan Komunikasi RESTful API**

*Backend* berperan sebagai pusat kendali logika bisnis dan manajemen data dalam sebuah aplikasi. Node.js digunakan sebagai *runtime environment* JavaScript di sisi peladen karena karakteristik *non-blocking I/O* dan arsitektur *event-driven*-nya yang efisien dalam menangani banyak permintaan secara bersamaan [13]. Di atas Node.js, kerangka kerja Express.js digunakan untuk membangun arsitektur RESTful API utama yang mengatur alur *routing*, proses unggah citra, manajemen data pengguna, hingga penyediaan fungsi ekspor riwayat uji berformat CSV [14].

Untuk menangani beban komputasi analitik yang intensif dari model ML, digunakan kerangka kerja FastAPI berbasis Python. FastAPI dirancang khusus untuk membangun API dengan performa eksekusi tinggi dan validasi tipe data secara asinkron [15]. FastAPI diimplementasikan sebagai *microservice* terpisah yang bertugas menerima data dari Node.js, menjalankan ekstraksi fitur warna, dan mengembalikan hasil prediksi Random Forest.

Integrasi antara Node.js dan FastAPI difasilitasi melalui arsitektur Representational State Transfer (REST). REST API memanfaatkan protokol HTTP standar untuk memungkinkan pertukaran *payload* data secara ringan, independen, dan skalabel antar layanan [16][17]. Selain itu, mekanisme autentikasi dan otorisasi diimplementasikan menggunakan JSON Web Token (JWT) untuk memastikan keamanan akses dan perlindungan privasi data riwayat pengujian mahasiswa [18].

#### E. Pemodelan Sistem UML dan Manajemen Basis Data

Perancangan arsitektur perangkat lunak didokumentasikan menggunakan Unified Modeling Language (UML), yaitu bahasa pemodelan visual standar untuk menspesifikasikan dan mendokumentasikan artefak sistem perangkat lunak [19]. Pada perancangan sistem ini, digunakan beberapa jenis diagram UML, antara lain *Use Case Diagram* untuk memetakan interaksi pengguna dengan sistem, *Activity Diagram* untuk merepresentasikan alur kerja

prosedural, *Sequence Diagram* untuk menggambarkan urutan komunikasi antar komponen, *Deployment Diagram* untuk memvisualisasikan topologi fisik sistem, *Class Diagram* untuk merepresentasikan struktur statis *backend*, serta *Entity Relationship Diagram* (ERD) untuk memodelkan struktur basis data relasional.

Manajemen data riwayat pengujian dikelola menggunakan sistem basis data relasional MySQL. Penggunaan *Structured Query Language* (SQL) memastikan bahwa setiap transaksi data citra dan hasil prediksi tersimpan secara terstruktur dengan penegakan integritas referensial yang kuat [20]. Struktur basis data dirancang dengan empat tabel utama yang saling berelasi, sebagaimana dirangkum pada Tabel II berikut.

Nama Tabel	Fungsi Utama	Relasi
Users	Menyimpan data autentikasi dan peran pengguna	Primary key untuk samples dan chat_history
Samples	Menyimpan metadata citra, jenis logam, konsentrasi, dan path citra	Foreign key ke users
Materials	Menyimpan materi Pembelajaran yang diunggah dosen	Dikelola oleh users berperan admin
chat_history	Mendokumentasikan riwayat percakapan chatbot per sesi	Foreign key ke users

Tabel II. Struktur Basis Data Sistem Tutor AI

### III. METODOLOGI PENELITIAN

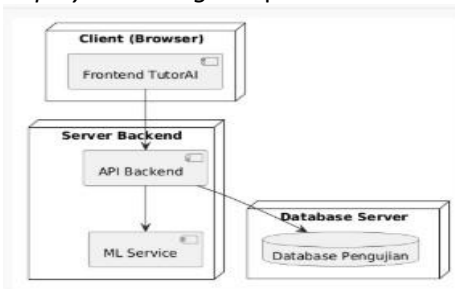
#### A. Arsitektur Sistem (*System Architecture*)

Untuk menjamin stabilitas dan skalabilitas sistem, diterapkan arsitektur *two-layer backend* yang memisahkan beban kerja I/O (*Input/Output*) dari beban kerja CPU (*Central Processing Unit*). Pemisahan ini bertujuan mencegah terjadinya *bottleneck* pada aplikasi ketika banyak pengguna melakukan simulasi praktikum secara bersamaan.

Lapis pertama dibangun menggunakan Node.js dengan kerangka kerja Express.js yang bertindak sebagai API Gateway. Lapisan ini difokuskan untuk menangani autentikasi pengguna melalui JSON Web Token (JWT), manajemen basis data, validasi *request*, serta pengelolaan unggah citra. Lapis kedua adalah layanan inferensi yang dikembangkan menggunakan FastAPI berbasis Python. FastAPI beroperasi secara independen untuk mengeksekusi ekstraksi fitur citra dan prediksi model ML. Komunikasi antar kedua layanan dilakukan melalui protokol internal HTTP POST yang membawa data citra dalam format *multipart/form-data*. Spesifikasi teknologi yang digunakan dalam pengembangan sistem ini dirangkum pada Tabel III berikut.

Komponen	Deskripsi Teknologi
Main Backend Runtime	Node.js (Express.js)
ML Service Runtime	Python (FastAPI)
Database Management	MySQL
Machine Learning Library	Scikit-learn, Pandas, Joblib
Algorithm	Random Forest Regressor

**Tabel III.** SPESIFIKASI TEKNOLOGI PENGEMBANGAN Rancangan topologi jaringan dan komunikasi antar layanan ini divisualisasikan dalam bentuk *Deployment Diagram* pada Gambar I.



**Gambar I.** Arsitektur fisik sistem (*Deployment Diagram*)

B. Pengumpulan Data dan Pra-pemrosesan Dataset primer yang digunakan

terdiri dari kumpulan citra digital larutan logam Besi (Fe) dan Tembaga (Cu) dengan variasi konsentrasi baku. Pengambilan citra dilakukan di dalam lingkungan laboratorium terkontrol dengan kondisi pencahayaan dan latar yang relatif seragam untuk menjaga konsistensi data. Penggunaan dataset hasil pengambilan langsung di laboratorium bertujuan untuk merepresentasikan kondisi praktikum nyata, sehingga model yang dihasilkan lebih relevan untuk diterapkan pada proses pembelajaran mahasiswa.

Sebelum dimasukkan ke dalam fase pelatihan, seluruh data citra melalui tahap pra-pemrosesan (*preprocessing*). Tahapan ini mencakup pemotongan dimensi citra (*cropping*) untuk memfokuskan area piksel murni pada larutan, serta eliminasi *noise* untuk mengurangi distorsi pantulan cahaya pada wadah uji. Dataset kemudian dipartisi menjadi data latih (*training set*) dan data uji (*testing set*) dengan rasio pembagian 75:25 menggunakan pustaka komputasi Scikit-learn.

### C. Ekstraksi Fitur Citra Digital

Proses ekstraksi fitur bertujuan untuk mentransformasi data visual citra menjadi representasi vektor numerik yang dapat dikalkulasi oleh algoritma regresi. Berdasarkan karakteristik kimia larutan, penelitian ini mengimplementasikan dua metode ekstraksi yang berbeda secara paralel:

- Analisis Logam Besi (Fe): Mengekstraksi 9 fitur kombinasi dari ruang warna RGB yang mencakup nilai rata-rata (*mean R*, *mean G*, *mean B*), normalisasi

warna ( $r\_norm$ ,  $g\_norm$ ,  $b\_norm$ ), serta rasio antar komponen warna ( $r\_g\_ratio$ ,  $r\_b\_ratio$ ,  $g\_b\_ratio$ ). Pendekatan ini dipilih karena indikator konsentrasi Fe berkorelasi kuat dengan tingkat kegelapan spektrum warna pada kanal RGB.

- Analisis Logam Tembaga (Cu): Mengekstraksi fitur murni dari ruang warna HSV yaitu komponen Hue (H), Saturation (S), dan Value (V). Transformasi warna ke dalam domain HSV diterapkan guna menjaga konsistensi dan stabilitas prediksi terhadap fluktuasi intensitas cahaya ambien saat pengambilan citra di lingkungan praktikum.

D. Pemodelan dan Parameterisasi *Machine Learning*

Model analitik dikembangkan menggunakan algoritma Random Forest Regressor. Proses parameterisasi dilakukan selama fase pelatihan, di mana model diinisialisasi dengan *hyperparameter*  $n\_estimators$  sebanyak 350 pohon keputusan. Pemilihan nilai 350 ditetapkan untuk mencapai titik ekuilibrium antara minimalisasi *error* prediksi dan efisiensi komputasi server. Model prediksi yang telah mencapai akurasi optimal kemudian diserialisasi ke dalam format berkas .pkl menggunakan pustaka Joblib, sehingga dapat dimuat secara instan ke dalam memori RAM oleh layanan FastAPI saat menerima permintaan analisis dari pengguna

E. Perancangan Basis Data  
Basis data dirancang

menggunakan sistem manajemen basis data relasional MySQL dengan empat tabel utama. Tabel *users* berfungsi sebagai pusat autentikasi yang menyimpan data akun dan peran pengguna. Tabel *samples* menyimpan metadata pengujian mencakup nama sampel, jenis logam (*metal\_type*: Fe atau Cu), nilai konsentrasi hasil prediksi (*concentration*), tanggal pengujian (*test\_date*), dan *path* direktori citra (*image\_path*), dengan relasi *foreign key* ke tabel *users*. Tabel *materials* digunakan untuk menyimpan materi pembelajaran yang diunggah oleh dosen, sementara tabel *chat\_history* mendokumentasikan riwayat percakapan *chatbot* per sesi pengguna.

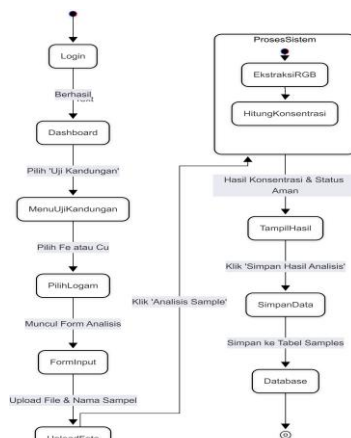
Field	Tipe Data	Keterangan
sample_id	INT (Primary Key)	Identitas unik untuk setiap sampel
sample_name	VARCHAR	Nama sampel yang diberikan pengguna
user_id	INT (Foreign Key)	Menghubungkan sampel dengan pengguna
test_date	DATE	Tanggal dilakukannya pengujian
metal_type	ENUM ('Fe', 'Cu')	Jenis logam yang dianalisis
concentration	DECIMAL	Nilai konsentrasi hasil prediksi (mg/L)
image_path	VARCHAR	Lokasi file citra sampel di server

Tabel IV. Struktur Tabel *Samples* pada Basis Data

F. Alur Kerja Integrasi Sistem

Alur kerja sistem (*workflow*) diawali dari pengiriman citra oleh pengguna melalui antarmuka Tutor AI dalam format *multipart/form-data*. *Endpoint* pada Node.js menerima paket data tersebut, melakukan validasi, dan meneruskannya ke layanan FastAPI melalui protokol internal HTTP POST. FastAPI merespons dengan mengembalikan nilai prediksi konsentrasi logam dalam satuan mg/L. Node.js kemudian menyimpan metadata pengujian beserta hasil prediksi tersebut ke dalam tabel *samples* pada basis data MySQL, sekaligus merelasikannya dengan tabel *users* untuk perekaman

riwayat historis pengujian mahasiswa. Perlu dicatat bahwa mekanisme penyimpanan berbasis aksi pengguna (*user-driven persistence*) diterapkan pada sistem ini. Artinya, hasil prediksi tidak langsung disimpan secara otomatis ke basis data, melainkan hanya akan disimpan apabila pengguna secara eksplisit menekan tombol "Simpan Hasil Analisis".



**Gambar II.** Activity Diagram

#### G. Pengujian Sistem

Pengujian sistem dilakukan menggunakan metode Black Box Testing, yaitu pendekatan pengujian yang mengevaluasi fungsionalitas sistem hanya berdasarkan spesifikasi input dan output tanpa meninjau struktur internal kode [22]. Pengujian dilakukan menggunakan perangkat lunak Postman untuk memvalidasi respons dari setiap *endpoint* API. Skenario pengujian mencakup empat fitur utama, yaitu validasi autentikasi JWT, keberhasilan unggah citra, ketepatan proses inferensi prediksi melalui FastAPI, serta ketepatan penyimpanan dan pengambilan data

riwayat pengujian dari basis data MySQL.

## IV. HASIL DAN PEMBAHASAN

### A. Implementasi dan Konfigurasi Backend API

Sistem *backend* berhasil diimplementasikan menggunakan arsitektur modular yang memisahkan logika pengendalian (I/O) dan logika analitik (CPU). Struktur folder *backend* disusun secara modular mencakup direktori utama seperti *routes*, *controllers*, *models*, *dataset*, dan *uploads*.

Lapisan API yang dibangun menggunakan Node.js (Express) dikonfigurasi untuk menangani permintaan (*request*) dari *frontend*, mencakup proses autentikasi menggunakan JSON Web Token (JWT), validasi tipe berkas citra, dan komunikasi basis data. Sementara itu, *microservice* analitik berbasis Python (FastAPI) berhasil dijalankan secara paralel sebagai layanan inferensi independen.

Ketika pengguna mengunggah citra larutan logam, Node.js menerima paket *multipart/form-data*, memverifikasi metadata, dan mengeksekusi panggilan asinkron (HTTP POST) ke layanan FastAPI. Setelah FastAPI menyelesaikan komputasi prediksi, hasilnya dikembalikan ke Node.js untuk direspons kembali ke klien. Pendekatan ini secara signifikan menjaga tingkat responsivitas (*latency*) aplikasi web Tutor AI tetap optimal

meskipun sedang melakukan kalkulasi *machine learning* di latar belakang.

**B. Implementasi Basis Data dan Mekanisme Penyimpanan**

Penyimpanan data operasional dan riwayat pengujian berhasil dikelola menggunakan basis data relasional MySQL. Implementasi *Entity Relationship* (ER) memastikan integritas data tetap terjaga. Sistem dirancang agar hasil prediksi konsentrasi (mg/L), identitas pengguna (dari tabel *users*), jenis logam (Fe/Cu), dan *path* direktori citra tersimpan secara konsisten di dalam tabel *samples*.

**C. Evaluasi Performa Model *Machine Learning***

Hasil eksperimen membuktikan bahwa model prediksi berhasil mempelajari pola karakteristik warna secara sangat baik, dengan rincian performa sebagai berikut:

1. Analisis Logam Besi (Fe): Model yang menggunakan 9 kombinasi fitur dari ruang warna RGB (*mean R, mean G, mean B, r\_norm, g\_norm, b\_norm, r\_g\_ratio, r\_b\_ratio, g\_b\_ratio*) menunjukkan performa yang sangat presisi. Hasil evaluasi mencatatkan nilai MAE sebesar 1,5991 dan  $R^2$  sebesar 0,9934. Nilai  $R^2$  yang mendekati 1,0 ini mengindikasikan bahwa fitur RGB sangat representatif dalam menjelaskan lebih dari 99% variasi perubahan tingkat kegelapan larutan Fe terhadap

nilai konsentrasinya. Tingginya akurasi ini disebabkan oleh karakteristik kimia larutan Fe yang menghasilkan gradasi warna yang kontras dan konsisten pada kanal RGB seiring perubahan konsentrasinya.

2. Analisis Logam Tembaga (Cu): Model Cu diekstraksi menggunakan fitur HSV murni (*h, s, v*). Evaluasi model Cu menghasilkan nilai MAE sebesar 7,7795 dan  $R^2$  sebesar 0,7831. Meskipun nilai  $R^2$  lebih rendah dibandingkan model Fe, penggunaan fitur dominan HSV terbukti memberikan tingkat stabilitas yang memadai dalam membedakan rona warna (*Hue*) sampel Cu di bawah kondisi fluktuasi pencahayaan laboratorium. Perbedaan performa antara model Fe dan Cu ini secara langsung mencerminkan perbedaan karakteristik visual kimiawi kedua larutan tersebut, di mana larutan Fe memiliki gradasi warna yang lebih kontras dan mudah dideteksi dibandingkan larutan Cu.

Rincian hasil evaluasi kinerja model secara kuantitatif disajikan pada Tabel V

Jenis Logam	Ruang Warna	MAE	$R^2$ Score
Besi (Fe)	RGB	1,5991	0,9934
Tembaga (Cu)	HSV	7,7795	0,7831

**Tabel V. HASIL EVALUASI MODEL PREDIKSI LOGAM**  
Data performa tersebut diperoleh dari hasil eksekusi skrip

pengujian pada lingkungan pengembangan, sebagaimana ditunjukkan pada Gambar III

```
PS D:\Kodingan\smt-5\mapang-PPeP\Tutor-AI-Farmasi\backend> python .\evaluate_model.py
Evaluasi model FE (RGB)
MAE : 1.5991
R² : 0.9934

Evaluasi model CU (FITUR MENYESUAIKAN MODEL TRAINING)
Fitur evaluasi CU: ['h', 's', 'v', 'mean_p', 'mean_b', 'mean_b']
MAE : 7.7795
R² : 0.7831
```

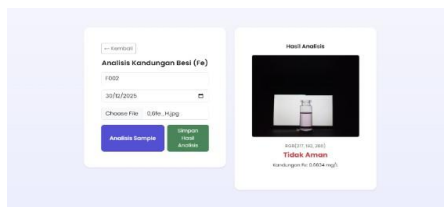
Gambar III. Hasil Evaluasi dan Metrik Performa Model pada Terminal

D. Pengujian Fungsionalitas (*Black Box Testing*)

Fitur Utama	Skenario Pegujian	Hasil	Status
Autentikasi	Login dan validasi token JWT	Berhasil	Valid
Unggah Sampel	Pengiriman citra via form-data	Tersimpan	Valid
Analisis AI	Prediksi kadar melalui FastAPI	Nilai keluar	Valid
Riwayat	Pengambilan data dari database	Data muncul	Valid

Tabel VI. HASIL PENGUJIAN FUNGSIONAL SISTEM

E. Antarmuka Hasil Pengujian



Gambar IV. Tampilan Halaman Uji Kandungan Besi

V. KESIMPULAN

Pertama, arsitektur *two-layer backend* yang memisahkan layanan Node.js sebagai API Gateway dan FastAPI sebagai *microservice* inferensi terbukti sangat efektif dalam menjaga stabilitas dan responsivitas sistem.

Kedua, pendekatan ekstraksi fitur warna yang disesuaikan dengan karakteristik visual masing-masing larutan logam terbukti secara signifikan meningkatkan ketahanan prediksi model.

Ketiga, mekanisme *user-driven persistence* yang diterapkan pada sistem terbukti efektif dalam menjaga integritas dan kualitas data pada basis

data MySQL..

DAFTAR PUSTAKA

- [1] M. Arjomandi and H. Shirkhanloo, "A Review: Analytical methods for heavy metals determination in environment and human samples," *Analytical Methods in Environmental Chemistry Journal*, vol. 2, no. 3, pp. 97–126, 2019.
- [2] L. Ciaccheri, B. Adinolfi, A. A. Mencaglia, and A. G. Mignani, "Smartphone-enabled colorimetry," *Sensors*, vol. 23, no. 12, p. 5559, 2023.
- [3]
- [4] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2021.
- [5] *Journal of Machine Learning Research*, "Journal of Machine Learning Research," 2025. [Online]. Available: <https://www.jmlr.org/>
- [6] V. M. Herrera, T. M. Khoshgoftaar, F. Villanustre, and B. Furht, "Random forest implementation and optimization for Big Data analytics on LexisNexis’s high performance computing cluster platform," *Journal of Big Data*, vol. 6, art. 68, 2019.
- [7] scikit-learn developers, "scikit-learn: Machine learning in Python," 2025. [Online]. Available: <https://scikit-learn.org/stable/>
- [8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2020.

- [9] V. Markus, A. A. Paul, R. S. Marks, and J. Caleb, "Smartphone Digital Image Colorimetry: An Affordable Easy-To-Use Technique," *Analytical Letters*, 2025.
- [10] M. S. Khrisat, H. G. Zaini, and Z. A. Alqadi, "Simple, flexible method to extract digital image features," *IETA*, 2021.
- [11] GeeksforGeeks, "Difference between RGB, CMYK, HSV, and YIQ color models," GeeksforGeeks, 2025. [Online]. Available: <https://www.geeksforgeeks.org/difference-between-rgb-cmyk-hsv-and-yiq-color-models/>
- [12] M. A. Amrozi, D. S. W. Figo, and R. Wahyusari, "Perbandingan segmentasi ruang warna HSV dan YCbCr untuk deteksi objek," *Infomatek*, vol. 26, no. 2, 2024.
- [13] OpenJS Foundation, "Node.js documentation," 2025. [Online]. Available: <https://nodejs.org/>
- [14] Express.js contributors, "Express – Node.js web application framework," 2025. [Online]. Available: <https://expressjs.com/>
- [15] FastAPI Developers, "FastAPI: High-performance Python web framework for building APIs," 2025. [Online]. Available: <https://fastapi.tiangolo.com/>
- [16] D. Pepadun, "Representational State Transfer (REST) sebagai arsitektur layanan web berbasis HTTP," *Jurnal Pepadun*, vol. 2, no. 1, pp. 81–89, 2021.
- [17] A. M. N. Riady, P. Paniran, and I. B. Suksmadana, "Perancangan backend API berbasis REST-API pada aplikasi rekomendasi resep makanan," *Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer*, vol. 2, no. 3, pp. 94–106, 2024.
- [18] R. Allafi and A. A. D. Darem, "Usability and security in online authentication systems," *International Journal of Advanced and Applied Sciences*, vol. 12, no. 6, pp. 1–12, 2025.
- [19] M. R. Wayahdi and F. Ruziq, "Pemodelan sistem penerimaan anggota baru dengan Unified Modeling Language (UML)," *Jurnal Minfo Polgan*, vol. 12, no. 1, pp. 1514–1521, 2023.
- [20] D. T. W. Ying et al., "Comparative analysis of SQL and NoSQL databases: Data models, use cases, and performance insights," *Information Research Communications*, vol. 2, no. 1, pp. 47–70, 2025.
- [21] D. Murdiani and M. Sobirin, "Perbandingan metodologi Waterfall dan RAD dalam pengembangan sistem informasi," *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, 2023.
- [22] N. Mahrozi and M. A. Yaqin, "Pengujian Aplikasi dengan Metode Blackbox Testing: Analisis Boundary Value dan Equivalence Partitioning," *Scientica: Jurnal Ilmiah Sains dan Teknologi*, 2024