Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

KLASIFIKASI CITRA *BREAST CANCER* BERBASIS ARSITEKTUR *MICROSERVICES*

Habibullah Akbar[1], Matius Eliezer Sinaga[2*]

Program Studi Magister Ilmu Komputer[1],
Program Studi Teknik Informatika[2]
Fakultas Ilmu Komputer[1], Fakultas Ilmu Komputer[2]
Universitas Esa Unggul[1], Universitas Esa Unggul[2]

[*]Corresponding author :
methewsinaga65@student.esaunggul.ac.id

Authors Email: habibullah.akbar@esaunggul.ac.id [1],
methewsinaga65@student.esaunggul.ac.id [2]

Received: December 18,2025**. Revised:** February 7,2026**. Accepted:** February 10,2026**. Issue Period:** Vol.10 No.1 (2026), Pp. 319-334

Abstrak: Perkembangan pesat Kecerdasan Buatan (AI) dan Pembelajaran Mesin (ML) telah mendorong inovasi di bidang medis, khususnya dalam sistem diagnosis berbasis citra seperti deteksi kanker payudara. Penelitian ini bertujuan untuk merancang dan mengimplementasikan arsitektur berbasis microservices untuk klasifikasi citra kanker payudara guna mencapai skalabilitas, modularitas, dan fleksibilitas yang lebih baik. Sistem yang diusulkan membagi fungsi menjadi tiga layanan pra-pemrosesan, klasifikasi, dan penyimpanan yang terhubung melalui API RESTful. Layanan model menggunakan FastAPI (Python) yang terintegrasi dengan model pembelajaran mendalam (breast_cancer_model.h5), sedangkan Java Spring Boot berfungsi sebagai backend dan Angular.js sebagai frontend. Pengujian eksperimental menggunakan Postman dan JMeter menunjukkan waktu respons 8,39 ms untuk GET /, 654 ms untuk POST /predict, dan 971,99 ms untuk GET /reload-model. Hasil ini menunjukkan bahwa sistem berbasis microservices memberikan kinerja yang efisien, modularitas tinggi, dan pemeliharaan yang lebih baik untuk aplikasi klasifikasi citra medis bertenaga AI.

Kata kunci: Kecerdasan Buatan; Rest API; Kanker; Microservice;

Abstract: *The rapid development of Artificial Intelligence (AI) and Machine Learning (ML) has driven innovation in the medical field, particularly in image-based diagnosis systems such as breast cancer detection. This research aims to design and implement a microservices-based architecture for breast cancer image classification to achieve better scalability, modularity, and flexibility. The proposed system divides functions into three services preprocessing, classification, and storage connected via RESTful APIs. The model service uses FastAPI (Python) integrated with a deep learning model (breast_cancer_model.h5), while Java*

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

Spring Boot serves as the backend and Angular.js as the frontend. Experimental testing using Postman and JMeter shows response times of 8.39 ms for GET /, 654 ms for POST /predict, and 971.99 ms for GET /reload-model. These results indicate the microservices-based system provides efficient performance, high modularity, and better maintainability for AI-powered medical image classification applications.

Keywords: *Artificial Intelligence; API Break; Cancer; microservices;*

I. PENDAHULUAN

Kanker payudara merupakan penyakit dengan prevalensi tinggi di dunia. Menurut World Health Organization (WHO), pada tahun 2020 terdapat 2,3 juta kasus baru kanker payudara, dengan angka kematian mencapai 685.000 jiwa. Deteksi dini melalui analisis gambar medis seperti mammography dapat meningkatkan tingkat kelangsungan hidup hingga 90% jika terdeteksi pada stadium awal [1]. Namun, proses diagnosis manual oleh radiolog memakan waktu, rentan kesalahan, dan terbatas oleh jumlah tenaga ahli.

Perkembangan artificial intelligence (AI) dan machine learning (ML) membuka peluang pengembangan sistem klasifikasi gambar medis yang akurat, cepat, dan skalabel. Model deep learning seperti Convolutional Neural Network (CNN) telah terbukti mampu mendeteksi kelainan pada gambar mammography dengan akurasi >85%. Tantangan teknis muncul dalam pengelolaan dataset besar, pemrosesan real-time, integrasi dengan sistem rumah sakit, dan skalabilitas infrastruktur [2].

Namun, pengembangan sistem deteksi otomatis berbasis AI menghadapi tantangan besar. Masalah utama muncul dari sisi pengolahan data gambar berukuran besar, kompleksitas model pembelajaran mendalam (*deep learning*), serta pengelolaan sistem yang efisien dan dapat diskalakan[3]. Arsitektur perangkat lunak yang digunakan menjadi faktor kunci dalam menentukan performa sistem AI.

Sebagian besar aplikasi AI masih menggunakan arsitektur monolitik, di mana seluruh komponen (*frontend*, *backend*, dan model AI) digabung dalam satu kesatuan system [4]. Pendekatan ini cenderung sulit dikelola, terutama saat volume data dan pengguna meningkat. Sebagai solusi, muncul pendekatan arsitektur microservices, yang memecah aplikasi menjadi layanan-layanan kecil yang berjalan secara independen, berkomunikasi melalui REST API.

Arsitektur microservices menawarkan solusi dengan memecah aplikasi menjadi layanan kecil, independen, dan dapat dikembangkan secara terpisah[5]. Dalam konteks klasifikasi gambar kanker payudara, microservices memungkinkan pemisahan fungsi: pra-pemrosesan gambar, inferensi model AI, penyimpanan hasil analisis, dan antarmuka pengguna. Setiap layanan berkomunikasi melalui REST API, di-containerize menggunakan Docker, dan diorkestrasi dengan Kubernetes.

Arsitektur microservices memecah aplikasi menjadi layanan kecil dan independen, memungkinkan pemisahan fungsi seperti pra-pemrosesan gambar, pelatihan model AI, inferensi, dan penyimpanan data pasien. Setiap layanan dapat dioptimalkan secara terpisah melalui komunikasi REST API. Penelitian sebelumnya, seperti “Penerapan Microservices untuk Sistem Diagnosis Medis Berbasis AI” [6], menunjukkan peningkatan efisiensi pemrosesan data medis, tetapi belum mengeksplorasi implementasi spesifik untuk klasifikasi gambar kanker payudara. Penelitian lain, “Analisis Performa Microservices pada Pengolahan Gambar Medis” [7], terbatas dalam integrasi model ML dan manajemen data pasien.

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

Penelitian ini mengisi gap dengan merancang dan mengimplementasikan sistem klasifikasi gambar kanker payudara berbasis microservices, fokus pada keunggulan, tantangan, dan implementasi praktis. Alasan penelitian diperlukan karena kurangnya studi unik yang mengintegrasikan microservices dengan klasifikasi gambar medis secara mendalam.

II. METODE DAN MATERI

2.1. Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimental dan komparatif, dengan fokus pada implementasi dan pengujian sistem berbasis microservices[8]. Sistem dibangun dengan membagi fungsi utama ke dalam tiga service: preprocessing service, classification service, dan storage service, masing-masing dikembangkan dengan teknologi yang berbeda dan terhubung melalui RESTful API [9].

2.2. Arsitektur Sistem

Secara umum, sistem memiliki tiga lapisan utama:

1. Frontend (Angular.js)

Bertugas menampilkan antarmuka pengguna, menerima input gambar, dan menampilkan hasil prediksi.

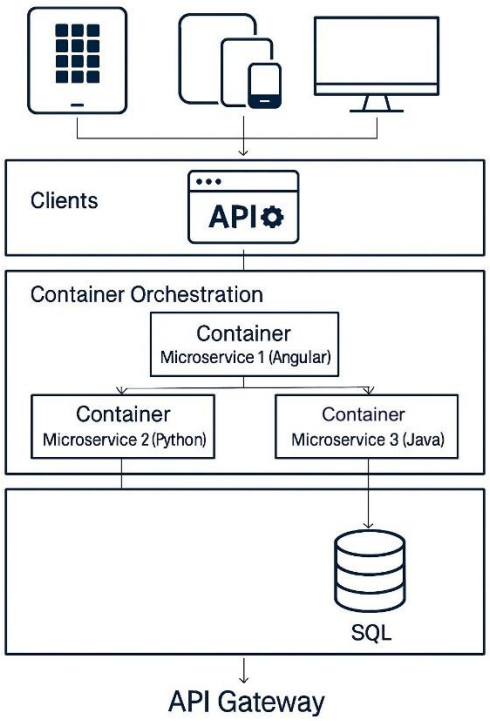
2. Backend (Java Spring Boot)

Berfungsi sebagai penghubung (API Gateway) antara frontend dan layanan AI, serta mengelola data hasil klasifikasi yang disimpan di database.

3. Model Service (Python FastAPI)

Menjalankan model deep learning breast_cancer_model.h5 untuk mengklasifikasikan gambar menjadi kategori “Cancer” atau “Non-Cancer”.

Komunikasi antar-layanan dilakukan melalui protokol HTTP dengan format data JSON.



Gambar 1. Arsitektur Microservices Kanker Payudara

DOI: 10.52362/jisamar.v10i1.2298

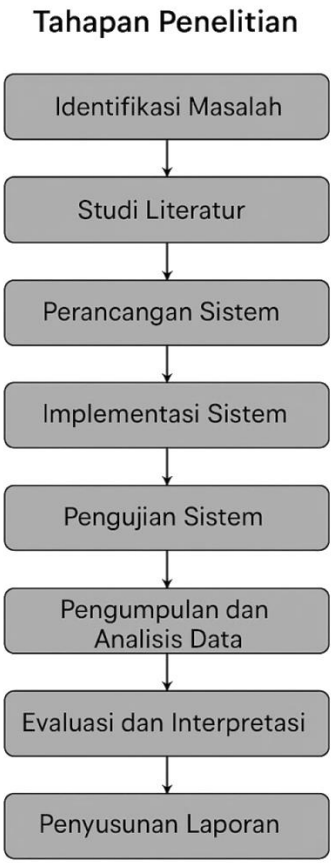
Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

2.3. Hardware dan Software Sistem

1. **Perangkat keras:**
   DELL Latitude 7300 (Intel Core i5 1.4 GHz, RAM 16 GB, SSD 500 GB)

2. **Perangkat lunak:**
   1. OS: macOS & Windows 10
   2. Backend Preproses: Java Spring Boot
   3. AI Service: FastAPI (Python)
   4. Database: MySQL
   5. Frontend: Angular.js
   6. Tools: Postman, Swagger
   7. Cloud server : VPS Hostinger

2.4. Tahapan Penelitiian



Gambar 2. Tahapan Penelitian

Berikut adalah tahapan-tahapan dalam penelitian [10]:

1. Identifikasi Masalah

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

Menentukan kebutuhan sistem klasifikasi gambar kanker payudara.

2. Studi Literatur
   Mengkaji jurnal dan dokumentasi terkait microservices dan klasifikasi gambar.
3. Perancangan Sistem
   Tiga layanan terpisah (Preprocess, Classification, Storage).
4. Implementasi Sistem
   Mengembangkan sistem menggunakan Python, Flask, Angular, Spring Boot TensorFlow, Docker, dan Kubernetes.
5. Pengujian Sistem
   Load Testing dengan JMeter untuk mengukur waktu respons, throughput, dan penggunaan sumber daya. Pengujian skalabilitas pada microservices dengan menambah instance layanan.
6. Pengumpulan dan Analisis Data
   Membandingkan performa kedua sistem berdasarkan metrik pengujian.
7. Evaluasi dan Interpretasi
   Menganalisis keunggulan dan tantangan masing-masing arsitektur.
8. Penyusunan Laporan
   Mendokumentasikan proses dan hasil penelitian.

III. PEMBAHASAN DAN HASIL

3.1. Implementasi dan Sistem

Implementasi dilakukan dengan membangun tiga komponen utama:

1. **FastAPI Service (Python):**
   Menyediakan endpoint /predict, /reload-model, dan / untuk klasifikasi gambar serta pemuatan ulang model deep learning tanpa perlu menghentikan server.

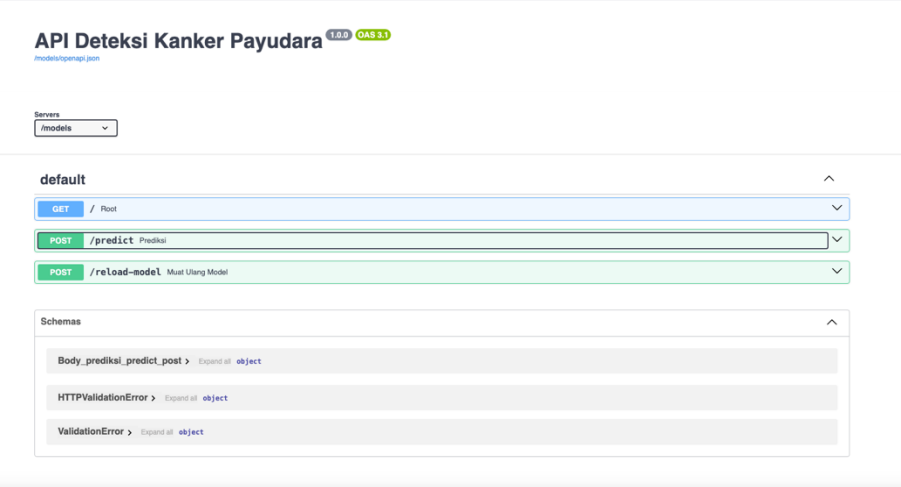
```python
from fastapi import FastAPI, UploadFile, File, HTTPException
import numpy as np
from PIL import Image
import tensorflow as tf
from tensorflow import keras
import os

app = FastAPI(
    title="API Deteksi Kanker Payudara",
    version="1.0.0",
    root_path="/models"
)

DAFTAR_KELAS = ["Cancer", "Non-Cancer"]
LOKASI_MODEL = "breast_cancer_model.h5"
model = None

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'


@app.on_event("startup")
async def muat_model_saat_startup():
    """Memuat model saat aplikasi dijalankan"""
    global model
    if not os.path.exists(LOKASI_MODEL):
        raise RuntimeError(f"Berkas model tidak ditemukan: {LOKASI_MODEL}")
    model = keras.models.load_model(LOKASI_MODEL, compile=False)
    print("Model berhasil dimuat")
```

Gambar 3. Kode Implementasi FAST API Python

DOI: 10.52362/jisamar.v10i1.2298

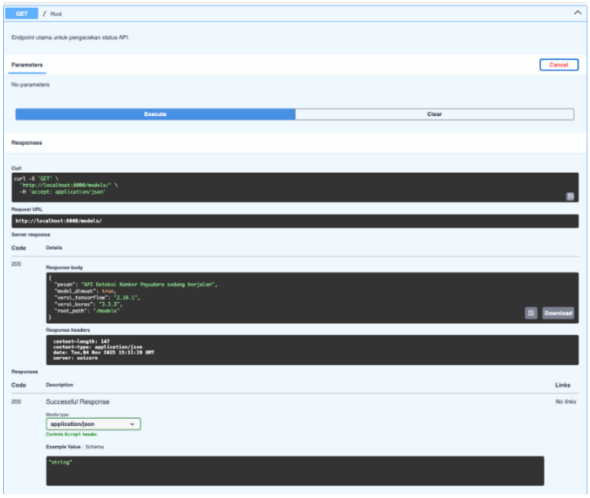
Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

API ini digunakan untuk Deteksi Kanker Payudara yang dibuat menggunakan library FAST API dari python, terdapat beberapa library yg dibuat untuk upload file, untuk manipulasi dam konversi gambar ke model dan untuk menjalankan model deep learning (.h5)



Gambar 1. EndPoint API Deteksi Kanker Payudara

Gambar 4 adalah EndPoint API Deteksi Kanker Payudara yang dimana terdapat beberapa API dalam beberapa method seperti: GET / Root, POST/predict, POST/reload model.

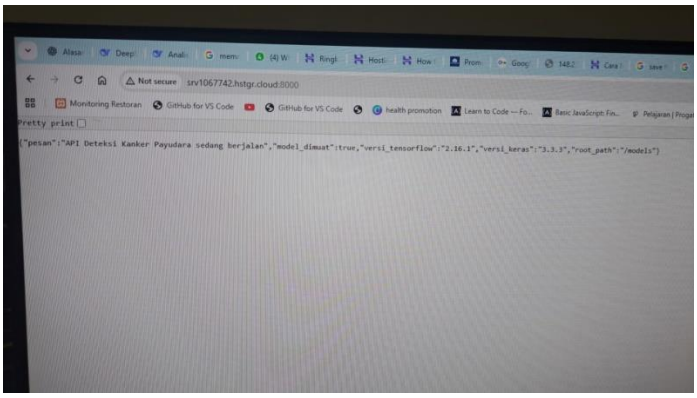


Gambar 2. endpoint utama untuk pengecekan API (GET)

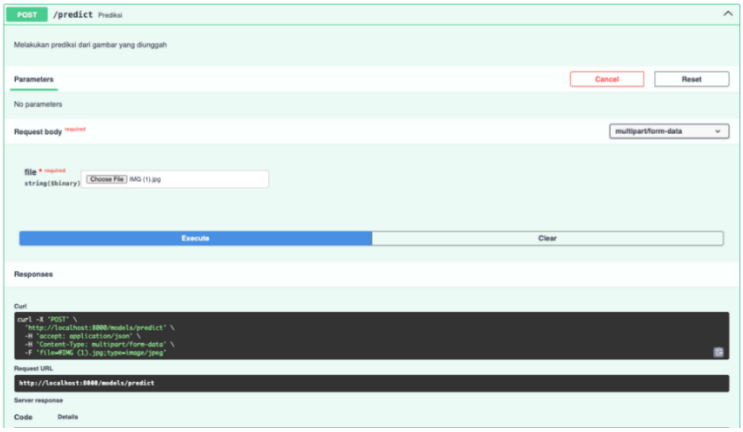
Gambar 5 adalah EndPoint API(GET) yang digunakan untuk pengecekan API dengan method GET dengan response body:

DOI: 10.52362/jisamar.v10i1.2298

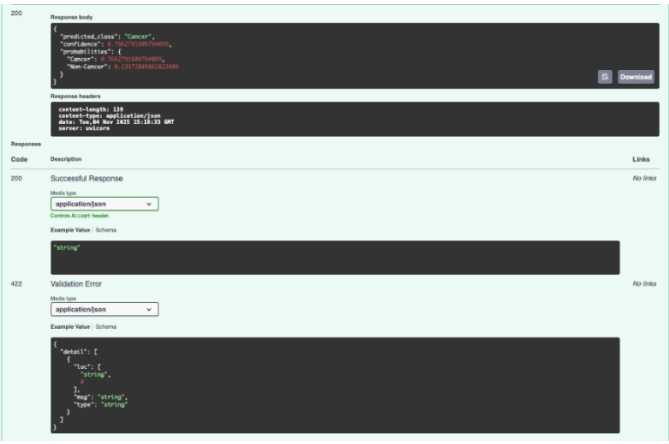
Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)



Gambar 3. Response Api Program Python Berjalan



Gambar 7. Endpoint prediksi

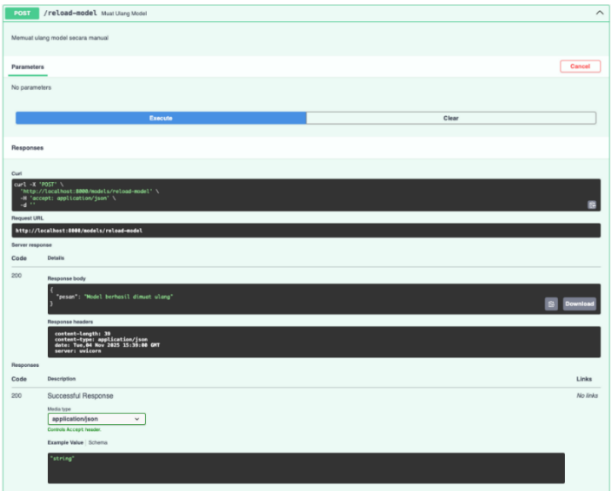


Gambar 8. Response Endpoint prediksi

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

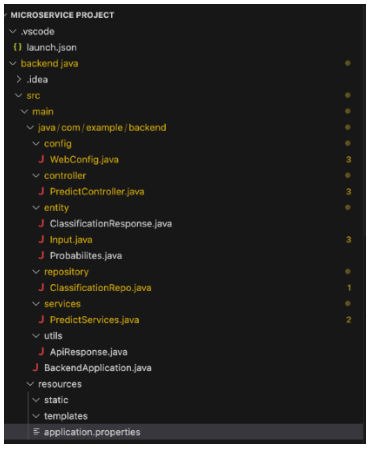
Gambar 8 Endpoint ini digunakan untuk mengirim gambar medis (misalnya mammogram) ke server API agar diproses oleh model deep learning breast_cancer_model.h5. Hasil yang dikembalikan berupa label prediksi ("Cancer" atau "Non-Cancer") beserta tingkat kepercayaan (confidence score).



Gambar 9. endpoint untuk memuat ulang model secara manual

Gambar 9 Endpoint ini berfungsi untuk memuat ulang file model breast_cancer_model.h5 tanpa perlu menghentikan atau me-restart server FastAPI. Hal ini sangat berguna apabila model diperbarui (misalnya versi baru hasil training ulang) dan ingin digunakan langsung oleh API yang sedang berjalan.

2. **Backend Java (Spring Boot):**
   Mengelola permintaan dari frontend, meneruskan file gambar ke service FastAPI, dan menyimpan hasil klasifikasi ke database. Mengandung modul utama seperti Controller, Entity, Repository, dan Service.

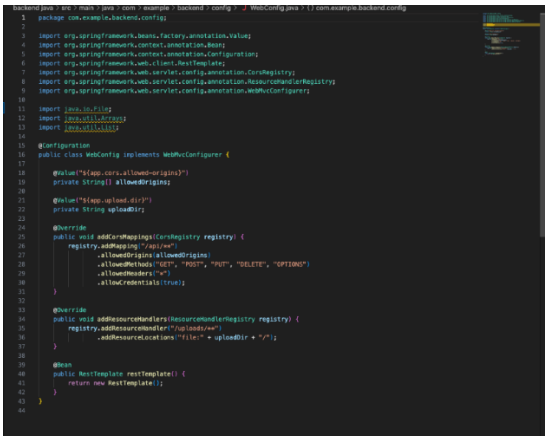


Gambar 10. Struktur Folder Backend Java

1. Config
   Berisi konfigurasi aplikasi seperti pengaturan CORS, bean, dan konfigurasi umum.

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)



Gambar 11. Code WebConfig.java

Gambar 11 WebConfig.java digunakan untuk mengatur konfigurasi web agar API dapat diakses oleh client.

File WebConfig.java digunakan untuk mengatur konfigurasi web pada backend agar dapat berkomunikasi dengan Frontend Angular.js dan mengelola resource statis. Konfigurasi ini mencakup:

a. Pengaturan CORS (Cross-Origin Resource Sharing)

Agar aplikasi Angular.js dapat mengakses API backend tanpa error, metode addCorsMappings digunakan untuk mengizinkan semua origin dan metode HTTP seperti GET, POST, PUT, dan DELETE.

b. Pengaturan Resource Handler

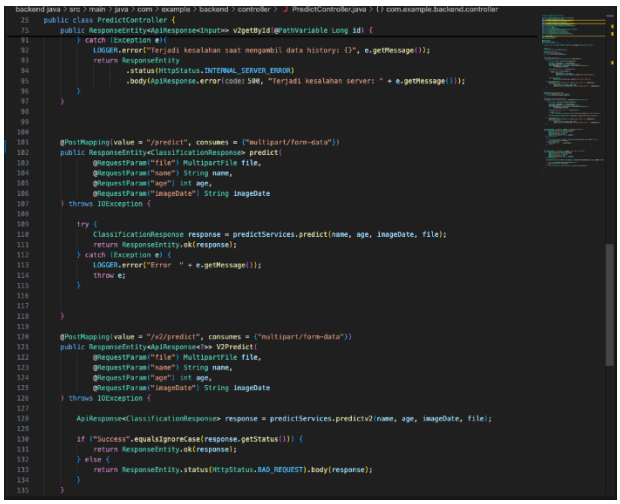
Metode addResourceHandlers digunakan untuk menentukan lokasi file statis yang dapat diakses oleh aplikasi.

c. Pengaturan View Resolver

Metode viewResolver digunakan untuk mengatur template engine jika diperlukan.

1. Controller

   Berisi class yang mengelola endpoint REST API untuk menerima request dari Frontend Angular.js.



Gambar 12. Code PredictController.java

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

Berisi PredictController.java menangani request prediksi dan memanggil service untuk menghubungkan ke API FastAPI.
PredictController.java adalah komponen controller pada backend Java yang berfungsi untuk:

a. Menyediakan endpoint REST API agar Frontend Angular.js dapat mengakses data dan melakukan prediksi.
b. Mengelola request untuk:
    1. Mengambil riwayat prediksi dari database MySQL.
    2. Mengambil detail riwayat berdasarkan ID.
    3. Mengunggah gambar untuk prediksi dan meneruskan ke PredictServices yang akan memanggil API FastAPI.

3. Frontend Angular.js:

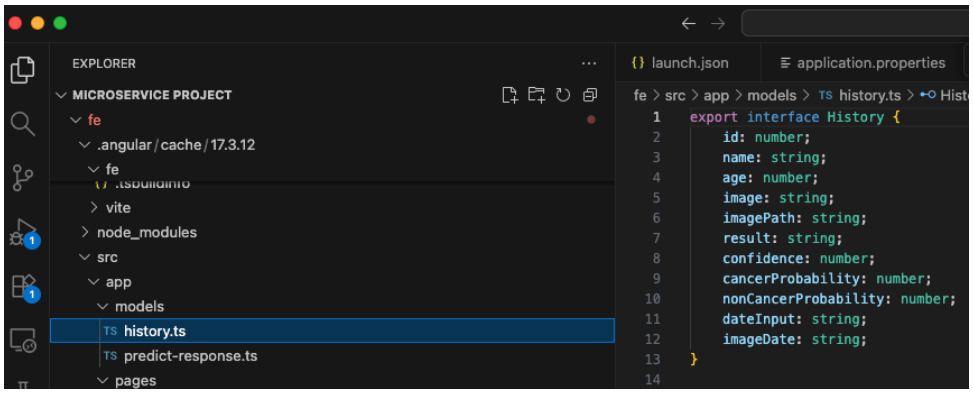
Menyediakan tiga halaman utama: Home, Predict, dan History, dengan navigasi sederhana dan tampilan responsif.

Gambar 13. Struktur code FE (Angular.js)

1. Folder models/
   Folder ini berfungsi untuk menampung model data yang digunakan pada aplikasi frontend.
   Model ditulis menggunakan TypeScript dalam bentuk *interface* agar struktur data yang diterima dari API backend dapat terdefinisi dengan jelas.
   1) history.ts
   Digunakan untuk mendefinisikan struktur data riwayat hasil prediksi. Contoh atribut yang digunakan antara lain: id, tanggal_prediksi, dan hasil_prediksi.

DOI: 10.52362/jisamar.v10i1.2298

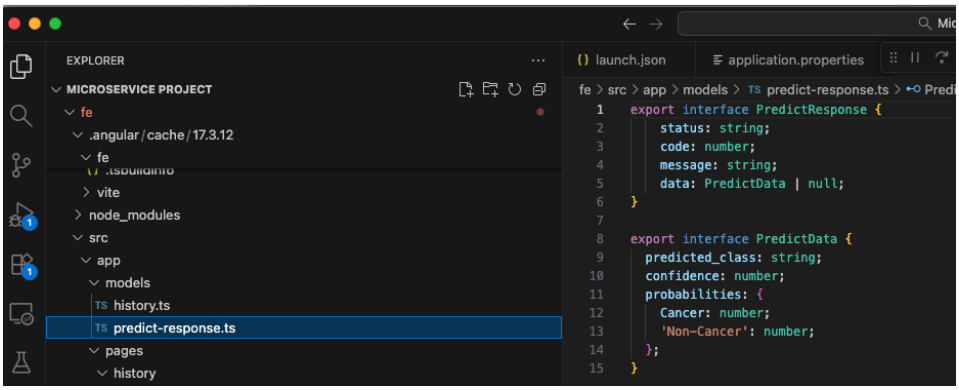
Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)



Gambar 14. Code history.ts

2) predict-response.ts

Menampung struktur respon hasil prediksi dari microservice machine learning. Model ini biasanya berisi atribut seperti label (hasil klasifikasi) dan confidence (tingkat kepercayaan model).



Gambar 15. Code Predict-response.ts

Dengan adanya folder models, pengelolaan tipe data menjadi lebih konsisten dan memudahkan proses *binding* antara komponen dan data API.

1. Folder pages/

Folder ini merupakan inti dari aplikasi Angular karena berisi kumpulan komponen halaman (page components) yang menampilkan berbagai fitur pada sistem.
Masing-masing halaman diimplementasikan dalam tiga file utama:

1. **.ts** : logika program (component logic),
2. **.html** : tampilan (template view),
3. **.scss** : gaya tampilan (style sheet).

Adapun halaman yang dibangun meliputi:

1. **main/**
   Halaman utama atau dashboard aplikasi. Komponen ini berfungsi menampilkan ringkasan informasi serta menjadi titik navigasi ke fitur lain seperti prediksi dan riwayat.
2. **predict/**
   Halaman untuk melakukan proses prediksi. Pengguna dapat memasukkan data atau mengunggah gambar, kemudian data tersebut akan dikirim ke backend

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

microservice untuk dilakukan proses klasifikasi. Hasil prediksi kemudian ditampilkan melalui binding data pada file predict.component.html.

3. **history/**
   Halaman ini menampilkan daftar riwayat hasil prediksi yang telah dilakukan sebelumnya. Data diambil melalui pemanggilan API ke microservice history service. Komponen ini juga memungkinkan pengguna melihat detail hasil prediksi pada halaman history-detail.

Struktur berbasis halaman ini memudahkan modularisasi kode dan meningkatkan keterbacaan (code maintainability).

1. File Utama Aplikasi
   1. **app.component.ts**
      Merupakan komponen utama (*root component*) pada aplikasi Angular yang menjadi induk dari seluruh komponen lainnya.
      Di dalam file ini terdapat *router-outlet* yang berfungsi untuk menampilkan halaman sesuai rute yang aktif.
   2. **app.component.html dan app.component.scss**
      Masing-masing digunakan untuk mendefinisikan struktur dan gaya tampilan global aplikasi.
   3. **app.routes.ts**
      Berfungsi untuk mendefinisikan *routing* antar halaman dalam aplikasi. Misalnya, rute '/' diarahkan ke MainComponent, rute '/predict' ke PredictComponent, dan rute '/history' ke HistoryComponent. Dengan adanya sistem routing ini, navigasi antar halaman menjadi dinamis tanpa perlu memuat ulang seluruh aplikasi.
   4. **app.config.ts dan app.config.server.ts**
      Digunakan untuk menyimpan konfigurasi global aplikasi seperti alamat *base URL API*, pengaturan environment, serta inisialisasi modul tambahan.
2. Folder assets/
   Folder ini berisi sumber daya statis (*static assets*) seperti gambar, file CSS tambahan, font, serta berkas JavaScript eksternal. Semua aset di dalam folder ini dapat digunakan oleh berbagai komponen dalam aplikasi tanpa harus mengulangi definisinya.

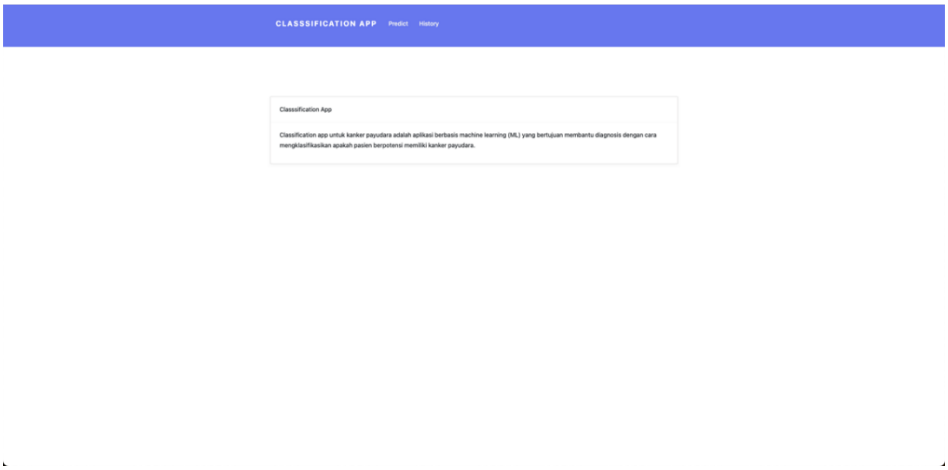
3.2. Implementasi User Interface

Antarmuka sistem terdiri atas tiga halaman:

1. **Home Page**
   Menampilkan informasi umum sistem dan navigasi utama.
   Halaman Home merupakan tampilan awal ketika pengguna membuka classification apps. Halaman ini berfungsi sebagai beranda utama yang memberikan informasi singkat mengenai tujuan dan fungsi dari sistem yang dikembangkan.Pada bagian atas halaman terdapat navbar berwarna biru keunguan yang berisi judul aplikasi “Classification.App” serta beberapa menu navigasi seperti Predict dan Tools. Menu ini memudahkan pengguna untuk berpindah dari halaman beranda ke halaman lain dalam aplikasi dapat dilihat pada gambar 16.

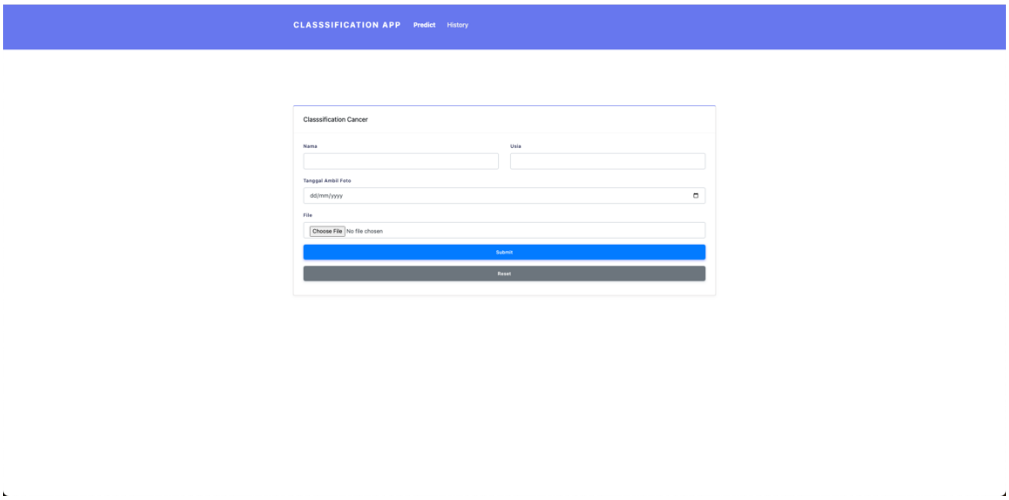
DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)



Gambar 16. Halaman Home Classification Cancer App

2. **Predict Page**
Menyediakan form upload gambar (mammogram), tombol "Predict" untuk mengirim ke backend, dan hasil klasifikasi lengkap dengan tingkat kepercayaan (*confidence*).
Halaman Predict merupakan halaman utama yang digunakan untuk melakukan proses klasifikasi gambar bangunan berdasarkan model machine learning yang telah dilatih sebelumnya. Halaman ini menjadi inti dari sistem karena di sinilah pengguna dapat melakukan pengujian langsung terhadap model prediksi gaya arsitektur.
Tampilan halaman dibuat dengan desain sederhana dan responsif agar mudah digunakan. Di bagian atas, terdapat navbar berwarna biru keunguan yang berisi judul aplikasi "Klasifikasi Gaya Arsitektur" serta menu navigasi seperti Home dan Predict, sehingga pengguna dapat berpindah antarhalaman dengan mudah.



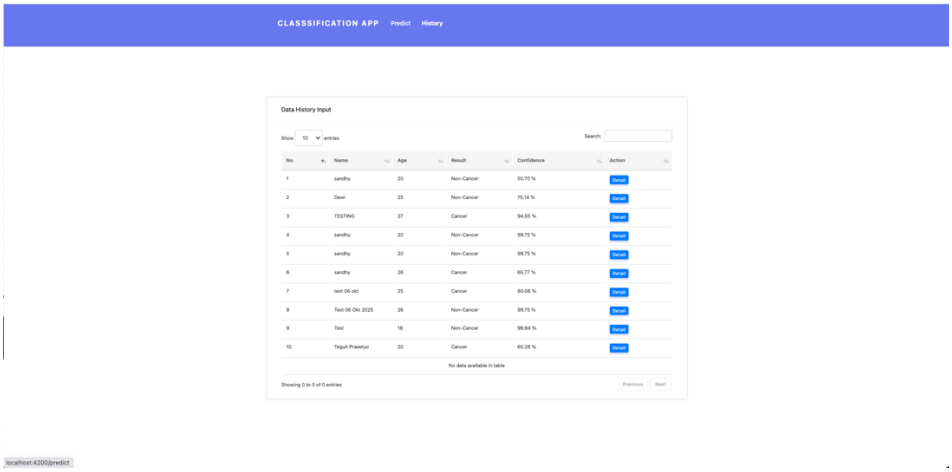
Gambar 17. Halaman Predict

3. **History Page**
Menampilkan riwayat hasil prediksi dalam tabel interaktif yang dapat difilter berdasarkan nama, usia, dan tanggal. Dari sisi desain, tampilan halaman History dirancang sederhana, bersih, dan profesional, menggunakan warna dasar putih dengan aksen biru pada tombol aksi. Fokus desain diarahkan pada keterbacaan data dan kemudahan navigasi, sesuai prinsip user-centered design.

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

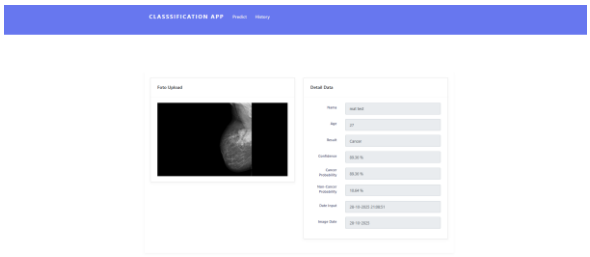
Dengan adanya halaman ini, sistem menjadi lebih transparan dan akuntabel, karena setiap hasil prediksi tersimpan dengan rapi dan dapat diakses kembali kapan saja. Hal ini juga memudahkan pengguna atau peneliti dalam melakukan analisis performa model berdasarkan data historis yang telah diklasifikasikan sebelumnya dapat dilihat pada gambar 17.



Gambar 18. Halaman History

4. History Detail

Merupakan tampilan hasil detail dari history sebelumnya saat menekan button detail, pada tampilan ini menampilkan gambar hasil klasifikasi, prediksi dan tanggal melakukan predikisi serta detail lainnya.



Gambar 19. Halaman History Detail

3.3. Pengujian Fungsional

Pengujian dilakukan menggunakan Postman untuk memastikan seluruh endpoint berjalan sesuai fungsinya:

1. GET/ untuk mengecek status API.
2. POST/predict untuk mengunggah gambar dan menerima hasil klasifikasi.
3. GET/reload-model untuk memuat ulang model AI secara dinamis.

Seluruh endpoint memberikan respons valid dan tidak mengalami timeout selama pengujian.

DOI: 10.52362/jisamar.v10i1.2298

Journal of Information System, Applied, Management, Accounting and Research.
http://journal.stmikjayakarta.ac.id/index.php/jisamar ,
jisamar@stmikjayakarta.ac.id , jisamar2017@gmail.com
e-ISSN: 2598-8719 (Online), p-ISSN: 2598-8700 (Printed) , Vol. 10 No.1 (February 2026)

3.4. Pengujian Performa

Pengujian performa menggunakan **Postman** untuk mengukur *response time* setiap endpoint.

Table 1. Hasil Pengujian Performa

Endpoint	Deskripsi	Response Time	Kategori
GET/	Health Check	8.39 ms	Sangat Cepat
POST/predict	Prediksi Gambar	654 ms	Ideal
GET/reload-model	Pemuatan Ulang Model	971.99 ms	Baik

Hasil pengujian menunjukkan seluruh layanan merespons di bawah 1 detik, sesuai standar REST API performa tinggi.

.

IV. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi sistem klasifikasi gambar kanker payudara berbasis arsitektur microservices, dapat disimpulkan bahwa pendekatan ini berhasil memberikan solusi yang lebih modular, fleksibel, dan mudah dikembangkan dibandingkan arsitektur monolitik. Sistem yang dibangun terdiri dari tiga layanan utama, yaitu layanan pra-pemrosesan, layanan klasifikasi berbasis model deep learning, serta layanan penyimpanan hasil, yang semuanya berkomunikasi melalui REST API dan berjalan secara independen. Implementasi menggunakan FastAPI untuk layanan model terbukti mampu menangani proses inferensi dengan cepat, sedangkan backend Spring Boot mampu mengatur alur komunikasi data secara stabil. Hasil pengujian performa menunjukkan bahwa seluruh endpoint merespons dalam waktu di bawah satu detik, dengan kinerja terbaik pada endpoint health check dan kinerja paling kompleks pada endpoint pemuatan ulang model. Hal ini menegaskan bahwa microservices sangat efektif untuk aplikasi berbasis AI yang membutuhkan pemrosesan intensif dan skalabilitas tinggi. Selain itu, integrasi antar-layanan berjalan dengan baik tanpa terjadi bottleneck yang signifikan, menunjukkan bahwa desain sistem mendukung efisiensi, ketahanan, dan kemudahan pemeliharaan. Dengan demikian, penelitian ini membuktikan bahwa arsitektur microservices dapat menjadi pendekatan ideal bagi sistem diagnosis medis berbasis machine learning yang membutuhkan kecepatan respons, modularitas, serta kemudahan pengembangan jangka panjang.

Meskipun sistem klasifikasi gambar kanker payudara yang dikembangkan telah menunjukkan kinerja yang baik, masih terdapat beberapa aspek yang dapat ditingkatkan pada penelitian selanjutnya. Pengembangan model deep learning sebaiknya dilakukan menggunakan dataset citra medis yang lebih besar, beragam, dan tervalidasi secara klinis agar menghasilkan akurasi dan tingkat kepercayaan yang lebih tinggi. Selain itu, penerapan mekanisme keamanan seperti autentikasi API, enkripsi data, dan kontrol akses perlu diperkuat mengingat sistem ini berkaitan dengan data medis yang bersifat sensitif. Infrastruktur sistem juga dapat ditingkatkan dengan menerapkan container orchestration seperti Kubernetes secara penuh untuk meningkatkan skalabilitas, fault tolerance, dan kemampuan load balancing terutama pada kondisi beban tinggi. Dari sisi fungsional, sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur pencatatan metadata pasien secara terstruktur, visualisasi riwayat prediksi yang lebih informatif, serta kemampuan integrasi dengan sistem rumah sakit berbasis HL7 atau FHIR. Penelitian selanjutnya juga disarankan untuk melakukan pengujian menggunakan data nyata dari institusi medis guna memperoleh validasi klinis yang lebih komprehensif. Selain itu, pengembangan aplikasi versi mobile atau responsive web dapat meningkatkan aksesibilitas bagi tenaga medis di lapangan. Secara keseluruhan, berbagai pengembangan ini diharapkan dapat meningkatkan keandalan, keamanan, dan nilai praktis sistem sehingga dapat digunakan sebagai alat bantu diagnosis yang lebih efektif dan siap diimplementasikan pada lingkungan medis sesungguhnya.

DOI: 10.52362/jisamar.v10i1.2298

REFERENASI

[1] I.Sari, M. Septiana, A. Sapitri, and Stik. Budi Mulia Sriwijaya, "Peningkatan Perilaku SADARI (Periksa Payudara Sendiri) pada Perempuan Terhadap Upaya Deteksi Dini Kanker Payudara," May 2023.

[2] Edwin Febrywinata, "Pengenalan Dan Klasifikasi Jenis Buah Menggunakan Metode CNN Secara Sederhana Dengan Menggunakan Google Colab," *Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika*, vol. 2, no. 4, pp. 185–193, Jun. 2024, doi: 10.61132/merkurius.v2i4.162.

[3] R. Hayami, "KLASIFIKASI TEKS BERITA BERBAHASA INDONESIA MENGGUNAKAN MACHINE LEARNING DAN DEEP LEARNING: STUDI LITERATUR," 2023. [Online]. Available: https://ieeexplore.ieee.org/

[4] M. Iqbal *et al.*, "PENERAPAN MONOLITIC ARSITECTURE PADA APLIKASI UJIAN ONLINE BERBASIS WEB," 2022. [Online]. Available: http://jurnal.goretanpena.com/index.php/JPSTM

[5] S. Siregar, "Analisis Keuntungan Menggunakan Microservices dalam Pengembangan Aplikasi Skala Besar," 2022.

[6] A. Jahiduddin, E. Sakti Pramukantoro, and F. A. Bakhtiar, "Pengembangan Infrastruktur Analisis Data Heart Rate berbasis Microservices menggunakan Kubernetes," 2020. [Online]. Available: http://j-ptiik.ub.ac.id

[7] S. Andriati Asri *et al.*, "Implementation of PWA in Scholarship Application Using Microservices Architecture for Enhancing User Engagement," 2024, doi: 10.32996/jcsts.

[8] R. Fiqri Syah, S. Renny Puspita, and R. Syahru, "Perbandingan Performa Web Services Yang Dibangun Menggunakan Arsitektur Monolithic Dan Microservices Pada Sistem Point of Sales," Mar. 2023.

[9] Hyunseok Chang, Murali Kodialam, T.V. Lakshman, and Sarit Mukherjee, *Microservice Fingerprinting and Classification using Machine Learning*. IEEE, 2019.

[10] Sugiyono, *Metode Penelitian Dan Pengembangan Research Dan Development*. Bandung : Alfabeta, 2019.

DOI: 10.52362/jisamar.v10i1.2298