Merkurius : Jurnal Riset Sistem Informasi dan Teknik Informatika
Volume 3 Nomor 5, September 2025
e-ISSN : 3031-8912; p-ISSN : 3031-8904; Hal 82-96
DOI: https://doi.org/10.61132/merkurius.v3i5.1071
Tersedia: https://journal.arteii.or.id/index.php/Merkurius

Deployment Aplikasi Deteksi Sampah pada *Google Cloud*

Rifqi Arief Pamungkas [1*], Isram Rasal [2]
[1-2] Informatika, Fakultas Teknologi Industri, Universitas Gunadarma, Indonesia
E-mail: ug.51421315@gmail.com [1], isramrasal@staff.gunadarma.ac.id [2]*

Alamat: Jalan Margonda Raya Nomor 100, Pondok Cina, Depok, Jawa Barat 16424
[*]Korespondensi penulis

Abstract*. The waste detection application is a solution to identify and classify waste. With the increasing waste problem, this application aims to assist in classifying waste into organic, inorganic, and unknown categories. This research implements a waste detection application that runs on smartphones with the Android operating system. The application is the result of a capstone project from the Bangkit program in the MSIB batch 6. The development of this application is divided into several parts, namely backend, frontend, and deployment. The author focuses on the deployment process of the application using Google Cloud Platform. The Google Cloud Platform infrastructure was chosen due to its scalability and flexibility. The services utilized include Google Compute Engine (GCE), Google Virtual Private Cloud (VPC), and Google Cloud Storage (GCS). The deployment process involves creating a new project in Google Cloud, configuring virtual machines, and setting up Google Cloud Storage. The server VM configuration includes the installation of SQL Server (MariaDB), deployment of the Machine Learning API, and Backend API. Testing was carried out on the Machine Learning API using Postman and the Backend API through a browser. The results show that Google Cloud Platform can be implemented as a cloud computing infrastructure for waste detection applications. The Machine Learning API is able to read objects in the form of images sent by users.*

Keywords*: Cloud; Deployment; Detection; Google; Trash.*

Abstrak. Aplikasi pendeteksi sampah merupakan solusi untuk melakukan identifikasi dan klasifikasi sampah. Dengan meningkatnya permasalahan sampah, aplikasi ini bertujuan untuk membantu klasifikasi sampah menjadi organik, anorganik, dan tidak diketahui. Penelitian ini mengimplementasikan aplikasi pendeteksi sampah yang berjalan di ponsel pintar dengan sistem operasi Android. Aplikasi ini merupakan hasil proyek capstone Bangkit pada program MSIB batch 6. Pengembangan aplikasi ini dibagi menjadi beberapa bagian, yaitu backend, frontend, dan deployment. Penulis berfokus pada proses deployment aplikasi menggunakan Google Cloud Platform. Infrastruktur Google Cloud Platform dipilih karena skalabilitas dan fleksibilitasnya. Layanan yang dimanfaatkan meliputi Google Compute Engine (GCE), Google Virtual Private Cloud (VPC), dan Google Cloud Storage (GCS). Proses deployment melibatkan pembuatan proyek baru di Google Cloud, konfigurasi mesin virtual, dan pengaturan Google Cloud Storage. Konfigurasi server VM meliputi instalasi SQL Server (MariaDB), deployment Machine Learning API, dan Backend API. Pengujian dilakukan terhadap Machine Learning API menggunakan Postman dan Backend API melalui browser. Hasil penelitian menunjukkan bahwa Google Cloud Platform dapat diimplementasikan sebagai infrastruktur komputasi awan untuk aplikasi pendeteksi sampah. Machine Learning API mampu membaca objek berupa gambar yang dikirimkan.

Kata kunci: Cloud; Deployment; Deteksi; Google; Sampah.

1. LATAR BELAKANG

Permasalahan sampah di Indonesia semakin serius dan memerlukan perhatian khusus. Berdasarkan laporan Sistem Informasi Pengelolaan Sampah Nasional (SIPSN) Kementerian Lingkungan Hidup dan Kehutanan (KLHK) tahun 2024, volume timbulan sampah nasional telah mencapai 34,2 juta ton per tahun dan sebagian besar belum dikelola dengan baik (KLHK, 2024). Kondisi ini menimbulkan berbagai konsekuensi negatif, di antaranya pencemaran lingkungan, banjir, serta ancaman terhadap kesehatan masyarakat. Oleh karena itu, diperlukan

Naskah Masuk: 15 Agustus 2025, Revisi: 30 Agustus 2025, Diterima: 20 September 2025, Terbit: 22 September 2025

strategi pengelolaan yang lebih efektif dengan memanfaatkan teknologi terkini, salah satunya melalui penerapan kecerdasan buatan untuk mendukung pemilahan dan daur ulang sampah.

Kemajuan dalam bidang machine learning membuka peluang baru dalam upaya penanganan permasalahan sampah. Teknologi ini memungkinkan dikembangkannya aplikasi berbasis visi komputer yang mampu mengklasifikasikan sampah menjadi kategori organik, anorganik, maupun residu secara otomatis. Hal tersebut dapat membantu masyarakat dalam melakukan pemilahan sampah dengan lebih mudah dan cepat (Bisong, 2019). Beberapa penelitian terdahulu menunjukkan keberhasilan penggunaan teknologi pengenalan gambar untuk deteksi sampah, meskipun masih terdapat tantangan seperti keterbatasan akurasi klasifikasi dan kendala pada infrastruktur pendukung (Irawan, 2024).

Perkembangan teknologi cloud computing turut berperan dalam mendukung implementasi sistem berbasis kecerdasan buatan. Google Cloud Platform (GCP), misalnya, menyediakan layanan inti seperti Google Compute Engine (GCE), Google Cloud Storage (GCS), dan Virtual Private Cloud (VPC) yang dapat dimanfaatkan untuk meningkatkan skalabilitas, fleksibilitas, serta efisiensi pengembangan aplikasi modern (Bisong, 2019). Penelitian terkini membuktikan bahwa penggunaan GCP untuk deployment aplikasi memberikan keunggulan dari sisi ketersediaan layanan, kecepatan pemrosesan, serta keandalan sistem (Irawan, 2024).

Penelitian ini bertujuan untuk mengimplementasikan aplikasi deteksi sampah berbasis machine learning pada perangkat Android dengan dukungan Google Cloud Platform. Fokus penelitian diarahkan pada tahap deployment aplikasi sehingga dapat diakses oleh pengguna secara optimal. Dalam implementasinya, layanan inti GCP seperti GCE untuk komputasi, VPC untuk pengelolaan jaringan, dan GCS untuk penyimpanan data dimanfaatkan secara terpadu. Hasil penelitian ini diharapkan dapat memberikan kontribusi nyata dalam pemanfaatan teknologi cloud untuk solusi pengelolaan sampah di Indonesia, sekaligus memperkuat upaya pelestarian lingkungan.

2. KAJIAN TEORITIS

Cloud Computing

Cloud computing adalah paradigma komputasi modern yang memungkinkan pengaksesan layanan seperti penyimpanan, server, dan aplikasi melalui internet tanpa harus memiliki infrastruktur fisik secara langsung, sehingga efisien dalam biaya dan sangat fleksibel (IBM, 2024; TechRadar, 2025). Teknologi ini memungkinkan penyesuaian kapasitas komputasi—baik naik maupun turun—sesuai kebutuhan langsung (*on-demand*), yang sangat

vital untuk aplikasi berbasis *machine learning* maupun layanan real-time lainnya (AWS Whitepaper, 2025; CompanionLink, 2025).

Google Cloud Platform

Google Cloud Platform (GCP) merupakan salah satu penyedia layanan *cloud computing* yang menawarkan berbagai fasilitas untuk komputasi, penyimpanan, jaringan, serta integrasi *machine learning* dalam satu ekosistem. Layanan utama seperti Google Compute Engine (GCE), Google Cloud Storage (GCS), dan Virtual Private Cloud (VPC) memungkinkan pengembang membangun aplikasi dengan performa tinggi, skalabilitas yang mudah, serta fleksibilitas sesuai kebutuhan sistem (Pratama & Nugroho, 2022).

Keunggulan GCP terletak pada kemampuannya menyediakan infrastruktur yang andal, aman, dan efisien untuk pengembangan maupun *deployment* aplikasi modern. Penelitian terdahulu menunjukkan bahwa pemanfaatan GCP dapat meningkatkan efisiensi penyimpanan data dan mendukung ketersediaan layanan dalam skala besar, sehingga menjadi solusi yang relevan untuk aplikasi berbasis *machine learning* dan integrasi data real-time (Bisong, 2019; Irawan, 2024).

MariaDB

MariaDB adalah sistem manajemen basis data relasional (RDBMS) yang dikembangkan oleh tim yang sebelumnya membangun MySQL, sehingga keduanya memiliki banyak kesamaan baik dari sisi struktur maupun fungsionalitas dasar (Christiono & Sama, 2020). MariaDB mempertahankan kompatibilitas penuh dengan MySQL sehingga mudah digunakan dalam migrasi maupun integrasi aplikasi yang sebelumnya berbasis MySQL, namun memiliki peningkatan pada sisi kinerja, keamanan, serta fleksibilitas. Keunggulan MariaDB juga terletak pada skalabilitasnya yang memungkinkan pengelolaan data dalam jumlah besar dan mendukung aplikasi berbasis *cloud*. Dalam penelitian berbasis aplikasi modern, MariaDB banyak digunakan untuk mengelola data karena sifatnya yang ringan, stabil, serta mendukung *deployment* pada berbagai platform, termasuk Google Cloud Platform.

Golang

Golang atau Go adalah bahasa pemrograman open-source yang dikembangkan oleh Google dengan tujuan utama menyederhanakan proses pengembangan perangkat lunak modern. Dalam konteks pengembangan aplikasi berbasis *cloud*, Golang banyak digunakan untuk membangun *backend services* karena performanya yang tinggi, skalabilitas, serta dukungan ekosistem pustaka yang luas. Keunggulan inilah yang menjadikan Golang cocok untuk implementasi layanan API yang di-*deploy* pada Google Cloud Platform (Albar, 2017).

Python dan Machine Learning

Python merupakan bahasa pemrograman serbaguna yang banyak dimanfaatkan dalam pengembangan situs web, perangkat lunak, otomasi tugas, analisis data, hingga pembelajaran mesin. Sebagai bahasa pemrograman umum, Python tidak terbatas pada permasalahan tertentu, melainkan dapat digunakan untuk berbagai macam aplikasi. Fleksibilitas dan kemudahan penggunaannya menjadikan Python salah satu bahasa pemrograman yang paling populer dan banyak digunakan di dunia saat ini (Anonim, 2024).

Machine Learning (ML) merupakan salah satu pendekatan kecerdasan buatan yang berfungsi untuk menirukan perilaku manusia dalam menyelesaikan tugas atau melakukan otomatisasi (Noviana & Rasal, 2023). Pada dasarnya, ML berusaha meniru cara manusia maupun makhluk cerdas belajar, memahami pola, dan melakukan generalisasi dari data yang tersedia (Ahmad, 2017). Dalam penerapannya, klasifikasi digunakan untuk mengelompokkan objek berdasarkan karakteristik tertentu, sementara prediksi atau regresi berperan dalam memperkirakan keluaran dari suatu masukan berdasarkan pola yang telah dipelajari melalui proses pelatihan (Effendi, 2021).

Postman

Postman adalah sebuah aplikasi yang berfungsi sebagai klien REST (*Representational State Transfer*) yang digunakan untuk mendukung proses pengujian serta pengembangan API (*Application Programming Interface*). Aplikasi ini dirancang agar memudahkan pengembang dalam melakukan validasi terhadap fungsionalitas API yang telah dibangun, sehingga Postman menjadi salah satu alat yang umum digunakan oleh para pengembang perangkat lunak (Kurniawan, 2020).

3. METODE PENELITIAN

Penelitian ini menggunakan metode Software Development Life Cycle (SDLC) model Waterfall untuk merancang dan mengimplementasikan aplikasi deteksi sampah berbasis *machine learning* yang di-*deploy* pada Google Cloud Platform (Noviana, 2022). SDLC merupakan metode pengembangan perangkat lunak yang terstruktur (Setiany et al., 2021). Metode SDLC banyak digunakan dalam rekayasa perangkat lunak karena mampu memberikan kerangka kerja yang terstruktur dan sistematis dalam proses pembangunan suatu sistem (Pargaonkar, 2023); (Alazzawi et al., 2023).

Tahapan penelitian dilakukan secara sistematis untuk memastikan aplikasi dapat berjalan sesuai kebutuhan. Adapun tahapan SDLC yang diterapkan dalam penelitian ini meliputi: (a) Tahap Perencanaan Pada tahap ini dilakukan persiapan awal yang mencakup

instalasi perangkat lunak yang dibutuhkan seperti MariaDB, Go, dan Python. Selain itu, peneliti juga mengumpulkan informasi dari berbagai sumber, termasuk artikel ilmiah, e-book, serta referensi daring yang relevan. Tujuan utama dari tahap ini adalah memastikan seluruh perangkat dan informasi pendukung tersedia sebelum memulai proses perancangan sistem. (b) Tahap Analisis Kebutuhan Tahap analisis kebutuhan dilakukan untuk mengidentifikasi spesifikasi teknis dan fungsional dari sistem yang akan diimplementasikan. Analisis difokuskan pada kebutuhan infrastruktur deployment, seperti spesifikasi perangkat keras, sistem operasi, serta perangkat lunak pendukung. Hasil analisis ini menjadi dasar dalam menentukan layanan Google Cloud Platform yang akan digunakan, yaitu Google Compute Engine (GCE), Google Cloud Storage (GCS), dan Google Virtual Private Cloud (VPC). (c) Tahap Perancangan Pada tahap ini dilakukan perancangan deployment aplikasi dengan menyusun bagan alir (flowchart) proses implementasi. Perancangan meliputi pembuatan proyek baru pada Google Cloud, konfigurasi mesin virtual, serta penyusunan alur komunikasi antara Machine Learning API, Backend API, dan basis data. Perancangan ini bertujuan untuk memberikan gambaran teknis yang jelas sebelum implementasi dilakukan. (d) Tahap Implementasi Tahap implementasi mencakup proses deployment aplikasi pada infrastruktur Google Cloud Platform. Aktivitas yang dilakukan meliputi konfigurasi server VM dengan instalasi SQL Server (MariaDB), deployment Machine Learning API menggunakan Python, serta deployment Backend API menggunakan bahasa Go. Selain itu, dilakukan pengaturan Google Cloud Storage untuk penyimpanan data gambar yang dikirimkan oleh pengguna aplikasi. (e) Tahap Uji Coba Tahap ini bertujuan untuk mengevaluasi keberhasilan proses deployment. Pengujian dilakukan terhadap Machine Learning API menggunakan aplikasi Postman untuk mengirim data gambar dan memvalidasi respon klasifikasi. Sementara itu, Backend API diuji melalui browser untuk memastikan fungsi layanan berjalan dengan baik. Uji coba ini menjadi tolok ukur bahwa aplikasi telah berhasil di-deploy dan dapat digunakan sesuai kebutuhan.

4. HASIL DAN PEMBAHASAN

Gambaran Umum

Aplikasi deteksi sampah dirancang dengan memanfaatkan metode *Software Development Life Cycle* (SDLC) model *Waterfall* yang bertujuan untuk melakukan identifikasi dan klasifikasi sampah ke dalam tiga kategori, yaitu organik, anorganik, serta kategori tidak dikenal. Implementasi aplikasi dilakukan pada perangkat telepon pintar berbasis Android dan dikembangkan sebagai bagian dari proyek *capstone* program Bangkit MSIB batch 6. Proses

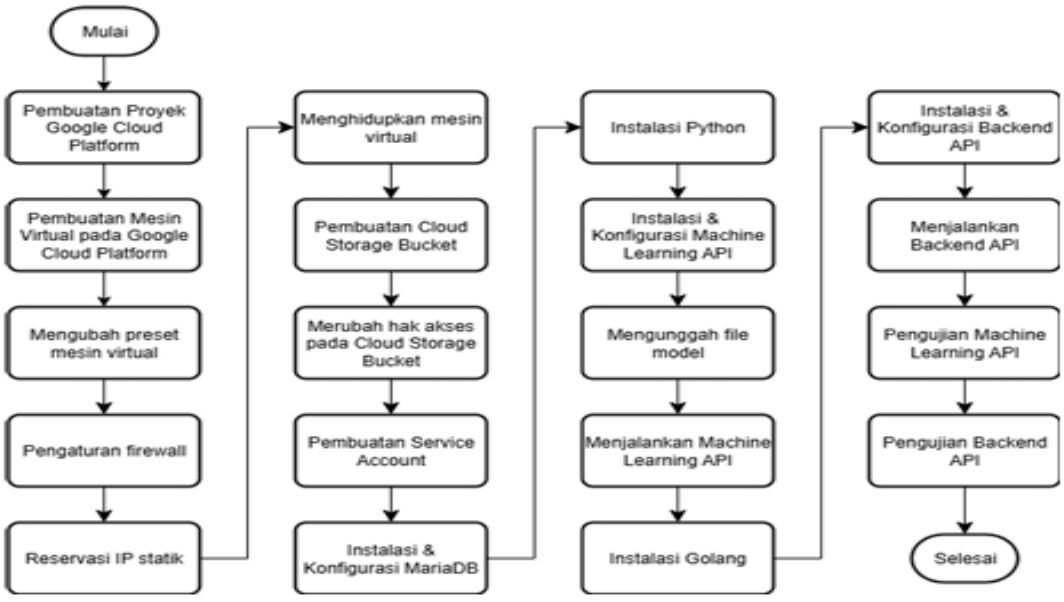
pengembangan dilaksanakan secara kolaboratif dengan pembagian tugas pada aspek *backend*, *frontend*, *machine learning API*, *backend API*, serta proses *deployment*. Fokus penelitian ini diarahkan pada tahap *deployment* aplikasi dengan memanfaatkan infrastruktur yang sesuai untuk mendukung kinerja sistem.

Analisis

Analisis kebutuhan dalam penelitian ini mencakup identifikasi perangkat keras dan perangkat lunak untuk memastikan aplikasi deteksi sampah dapat di-*deploy* secara optimal pada Google Cloud Platform. Perangkat keras yang digunakan adalah komputer HP EliteDesk 705 G2 SFF dengan prosesor AMD PRO A8-6850B, RAM 20 GB, dan penyimpanan 500 GB, sedangkan perangkat lunak yang digunakan meliputi sistem operasi Windows 11 64-bit, Overleaf untuk dokumentasi, Microsoft Edge sebagai peramban, serta Postman untuk pengujian API.

Perancangan

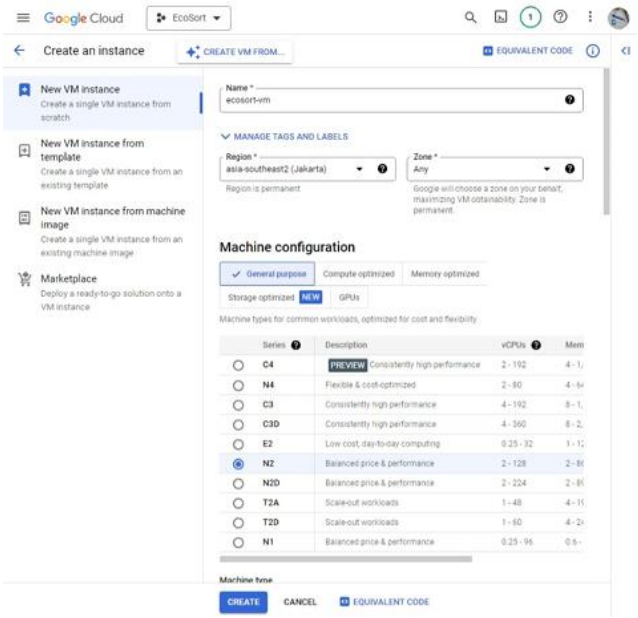
Tahap perancangan dilakukan untuk memberikan gambaran teknis proses *deployment* aplikasi deteksi sampah pada Google Cloud Platform, sebagaimana ditunjukkan pada Gambar 1. Proses dimulai dari pembuatan proyek baru, pembuatan mesin virtual, pengaturan *firewall*, serta reservasi IP statis untuk menjaga kestabilan koneksi. Selanjutnya dilakukan instalasi dan konfigurasi MariaDB sebagai basis data, pembuatan *cloud storage bucket* beserta *service account*, serta instalasi Python untuk menjalankan *Machine Learning API* dan Golang untuk membangun *Backend API*. Tahap ini diakhiri dengan konfigurasi dan eksekusi kedua API tersebut agar sistem dapat berjalan sesuai dengan rancangan penelitian.



Gambar 1. Struktur Navigasi Donatur.

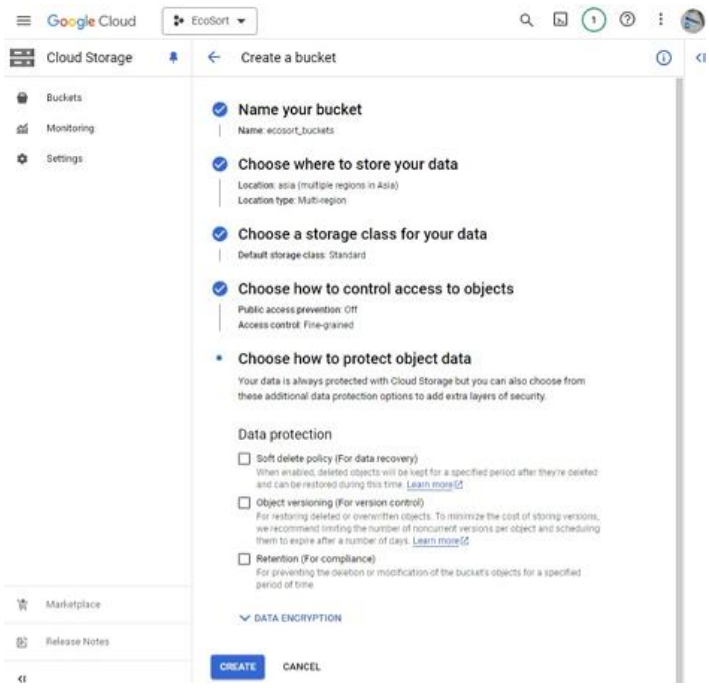
Deployment

Pada tahap awal proses *deployment* menggunakan Google Cloud, langkah pertama yang dilakukan adalah membuat proyek baru yang mencakup pengisian form berisi nama proyek, akun pembayaran, organisasi, serta lokasi, kemudian dilanjutkan dengan pembuatan mesin virtual pada menu *Compute Engine*. Sebelum membuat mesin virtual, pengguna harus terlebih dahulu mengaktifkan *Compute Engine API*, setelah itu barulah dapat membuat *instance* baru dengan menentukan konfigurasi yang dibutuhkan.



Gambar 2. Laman Pembuatan Mesin Virtual.

Konfigurasi mesin virtual meliputi pemilihan tipe mesin, di mana pada penelitian ini digunakan preset n2-standard-4 dengan spesifikasi 4 vCPU dan 16 GB RAM, serta penyesuaian kapasitas penyimpanan dari standar 10 GB menjadi 100 GB dengan menggunakan sistem operasi Debian 12. Selain itu, dilakukan juga pengaturan *firewall* untuk mengizinkan lalu lintas HTTP dan HTTPS agar aplikasi dapat diakses melalui jaringan, serta reservasi IP statik baik internal maupun eksternal guna memastikan alamat IP server tidak berubah. Seluruh konfigurasi tersebut merupakan bagian dari pemanfaatan layanan Google Compute Engine dan Google Virtual Private Cloud, yang bertujuan menyediakan infrastruktur komputasi yang stabil, aman, dan fleksibel untuk kebutuhan deployment aplikasi.



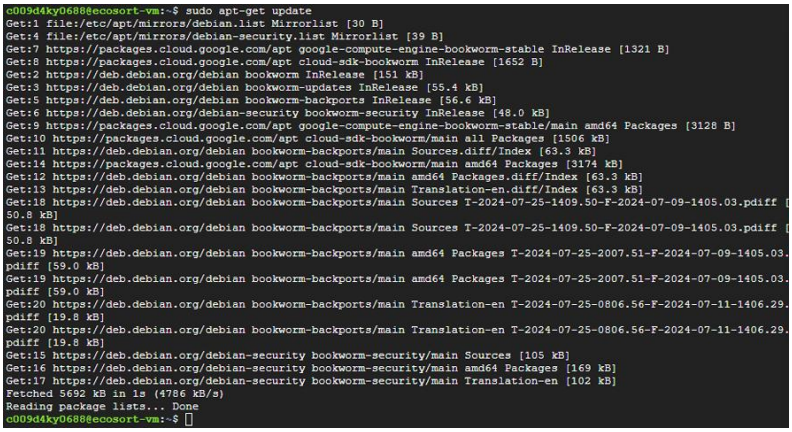
Gambar 3. Pembuatan Storage Buckets.

Tahap berikutnya pada proses *deployment* adalah pemanfaatan layanan Google Cloud Storage (GCS) yang digunakan sebagai media penyimpanan data gambar hasil input pengguna aplikasi deteksi sampah. Proses ini diawali dengan membuat *bucket* baru pada menu *Cloud Storage* dengan mengisi detail yang diperlukan seperti nama *bucket*, lokasi penyimpanan, dan opsi perlindungan data sesuai kebutuhan penelitian. Setelah *bucket* berhasil dibuat, dilakukan pengaturan hak akses agar data yang tersimpan dapat digunakan sesuai tujuan, yaitu dengan menambahkan *principal* baru berupa *allUsers* yang diberi peran sebagai *Storage Object Viewer*, sehingga memungkinkan data gambar diakses oleh aplikasi tanpa mengubah atau menghapus isinya. Selain itu, konfigurasi keamanan tambahan dilakukan melalui pembuatan *service account* dengan peran *Storage Object Admin* dan penyimpanan *private key* dalam format JSON, yang nantinya dipakai untuk mengautentikasi aplikasi ketika mengakses penyimpanan cloud. Dengan konfigurasi ini, GCS dapat berfungsi secara optimal sebagai penyimpanan objek yang aman, terdistribusi, dan mudah diintegrasikan ke dalam sistem, sehingga mendukung kelancaran proses klasifikasi sampah berbasis *machine learning*.

Konfigurasi Infrastruktur

Tahap konfigurasi infrastruktur membahas tahap lanjutan setelah pembuatan dan pengaturan akses pada Google Cloud Platform, yaitu melakukan konfigurasi lingkungan server agar siap digunakan untuk *deployment* aplikasi deteksi sampah. Proses ini diawali dengan membuka konsol mesin virtual melalui fitur *Open in browser window*, kemudian dilakukan

pembaruan paket aplikasi menggunakan perintah sudo apt-get update untuk memastikan sistem memiliki versi paket terbaru yang stabil.

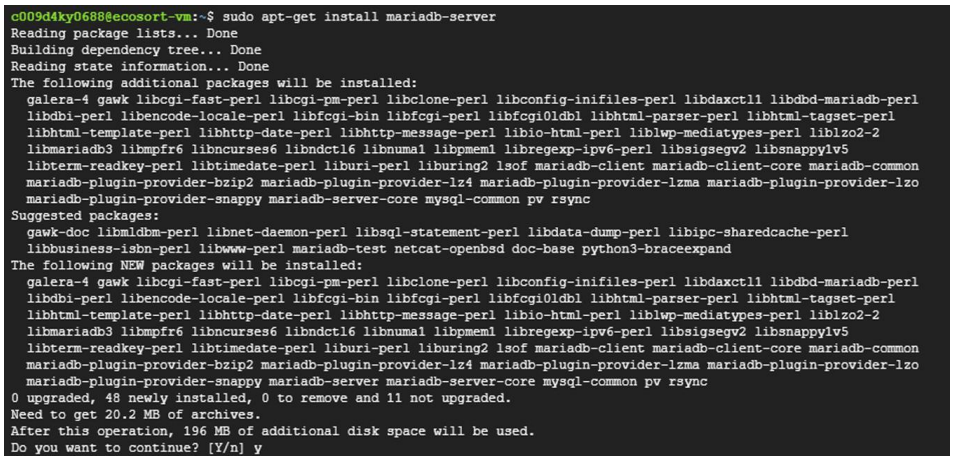


Gambar 4. Pembaruan Paket Aplikasi.

Setelah proses pembaruan selesai, langkah berikutnya adalah melakukan instalasi dan konfigurasi komponen utama yang mendukung jalannya aplikasi, yaitu SQL Server (MariaDB) untuk pengelolaan basis data, Machine Learning API yang digunakan sebagai inti klasifikasi gambar sampah, serta Backend API yang menghubungkan aplikasi dengan layanan penyimpanan dan basis data.

SQL Server (Maria DB)

Konfigurasi SQL Server menggunakan MariaDB dilakukan sebagai langkah awal untuk menyiapkan basis data yang akan mendukung jalannya aplikasi deteksi sampah. Tahap ini dimulai dengan instalasi paket *mariadb-server* menggunakan perintah terminal sudo apt-get install mariadb-server.



Gambar 5. Instalasi Paket MariaDB.

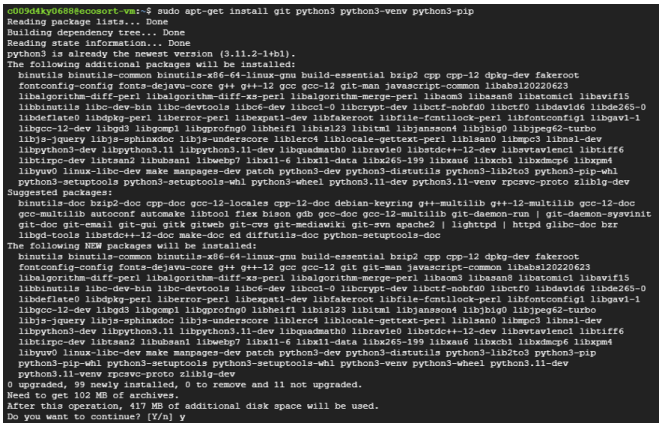
Setelah instalasi selesai, dilakukan konfigurasi keamanan bawaan MariaDB melalui perintah sudo mariadb-secure-installation, yang mencakup pengaturan autentikasi, perubahan kata sandi root, serta penghapusan akun anonim agar sistem lebih aman. Selanjutnya, peneliti

masuk ke dalam MariaDB dengan akses root untuk membuat basis data baru yang berfungsi sebagai tempat penyimpanan data aplikasi. Setelah itu dibuat pula akun pengguna baru dengan hak akses penuh terhadap basis data yang telah dibuat, sehingga memudahkan aplikasi dalam melakukan operasi baca-tulis data secara terintegrasi.

Gambar 6. Konfigurasi MariaDB.

Machine Learning API

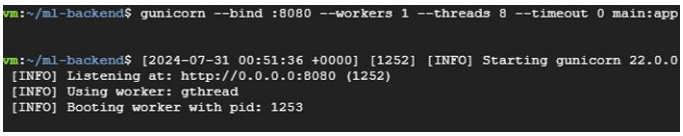
Konfigurasi Machine Learning API dimulai dengan instalasi paket pendukung seperti *git*, *python3*, *python3-venv*, dan *python3-pip* yang berfungsi menyiapkan lingkungan kerja berbasis Python.



Gambar 7. Instalasi Paket Penunjang Machine Learning API.

Setelah itu dibuat direktori khusus dan *virtual environment* untuk menjaga isolasi paket yang akan digunakan dalam proses pengembangan API. Tahap berikutnya adalah menginstal

paket-paket Python yang relevan, termasuk *yolov10*, *supervision*, dan *roboflow*, yang berperan penting dalam pemrosesan model deteksi gambar. Model *machine learning* yang telah dilatih sebelumnya kemudian diunggah ke server dan dipindahkan ke direktori kerja API agar dapat diakses oleh sistem. Untuk menjalankan API, digunakan *framework* Flask yang dikombinasikan dengan *gunicorn* sebagai server aplikasi, sehingga API dapat berjalan secara stabil dengan dukungan *multithreading*. Setelah API dijalankan, dilakukan pengujian awal untuk memastikan bahwa model dapat membaca dan mengklasifikasikan gambar sesuai dengan data latih.



Gambar 8. Menjalankan Machine Learning API.

Backend API

Konfigurasi Backend API dilakukan untuk menghubungkan aplikasi dengan layanan *machine learning* serta basis data sehingga alur kerja sistem dapat berjalan secara menyeluruh. Tahap ini dimulai dengan instalasi bahasa pemrograman Go (Golang) pada server, yang digunakan sebagai dasar pembangunan *backend service*.

Gambar 9. Instalasi Golang.

Setelah instalasi, dibuat direktori kerja khusus kemudian dilakukan *cloning* repositori *backend* dari proyek ke dalam server. Selanjutnya, *service account* yang sebelumnya telah dibuat pada Google Cloud diunggah ke dalam direktori dan diatur agar dapat digunakan untuk autentikasi akses ke layanan Google Cloud Storage. Pada tahap ini juga dibuat file *.env* yang berfungsi menyimpan variabel lingkungan, seperti kredensial akses dan konfigurasi API, sehingga *backend service* dapat berjalan dengan baik. Setelah seluruh persiapan selesai, Backend API dijalankan menggunakan command go run dan dilakukan penyesuaian konfigurasi sistem melalui visudo untuk memastikan path Golang dikenali secara otomatis. Selama proses berjalan, sistem akan mengunduh dependensi yang dibutuhkan, hingga akhirnya Backend API aktif dan dapat menerima permintaan dari aplikasi maupun berkomunikasi dengan Machine Learning API dan basis data.

```
go: downloading github.com/asaskevich/govalidator v0.0.0-20230301143203-a9d515a09cc2
go: downloading github.com/gabriel-vasile/mimetype v1.4.4
go: downloading golang.org/x/crypto v0.23.0
go: downloading cloud.google.com/go/storage v1.41.0
go: downloading gorm.io/driver/mysql v1.5.6
go: downloading cloud.google.com/go v0.112.2
go: downloading github.com/golang-jwt/jwt v3.2.2+incompatible
go: downloading github.com/google/uuid v1.6.0
go: downloading github.com/mattn/go-colorable v0.1.13
go: downloading github.com/mattn/go-isatty v0.0.20
go: downloading github.com/mattn/go-runewidth v0.0.15
go: downloading github.com/valyala/bytebufferpool v1.0.0
```

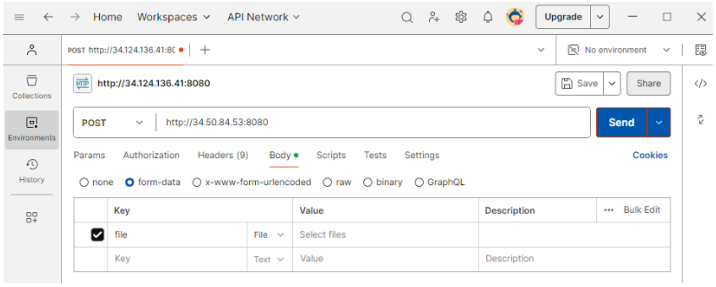
Gambar 10. Menjalankan Backend API.

Uji Coba

Pada tahap uji coba, dilakukan uji terhadap seluruh komponen aplikasi yang telah di-*deploy* dipastikan dapat berjalan sesuai dengan tujuan. Setelah proses konfigurasi infrastruktur selesai dilakukan, tahap pengujian difokuskan pada dua komponen utama, yaitu *Machine Learning API* dan *Backend API*.

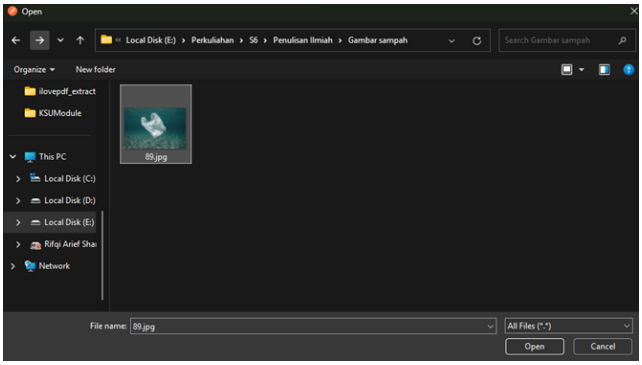
Machine Learning API

Pengujian Machine Learning API dilakukan menggunakan aplikasi Postman untuk memastikan layanan mampu memproses permintaan sesuai rancangan. Proses pengujian dimulai dengan mengatur metode HTTP dari *GET* menjadi *POST*, lalu memasukkan alamat IP mesin virtual yang telah dikonfigurasi beserta port yang digunakan.



Gambar 11. Aplikasi Postman.

Selanjutnya, ditambahkan file gambar yang akan diuji dengan memilih berkas dari perangkat lokal, kemudian permintaan dikirimkan ke API.

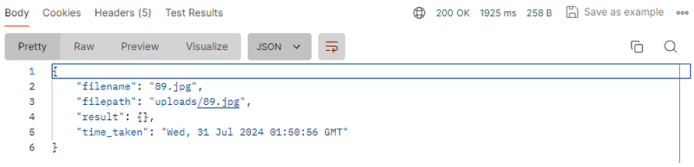


Gambar 12. Pencarian Gambar.

Hasil pengujian menunjukkan dua kemungkinan respon, yaitu tidak terdeteksinya objek apabila gambar yang dikirimkan tidak sesuai dengan data latih, atau sebaliknya menampilkan

hasil klasifikasi objek beserta tingkat kepercayaan (confidence score) ketika gambar yang dikirimkan sesuai dengan model yang telah dilatih.

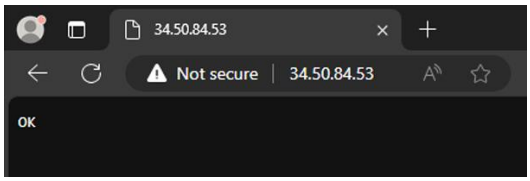
Gambar 13. Respon bila terdeteksi.



Gambar 14. Respon bila tidak terdeteksi.

Backend API

Pengujian Backend API dilakukan melalui peramban (*browser*) untuk memastikan layanan backend dapat berjalan dan merespon permintaan dengan benar. Proses uji coba dilakukan dengan mengakses alamat IP mesin virtual yang telah dikonfigurasi sebelumnya. Hasil pengujian menunjukkan bahwa ketika Backend API berjalan dengan baik, sistem memberikan respon berupa pesan “OK” sebagai indikator keberhasilan.



Gambar 15. Pengujian Backend API.

Respon sederhana ini sudah cukup untuk membuktikan bahwa layanan backend aktif, dapat diakses, dan siap menerima komunikasi dari komponen lain, seperti Machine Learning API maupun aplikasi pengguna. Dengan demikian, hasil uji coba ini memastikan bahwa integrasi antar layanan dalam sistem deployment dapat berjalan secara lancar dan sesuai dengan tujuan penelitian.

5. KESIMPULAN DAN SARAN

Hasil penelitian menunjukkan bahwa Google Cloud Platform (GCP) berhasil diimplementasikan sebagai infrastruktur aplikasi deteksi sampah dengan memanfaatkan layanan utama seperti Google Compute Engine (GCE), Google Cloud Storage (GCS), dan Google Virtual Private Cloud (VPC) yang mampu mendukung proses deployment secara

efektif. Machine Learning API dan Backend API dapat dijalankan dengan baik, terbukti dari hasil pengujian yang sesuai harapan, termasuk kemampuan API dalam mengidentifikasi objek sampah dari gambar pengguna. Namun, penelitian ini memiliki keterbatasan karena belum didukung fitur autoscaling, sehingga disarankan pada penelitian selanjutnya untuk memanfaatkan Google App Engine agar sistem lebih adaptif terhadap beban kerja yang dinamis dan mendukung pengembangan yang lebih optimal.

DAFTAR REFERENSI

Ahmad, A. (2017). *Mengenal artificial intelligence, machine learning, neural network, dan deep learning.* DocPlayer. https://docplayer.info/62490785-Mengenal-artificialintelligence-machine-learning-neural-networkdan-deep-learning.html

Alazzawi, A., Yas, Q. M., & Rahmatullah, B. (2023). A comprehensive review of software development life cycle methodologies: Pros, cons, and future directions. *Iraqi Journal for Computer Science and Mathematics, 4*(4), 173–190. https://doi.org/10.52866/ijcsm.2023.04.04.014

Albar, M. H. (2019, December 18). *Pengenalan bahasa pemrograman Go (Golang).* Medium. https://medium.com/developer-student-club-universitasbrawijaya/pengenalan-bahasa-pemrograman-go-golang-7af58e1f3932

Amazon Web Services. (2025). *Six advantages of cloud computing.* AWS Whitepaper.

Anonim. (2024). *What is Python used for? A beginner's guide.* Coursera. https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

Bisong, E. (2019). Google Compute Engine (GCE). In *Building machine learning and deep learning models on Google cloud platform* (pp. 133–159). Apress. https://doi.org/10.1007/978-1-4842-4470-8_5

Christiono, K., & Sama, H. (2020). Studi komparasi database management system antara MariaDB dan PostgreSQL terhadap efisiensi penggunaan sumber daya komputer. *Cognitive Artificial Intelligence Research (CABSST), 1*(1), 1–7. Universitas Internasional Batam.

CompanionLink. (2025, March 19). *The role of cloud computing in AI and machine learning.* CompanionLink.

Effendi, R. N. (2021). Perancangan web aplikasi analisis sentimen media sosial Twitter dengan metode Valence Aware Dictionary and Sentiment Reasoner (VADER) menggunakan PHP dan MySQL pada Pemerintah Kota Bekasi. *Jurnal Ilmiah KOMPUTASI, 20*(1), 1–10. https://doi.org/10.32409/jikstik.20.1.369

IBM. (2024). *What are the benefits of cloud computing?* IBM.

Irawan, A. L. H. (2024). Implementation Google Cloud Platform as data storage in industry. *Riwayat: Educational Journal of History and Humanities, 7*(2), 462–471. https://doi.org/10.24815/jr.v7i2.37699

Kementerian Lingkungan Hidup dan Kehutanan (KLHK). (2024). *Sistem Informasi Pengelolaan Sampah Nasional (SIPSN).* KLHK.

Kurniawan, N. (2020, June 4). *Postman.* Medium. https://medium.com/@novancimol12/postman-4f181d625fe1

Noviana, R. (2022). Pembuatan aplikasi penjualan berbasis web Monja Store menggunakan PHP dan MySQL. *Jurnal Teknik dan Science, 1*(2), 112–124. https://doi.org/10.56127/jts.v1i2.128

Noviana, R., & Rasal, I. (2023). Penerapan algoritma Naive Bayes dan SVM untuk analisis sentimen boy band BTS pada media sosial Twitter. *Jurnal Teknik dan Science, 2*(2), 51–60. https://doi.org/10.56127/jts.v2i2.791

Pargaonkar, S. (2023). A comprehensive research analysis of software development life cycle (SDLC) Agile & Waterfall model advantages, disadvantages, and application suitability in software quality engineering. *International Journal of Scientific and Research Publications, 13*(8), 14015. https://doi.org/10.29322/IJSRP.13.08.2023.p14015

Pratama, R. A., & Nugroho, H. (2022). Implementasi cloud computing untuk pengembangan aplikasi mobile di Indonesia. *Jurnal RESTI, 6*(4), 710–718. https://doi.org/10.29207/resti.v6i4.1234

Setiany, A. P., Noviyanto, D., Irfansyahfalah, M., Aisah, S., Saifudin, A., & Kusyadi, I. (2021). Penggunaan metode System Development Life Cycle (SDLC) dalam analisis dan perancangan sistem informasi penerimaan kas sekolah. *Jurnal Teknologi Sistem Informasi dan Aplikasi, 4*(3), 179–186.

TechRadar. (2025, August 8). *Why enterprises can't afford to ignore cloud optimization in 2025.* TechRadar.