

Penggunaan *Google Cloud Platform* Untuk Membuat *Backend Aplikasi EcoScan : Waste Detection*

Abiyyu Rafi¹, Wiga Maulana Baihaqi.²

^{1,2}Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto, Purwokerto, Indonesia.¹
email : abiyyurafi12@gmail.com¹, wiga@amikompurwokerto.ac.id²

Info Artikel

Sejarah artikel:

Diterima 13-05-2025

Revisi 20-06-2025

Diterima 13-07-2025

Kata kunci:

Google Cloud Platform

Backend Aplikasi

Pengolahan Sampah

ABSTRAK

Dalam perkembangan teknologi yang dapat mempermudah segala aktifitas yang dijalankan manusia khususnya terhadap barang-barang elektronik seperti Handphone, TV, Komputer. Dalam perancangan dan pembuatan aplikasi ini menerapkan teknologi *Cloud Computing*. Salah satu penyedia layanan *Cloud Computing* adalah *Google Cloud Platform* (GCP). Proyek *Ecoscan* bertujuan untuk meningkatkan efisiensi dalam pengenalan dan pemisahan jenis limbah, sehingga memfasilitasi proses daur ulang dan pengelolaan limbah secara keseluruhan. GCP dipilih sebagai *platform backend* untuk menyediakan infrastruktur yang skalabel, aman, dan handal bagi aplikasi. Kami merancang dan mengimplementasikan sistem *backend* yang memanfaatkan layanan GCP, seperti *Cloud Storage* untuk penyimpanan data, *Cloud Vision API* untuk analisis gambar, dan *Cloud Functions* untuk pemrosesan tanpa server.

Penulis yang sesuai:

Abiyyu Rafi

Program Studi Teknologi Informasi Fakultas Ilmu Komputer Universitas Amikom Purwokerto

Email: abiyyurafi12@gmail.com

1. PENDAHULUAN

Saat ini yang terjadi dalam bidang industri di Indonesia adalah memanfaatkan teknologi informasi sebaik mungkin, dimana banyak aspek kehidupan manusia telah terbantu atau tergantikan dengan hadirnya sebuah teknologi digital untuk mencapai efisiensi maksimal dalam mengembangkan model bisnis baru berbasis digital. Salah satu teknologi yang sedang ramai dibicarakan adalah *Cloud Computing*. *Cloud Computing* merupakan model *client-server*, arsitektur *client-server Cloud Computing* memungkinkan pengguna untuk mengakses sumber daya termasuk server, penyimpanan, jaringan, dan perangkat lunak dari mana saja kapan saja sebagai sebuah layanan[1]. *Cloud Computing* dapat diartikan juga sebagai sebuah mekanisme, dimana sekumpulan sumber teknologi informasi yang saling terhubung dan nyaris tanpa batas, baik itu infrastruktur maupun aplikasi yang dimiliki dan dikelola sepenuhnya oleh pihak ketiga sehingga memungkinkan customer untuk menggunakan *resource* tersebut secara *on demand* melalui *network* baik yang sifatnya jaringan *public*[2][3].

Indonesia adalah salah satu negara yang mempunyai masalah terhadap lingkungan khususnya masalah sampah. Indonesia masih mempunyai banyak masyarakat yang belum memiliki kesadaran membuang sampah pada tempatnya[4][5], hal tersebut dapat membuat sampah berada dimana-mana dan dapat menyebabkan berbagai hal negatif. Manusia setiap harinya menghasilkan limbah, baik limbah rumah tangga maupun industri yang memiliki berbagai jenis dan bentuk. Sampah dapat menjadi masalah karena mengganggu kesehatan manusia, menyebabkan bau busuk dan polusi udara. Kesadaran membuang sampah ditempatnya saat ini dianggap sangat kurang[6].

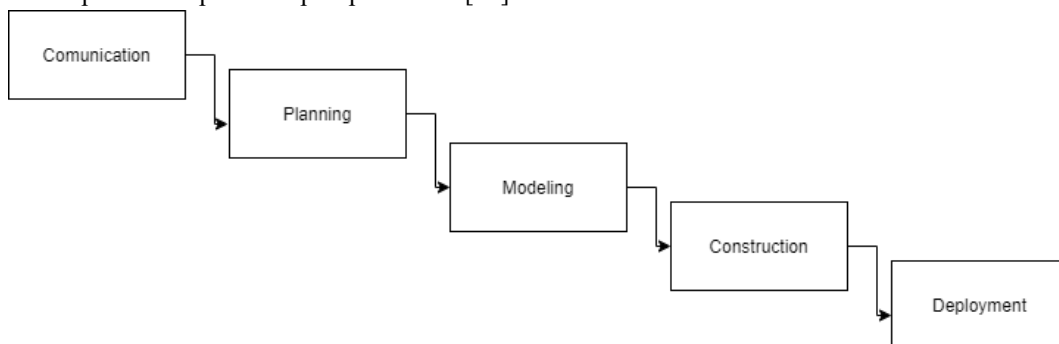
Dengan perkembangan teknologi saat ini, penggunaan *Cloud Computing* merupakan salah satu alternatif yang dapat digunakan sebagai pengganti dari kebutuhan perangkat keras dalam menjalankan aplikasi secara *realtime*. Implementasi *Cloud Computing* dilakukan dengan tujuan untuk mengurangi biaya dari pembuatan aplikasi, sehingga diharapkan aplikasi yang dibuat dapat berjalan secara optimal tanpa memerlukan biaya yang mahal[7].

Pemanfaatan backend berbasis cloud dalam aplikasi pengelolaan sampah juga telah banyak diterapkan pada penelitian sebelumnya. Prayoga dan Yudistira [8] menunjukkan bahwa penggunaan layanan cloud sebagai backend pada sistem monitoring sampah pintar mampu menyederhanakan pengelolaan data dan meningkatkan ketersediaan layanan. Selain itu, Pamungkas dan Rasal [9] membuktikan bahwa implementasi Google Cloud Platform sebagai infrastruktur backend aplikasi deteksi sampah dapat mendukung deployment aplikasi secara scalable dan fleksibel tanpa memerlukan pengelolaan server secara mandiri.

Dengan adanya permasalahan tersebut dan dengan berkembangnya teknologi yang ada untuk saat ini, penelitian ini memanfaatkan *platform Google Cloud Platform* ini untuk menjadi tempat pembuatan *backend* untuk aplikasi yang sedang kamu rancang yaitu aplikasi *Ecoscan*. Dengan adanya aplikasi *Ecoscan* ini pengguna dapat melakukan scanning terhadap sampah yang mereka miliki dan juga dapat memilah sampah sesuai dengan kategori yang telah ditetapkan juga oleh kementerian Lingkungan Hidup. Dengan itu pengguna dapat menjaga lingkungan sekitar mereka dengan cara tersebut.

2. METODE

Tahapan pengerjaan penelitian ini menggunakan metode *waterfall*. Metode *waterfall* adalah pendekatan sistematis dan berurutan yang dimulai dengan analisis sistem dan kebutuhan pengguna dan berkembang melalui tahap perencanaan seperti desain atau desain sistem dan basis data, pengkodean, pengujian, dan pemeliharaan sistem [10]. Metode Waterfall merupakan salah satu model dalam *Software Development Life Cycle* (SDLC) yang paling banyak digunakan pada pengembangan perangkat lunak yang memiliki kebutuhan yang sudah jelas dan stabil [11]. Dan dalam pengembangan backend aplikasi berbasis cloud karena memudahkan proses perancangan arsitektur sistem dan meminimalkan perubahan pada tahap implementasi[12].



Gambar 1. Metode *Waterfall*

2.1 Communication

Dalam metode Waterfall, tahap komunikasi atau analisis kebutuhan sangat penting karena membantu menentukan kebutuhan sistem secara menyeluruh sebelum masuk ke tahap perancangan. Studi sebelumnya menunjukkan bahwa keberhasilan desain arsitektur backend dan implementasi layanan cloud dipengaruhi langsung oleh proses pengumpulan kebutuhan yang baik[13]. Tahapan ini terdiri dari proses identifikasi masalah dengan melakukan studi untuk mengetahui kebutuhan layanan sistem dari kendala, kapabilitas, teknologi melalui brainstorming dan konsultasi dengan mentor. Tahapan ini dilakukan secara langsung melalui virtual meeting. Setelahnya menentukan fungsional dan nonfungsional pada sistem. Hasil dari tahap ini berupa rancangan arsitektur *Backend* yang memanfaatkan teknologi *Cloud Computing* pada *Platform as a Service* (Paas) *Google Cloud Platform*[10].

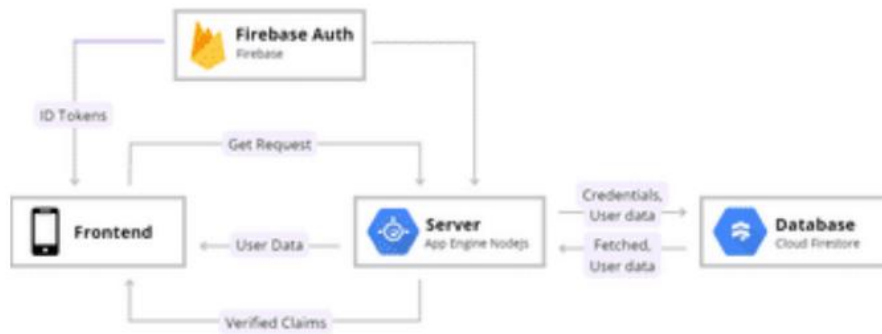
2.2 Planning

Tahapan perencanaan ini terdiri dari proses *scoping*, *deliverables*, *scheduling*, *local Deployment* dan *Deployment Google Cloud Platform*. Pada tahap ini juga kami mengkomunikasikan kepada tim dari divisi lain yang terkait mengenai *timeline* yang akan kita kerjakan untuk menyelesaikan aplikasi. Sehingga dari masing masing divisi bisa tau bagaimana atau hal apa saja yang perlu dikerjakan guna untuk menyelesaikan aplikasi *Ecoscan* ini.

2.3 Modeling

Perancangan arsitektur backend berbasis RESTful API merupakan pendekatan yang umum digunakan dalam pengembangan aplikasi modern berbasis cloud [14]. Tahap ini terdiri dari analisis dan desain sistem. Dilakukan

analisa terhadap desain arsitektur pada layanan *Google Cloud Platform*, hasil dari proses ini adalah sebuah desain rancangan arsitektur pada layanan *Google Cloud Platform* yang dapat dilihat pada gambar 2.



Gambar 2. Arsitektur *Back-end* aplikasi *Ecoscan*

Pada gambar 2 menunjukkan perancangan arsitektur dari *Backend* aplikasi *Ecoscan*. Secara garis besar, *Frontend* akan mengirimkan *request* melalui server App Engine Nodejs kepada database *Cloud Firestore*. Data yang masuk ke database akan dikelola oleh tim *Back-end*, dan setiap data yang ditampilkan ke user atau *frontend* itu tersimpan didatabase yang sudah dibuat.

2.4 Construction

Tahapan ini terdiri dari penulisan kode program dan pengujian sistem. Tahapan penulisan kode program dilakukan dengan mengimplementasikan *RestFul API* menggunakan Tools Visual Studio Code dan *testing* kode program secara *local* dengan Postman. Framework yang digunakan pada *Backend* adalah NodeJS, Express, dan Nodemon. Penggunaan Node.js dan Express sebagai framework backend telah banyak digunakan pada penelitian sebelumnya karena memiliki performa yang baik dalam menangani request secara asynchronous serta mudah diintegrasikan dengan layanan cloud. Pengujian API menggunakan Postman juga merupakan metode yang umum digunakan untuk memastikan fungsi RESTful API berjalan sesuai dengan kebutuhan sistem[11].

2.5 Deployment

Pada tahap implementasi *Cloud Computing* dilakukan konfigurasi requirements yang di implementasikan pada *environment Google App Engine*. Selain itu dilakukan pemeliharaan sistem dengan memonitor trafik menggunakan fitur *Google Cloud Platform* yaitu *Log Explorer*[10].

3. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah aplikasi bernama *Ecoscan* yang menyediakan layanan edukasi yang diharapkan dapat meningkatkan kedisiplinan pengguna. Pada bagian ini, akan dijabarkan tentang sistematika dari jalannya program aplikasi. Pengguna terlebih dahulu melakukan registrasi akun pada aplikasi *Ecoscan*, kemudian pelanggan memilih kategori layanan pada menu utama. Aplikasi *Ecoscan* memiliki beberapa kategori layanan yaitu fitur *scan*, fitur *scan* ini digunakan supaya user dapat mengidentifikasi sampah yang mereka miliki itu organik atau an organik. Dengan adanya fitur ini diharapkan pengguna dapat membedakan dan mengolah sampah sesuai dengan jenis sampah tersebut. Selanjutnya ada fitur rekomendasi, fitur ini digunakan untuk pengguna supaya dapat melihat contoh langsung pengolahan sampah tersebut, dan di fitur ini juga terdapat artikel yang dapat dibaca oleh pengguna dan dapat dijadikan sebagai bahan bacaan yang dapat menambah wawasan pengguna mengenai bagaimana pengolahan sampah yang baik dan benar sesuai dengan jenisnya masing masing.

3.1. Pembuatan Login dan Register

Pembuatan sistem login dan register yang dilakukan oleh tim dari *Back-end*, ini dibuat guna memudahkan nantinya bagian *frontend* menghubungkan data yang masuk dari user dan langsung masuk ke database yang dibuat oleh *backend*, sehingga API ini dibuat guna memudahkan dan menyambungkan hal tersebut.

```
const jwt = require('jsonwebtoken');
const User = require('../models/userModel');
const dotenv = require('dotenv');

dotenv.config();

const register = (req, res) => {
  const { fullname, email, password, city } = req.body;

  User.createUser(fullname, email, password, city, (err, result) => {
    if (err) {
      if (err.message === 'Email is already taken') {
        return res.status(400).json({ success: false, message: 'Registration failed. This email is already taken' });
      }
      return res.status(500).json({ success: false, message: 'Oops! Something went wrong on our end. Please try again' });
    }
    return res.status(201).json({ success: true, message: 'Registration successful. Welcome aboard!' });
  });
};
```

Gambar 3. API untuk menu register

```
const login = (req, res, next) => {
  const { email, password } = req.body;

  User.getUserByEmailAndPassword(email, password, (err, user) => {
    if (err) {
      return res.status(500).json({ success: false, message: 'Oops! Something went wrong on our end. Please try again' });
    }
    if (!user) {
      return res.status(401).json({ success: false, message: 'Login failed. Invalid email or password. Please try again' });
    }

    const token = jwt.sign({ id: user.id, email: user.email }, process.env.JWT_SECRET, { expiresIn: '7d' });

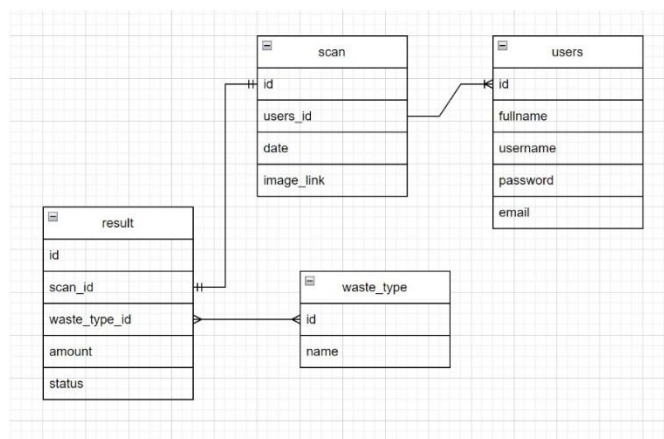
    return res.json({
      success: true,
      message: 'Login successful. Enjoy your day!',
      userDetails: {
        userId: user.id,
        fullname: user.fullname,
        email: user.email,
        token: token
      }
    });
  });
};
```

Gambar 4. API untuk menu login

Setelah API tersebut ada, kita terapkan kedalam tampilan yang sudah dibuat oleh tim *frontend* untuk fitur login dan register ini di dalam UI aplikasi *Ecoscan*.

3.2. Struktural Database

Pembuatan database ini bertujuan untuk menyimpan data data yang diperlukan oleh *front-end* untuk menjalankan fitur fitur yang ada diaplikasi *Ecoscan*. Selain itu database disini juga berguna sebagai tempat penyimpanan sekaligus pengolahan data-data user yang masuk, seperti data login, data register dan lainnya.

Gambar 5. Relasi database *Ecoscan*

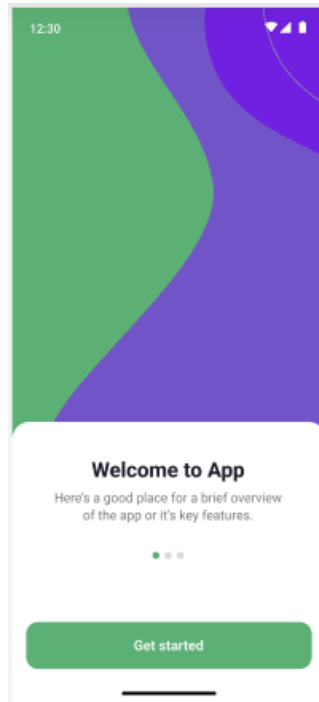
Dengan adanya relasi database tersebut, akan memudahkan tim *backend* untuk membuat database, karena jelas tertera tujuan dan relasi antar fiturnya kemana saja.

3.3. Tampilan Aplikasi

Berikut ini adalah hasil setelah tahap perancangan dari antarmuka aplikasi. Aplikasi *Ecoscan* memiliki, halaman login, register, halaman rekomendasi, halaman *profile*, dan juga fitur *scan*.

a. Tampilan awal aplikasi

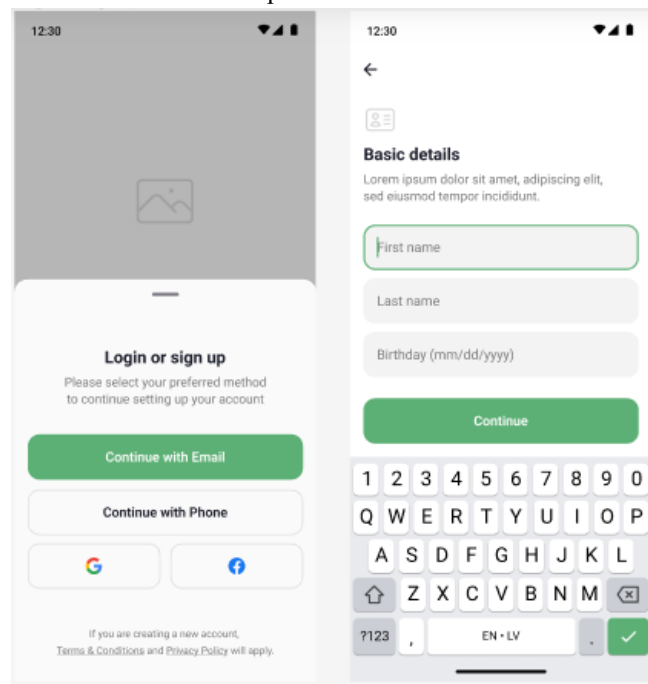
Tampilan awal ini dibuat sebagai tampilan selamat datang kepada pengguna saat memasuki aplikasi *Ecoscan*, dengan adanya tampilan awal ini dapat lebih mem-branding aplikasi *Ecoscan* ini sendiri.



Gambar 6. Tampilan awal aplikasi *Ecoscan*

b. Tampilan Login & Register

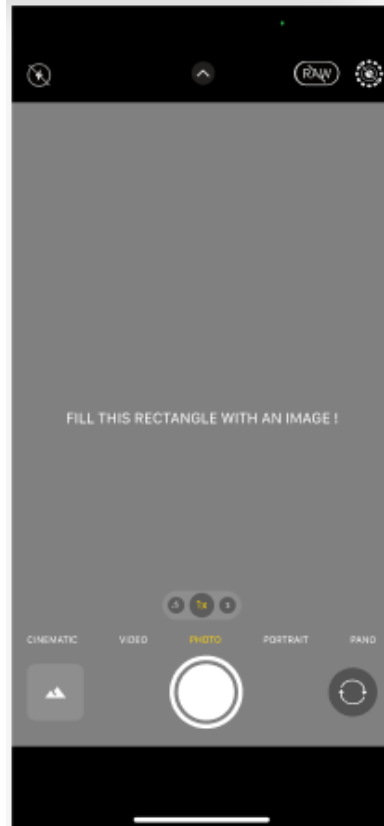
Tampilan Login ini digunakan untuk masuk menu utama halaman aplikasi *Ecoscan*, dengan melakukan login pengguna dapat mengakses fitur-fitur yang ada di aplikasi *Ecoscan*. Sedangkan Register digunakan user untuk mendaftarkan dirinya untuk memiliki *account* di aplikasi *Ecoscan* ini.



Gambar 7. Login & Register

c. Tampilan Fitur *Scan*

Fitur *Scan* ini merupakan fitur utama dari aplikasi *Ecoscan* ini, dengan adanya fitur ini dapat membantu memudahkan pengguna dalam mengelola dan mengelompokkan sampah yang mereka miliki. Dan mengklasifikasikannya menjadi barang organik ataupun anorganik.



Gambar 8. Fitur *Scan* aplikasi *Ecoscan*

4. KESIMPULAN

Cloud Computing merupakan model pemberian layanan teknologi informasi untuk pengguna secara fleksibel dengan server virtual, skalabilitas besar, dan manajemen layanan. Layanan teknologi informasi ini dapat digunakan oleh organisasi untuk mempermudah menjalankan proses bisnisnya, dengan menggunakan *Cloud Computing*, kita dapat mengakses data atau program dimana saja, kapan saja, dan dengan perangkat apapun. Adapun salah satu penyedia layanan dari *Cloud Computing* ini adalah *Google Cloud Platform* (GCP). Infrastruktur server yang akan dibangun pada penelitian ini menggunakan produk *Compute Engine*, *Cloud SQL*, *Cloud Storage*, dan *Container Registry* yang ada pada Teknologi *Google Cloud Platform* (GCP) dan menerapkan fitur *autohealing*, *multiple zones*, *load balancing*, *autoscaling*, *automatic updating*, dan *failover* sebagai metode untuk menjaga [15].

Dengan memanfaatkan *Google Cloud Platform* dapat memudahkan dalam pembuatan *Back-end* aplikasi *Ecoscan* dan dapat mudah untuk saling bekerjasama, dan memiliki *cost* biaya yang lebih terjangkau. Karena penggunaan server secara virtual dengan sistem bayar ketika digunakan. *Google Cloud Platform* merupakan salah satu PaaS yang tepat dan sangat disarankan untuk digunakan jika ingin *men-develope* sebuah aplikasi, karena fitur-fitur yang mudah dipahami, dan memudahkan untuk saling bekerja sama tanpa harus bertatap muka antar tim pengembang.

REFERENSI

- [1] D. Kalaivani, "Cloud Computing : A Study on Architecture , Types , Benefits and Challenges," vol. 16, no. 3, pp. 1–7.
- [2] D. L. F. Annisa and R. Andryani, "Penerapan Cloud Computing Dalam Aplikasi Panggil Teknisi Berbasis Android Menggunakan Google Cloud Platform," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 6, no. 2, pp. 1292–1300, 2022.



- [3] "Cloud computing".
- [4] L. J. Lingga, M. Yuana, N. A. Sari, H. N. Syahida, and C. S. Shahron, "Sampah di Indonesia : Tantangan dan Solusi Menuju Perubahan Positif," vol. 4, pp. 12235–12247, 2024.
- [5] E. Sembiring *et al.*, "Improving household waste management in Indonesia : A mixed-methods approach for waste Sorting," *Clean. Waste Syst.*, vol. 9, no. September, p. 100185, 2024, doi: 10.1016/j.clwas.2024.100185.
- [6] S. Cahyati and Y. Ramdhani, "Aplikasi android monitoring tempat sampah pintar berbasis internet of things," *eProsiding Tek. Inform.*, vol. 2, no. 1, pp. 112–121, 2021.
- [7] M. R. Alfarizi and H. P. Permatasari, "Implementasi Cloud Computing Terhadap Aplikasi Pemesanan Parkir Di Kota Jakarta," *UG J.*, pp. 37–49, 2022.
- [8] M. W. Prayoga, B. Gede, K. Yudistira, P. Studi, I. Komputer, and U. P. Ganesha, "SISTEM MONITORING DAN PENGELOLAAN SAMPAH," vol. 13, no. 3.
- [9] R. A. Pamungkas, I. Rasal, F. T. Industri, and U. Gunadarma, "Deployment Aplikasi Deteksi Sampah pada Google Cloud penanganan permasalahan sampah . Teknologi ini memungkinkan dikembangkanya aplikasi 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 memberikan keunggulan dari sisi ketersediaan layanan , kecepatan pemrosesan , serta keandalan Penelitian ini bertujuan untuk mengimplementasikan aplikasi deteksi sampah berbasis penelitian diarahkan pada tahap deployment aplikasi sehingga dapat diakses oleh pengguna teknologi cloud untuk solusi pengelolaan sampah di Indonesia , sekaligus memperkuat upaya Cloud Computing pengaksesan layanan seperti penyimpanan , server , dan aplikasi melalui internet tanpa harus," vol. 3, no. September, 2025.
- [10] P. Aplikasi, G. Berbasis, and C. Computing, "Implementasi Platform As A Service (PAAS) Pada Aplikasi Getfix Berbasis Cloud Computing," *J. Sains dan Inform.*, vol. 8, no. 2, pp. 86–95, 2022, doi: 10.22216/jsi.v8i2.1653.
- [11] S. Informasi, K. Kota, F. Ilmu, and T. Informasi, "Perancangan dan Implementasi RESTful API untuk Aplikasi Mobile Pembelajaran Flora dan Fauna pada Google Cloud Platform," vol. 4, no. 1, pp. 58–69, 2024, doi: 10.54259/satesi.v4i1.2850.
- [12] M. M. Wijaya, H. Zuchlizal, N. Rohiim, and R. J. Ramadani, "Pengembangan Sistem Pengelolaan Sampah Pintar Berbasis IoT untuk Efisiensi Pengumpulan Sampah di Kampus," vol. 4, pp. 66–72, 2025.
- [13] R. Abdi and S. Wulandari, "Rancang Bangun Sistem Informasi Manajemen Perpustakaan Berbasis Cloud Server," vol. 1, no. 3, pp. 93–99, 2025.
- [14] A. Maulana, N. Riady, P. Paniran, and I. M. B. Suksmadana, "Perancangan Backend Api Berbasis Rest -Api pada Aplikasi Rekomendasi Resep Makanan (Rest-API Based Backend Api Design In Food Recipe Recommendation Application)," vol. 2, no. 3, 2024.
- [15] G. Nopi, Ramsari, & Arif, "472-1953-1-Pb," *Implementasi Infrastruktur Serv. Berbas. Cloud Comput. Untuk Web Serv. Berbas. Teknol. Google Cloud Platf.*, vol. VII, 2022, doi: 10.28989/senatik.v7i1.472.