

# Integrasi Cloud Computing dalam Pemantauan Pemberian Makanan Kucing Otomatis Berbasis Deep Learning

1<sup>st</sup> Muhammad Raihan Butar-Butar  
*Fakultas Teknik Elektro  
 Universitas Telkom  
 Bandung, Indonesia*  
 mraihan@student.telkomuniversity.ac.id

2<sup>nd</sup> Meta Kallista  
*Fakultas Teknik Elektro  
 Universitas Telkom  
 Bandung, Indonesia*  
 metakallista@telkomuniversity.ac.id

3<sup>rd</sup> Rifqi Muhammad Fikri  
*Fakultas Teknik Elektro  
 Universitas Telkom  
 Bandung, Indonesia*  
 rifqmff@telkomuniversity.ac.id

**Abstrak** — Kucing merupakan salah satu hewan peliharaan yang populer di Indonesia dan memberikan manfaat emosional bagi pemiliknya. Namun, kucing domestik memerlukan perawatan rutin, termasuk pemberian makan yang konsisten, yang sering terkendala jika pemilik memiliki kesibukan di luar rumah. Penelitian ini bertujuan mengembangkan Smart Pet Feeder, sebuah sistem berbasis Internet of Things (IoT) yang mampu memberikan makanan secara otomatis, terjadwal, dan dapat dipantau dari jarak jauh. Metode pengembangan mencakup perancangan backend REST API menggunakan FastAPI yang diintegrasikan dengan layanan Google Cloud Platform, meliputi Cloud Run untuk *deployment*, Cloud Storage untuk penyimpanan data foto kucing dan model *deep learning*, serta Compute Engine untuk pemrosesan *deep learning*. API dibangun untuk mengelola autentikasi pengguna, jadwal pemberian makan, profil kucing, pemantauan real-time, dan integrasi model deteksi kucing. Pengujian dilakukan untuk mengevaluasi kinerja API, keandalan penyimpanan data, dan akurasi integrasi model. Hasil pengujian menunjukkan backend berbasis cloud ini responsif, skalabel, dan dapat diakses dari berbagai perangkat, meskipun memerlukan koneksi internet yang stabil. Dengan adanya sistem ini, proses pemberian makan kucing dapat dilakukan secara lebih efisien, terjadwal, dan dapat dipantau kapan saja.

**Kata kunci**— smart pet feeder, internet of things, google cloud platform, cloud run, cloud storage.

## I. PENDAHULUAN

Kucing domestik merupakan hewan peliharaan populer di Indonesia karena perawatannya mudah, sifatnya mandiri, dan mampu memberikan dukungan emosional bagi pemiliknya, seperti menurunkan stres, kecemasan, dan tekanan darah melalui interaksi sederhana maupun suara dengkuran yang bersifat terapeutik. Namun, pemilik dengan jadwal padat atau sering bepergian sering menghadapi tantangan dalam memastikan kucing mendapatkan makanan, air, serta pemantauan kesehatan dan perilaku secara teratur. Perubahan pola makan atau perilaku sosial dapat menjadi indikator awal masalah kesehatan, tetapi pengamatan manual sering kurang akurat dan memerlukan teknologi pendukung untuk mendeteksi perubahan halus. Ketidakmampuan memantau secara konsisten dapat menyebabkan keterlambatan penanganan yang berisiko memperburuk kondisi kesehatan kucing[1]. Perkembangan teknologi Internet of Things (IoT) dan cloud computing telah mendorong inovasi pada perangkat otomatis untuk pemeliharaan hewan peliharaan, termasuk smart feeder yang mampu memberikan makanan secara terjadwal. Beberapa

penelitian dan produk komersial telah mengimplementasikan fitur kontrol jarak jauh melalui aplikasi mobile, pemantauan real-time dengan kamera, serta penggunaan machine learning untuk deteksi hewan. Pemanfaatan *deep learning* berbasis YOLO telah banyak digunakan untuk mengenali hewan peliharaan secara visual, memungkinkan sistem membatasi akses makanan hanya kepada hewan tertentu. Dari sisi infrastruktur, layanan cloud seperti Google Cloud Platform (GCP), AWS, dan Azure menyediakan kemampuan penyimpanan data, eksekusi model, dan *deployment* API yang mendukung skalabilitas. Penelitian ini bertujuan untuk mengembangkan sistem backend terintegrasi berbasis REST API yang mampu mengelola pemberian makanan kucing secara otomatis, terjadwal, dan dapat dipantau jarak jauh. Sistem dibangun menggunakan FastAPI dan di-*deploy* pada layanan Google Cloud Platform, memanfaatkan Cloud Run untuk *deployment*, Cloud Storage untuk penyimpanan data dan model, serta Compute Engine untuk pemrosesan *deep learning*. API yang dikembangkan dirancang untuk mendukung autentikasi pengguna, manajemen jadwal makan, pengelolaan profil kucing, pemantauan real-time, serta integrasi model deteksi kucing, sehingga diharapkan dapat menghasilkan sistem yang responsif, skalabel, dan mudah diintegrasikan dengan aplikasi mobile maupun perangkat IoT.

## II. KAJIAN TEORI

### A. Google Cloud Platform

Google Cloud Platform (GCP) merupakan platform komputasi awan yang menyediakan berbagai layanan modular mulai dari infrastruktur (IaaS), platform (PaaS), hingga layanan serverless, mencakup komputasi, penyimpanan, dan analitik data. GCP dihargai karena kemampuannya dalam menyediakan sumber daya komputasi yang dapat diskalakan, penyimpanan yang andal, dukungan pemrosesan data berskala besar, serta alat analitik yang kuat[2].

### B. Google Cloud Run

Cloud Run adalah layanan Platform as a Service (PaaS) berbasis *serverless* yang memungkinkan pengembang menjalankan aplikasi dalam bentuk *container* tanpa perlu melakukan pengelolaan server atau infrastruktur secara langsung. Melalui layanan ini, pengembang cukup mendaftarkan *container image*, sementara Google Cloud akan secara otomatis menangani proses *provisioning*, *load balancing*, dan *auto-scaling* sesuai kebutuhan, termasuk menghentikan instans saat tidak ada lalu lintas (*scale-to-zero*). Layanan ini juga terintegrasi dengan Cloud Logging dan Cloud Monitoring, serta mendukung berbagai bahasa pemrograman dan *framework*, sehingga mempermudah proses penerapan, pemantauan, dan *debugging* aplikasi secara real-time[3].

### C. Google Cloud Storage

Google Cloud Storage (GCS) adalah layanan penyimpanan objek (object storage service) dari Google Cloud yang dirancang untuk menyimpan dan mengelola data dalam jumlah besar secara aman dan terukur. GCS menawarkan berbagai kelas penyimpanan seperti Standard, Nearline, Coldline, dan Archive yang dapat dipilih sesuai kebutuhan akses data. Layanan ini mendukung integrasi dengan berbagai layanan Google Cloud lainnya, menyediakan fitur *versioning*, *lifecycle management*, serta enkripsi otomatis untuk keamanan data. Selain itu, GCS dapat diakses melalui antarmuka web, *command-line interface*, maupun API, sehingga memudahkan pengembang dalam mengunggah, mengunduh, dan mengelola data dari berbagai platform[4].

### D. Google Cloud Virtual Machine Instance

Google Cloud Virtual Machine melalui layanan Google Compute Engine (GCE) memungkinkan pengguna membuat dan menjalankan mesin virtual dengan konfigurasi sistem operasi, CPU, memori, dan penyimpanan yang fleksibel. VM ini dapat digunakan untuk berbagai kebutuhan, seperti menjalankan aplikasi, pemrosesan data, atau hosting server, dengan dukungan fitur seperti live migration, automatic restart, dan integrasi layanan Google Cloud untuk meningkatkan ketersediaan dan keandalan[5].

### E. Cloud Firestore

Cloud Firestore adalah layanan basis data NoSQL berbasis dokumen dari Google Firebase yang memungkinkan penyimpanan, sinkronisasi, dan pengambilan data secara real-time. Data disimpan dalam koleksi dan dokumen, mendukung sinkronisasi otomatis, kueri yang fleksibel, dan dapat diintegrasikan dengan layanan Firebase lain.

### F. Raspberry Pi 4B

Raspberry Pi 4 Model B adalah komputer papan tunggal (single-board computer) berukuran kecil yang dilengkapi prosesor quad-core ARM Cortex-A72, RAM hingga 8GB, port USB 3.0, HDMI ganda, dan konektivitas nirkabel seperti Wi-Fi serta Bluetooth. Perangkat ini mendukung berbagai sistem operasi berbasis Linux dan mampu menjalankan aplikasi komputasi ringan hingga menengah, termasuk pemrosesan data dan kontrol perangkat keras.

## III. METODE

Penelitian ini menggunakan pendekatan rekayasa sistem berbasis Internet of Things (IoT) yang mencakup perancangan perangkat keras, pengembangan backend berbasis REST API, pengembangan aplikasi mobile, integrasi dengan layanan cloud computing, dan pengujian sistem secara langsung di lapangan. Sistem dirancang untuk mengatur jadwal pemberian makanan kucing secara otomatis, mengelola profil kucing, serta menampilkan data pemantauan secara real-time melalui aplikasi mobile.

### A. Waktu dan Tempat Penelitian

Kegiatan penelitian dilaksanakan mulai awal Mei hingga akhir Juli 2025. Proses pengembangan dilakukan secara bertahap, dimulai dari perancangan, implementasi, hingga pengujian. Uji coba lapangan dilakukan di lingkungan rumah pengguna untuk memverifikasi fungsionalitas backend, konektivitas dengan perangkat IoT, serta integrasi layanan cloud.

### B. Prosedur Penelitian

Prosedur penelitian terdiri dari:

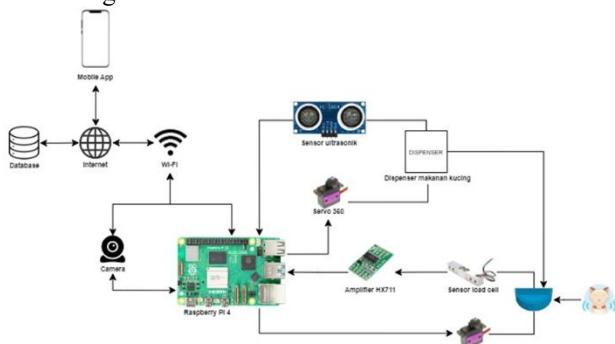
1. Studi literatur mengenai Internet of Things, arsitektur REST API, layanan Google Cloud Platform (Cloud Run, Cloud Storage, Compute Engine), dan Firestore.
2. Perancangan arsitektur sistem mencakup:
  - Desain backend menggunakan FastAPI.
  - Integrasi dengan Firestore untuk penyimpanan dan pengelolaan data.
  - Skema penyimpanan model *deep learning* dan file di Cloud Storage.
3. Implementasi sistem meliputi:
  - Pengembangan API untuk autentikasi pengguna, pengelolaan jadwal makan, profil kucing, dan pemantauan real-time.
  - Integrasi backend dengan Raspberry Pi 4B untuk kontrol aktuator, pembacaan sensor, dan pemrosesan data visual dari kamera.
4. Pengujian sistem
  - Pengujian fungsional API dan integrasi perangkat IoT.
  - Pengujian kinerja backend di lingkungan cloud untuk mengukur responsivitas dan skalabilitas.

### C. Sumber Data dan Cara Perolehannya

Data pada penelitian ini diperoleh langsung dari hasil pembacaan sensor yang terpasang pada perangkat Smart Pet Feeder. Sensor beban (load cell) digunakan untuk mengukur berat makanan yang dikeluarkan, sedangkan sensor ultrasonik digunakan untuk mengukur ketinggian atau volume makanan di wadah penyimpanan. Selain itu, kamera yang terhubung ke Raspberry Pi 4B berfungsi untuk menangkap citra kucing yang berada di dekat perangkat. Seluruh data dari sensor dan kamera diproses oleh Raspberry Pi 4B sebagai pengendali utama, kemudian dikirimkan ke

backend melalui API dengan koneksi internet. Data tersebut selanjutnya disimpan di Firestore dan Cloud Storage untuk pemrosesan lanjutan, serta ditampilkan kepada pengguna melalui aplikasi mobile.

#### D. Rancangan Sistem



GAMBAR 1  
Rancangan Umum Sistem

Rancangan sistem ini terdiri atas tiga bagian utama, yaitu:

##### 1. Sensor, Aktuator, dan Raspberry Pi 4B

Dua sensor yang digunakan yaitu ultrasonik untuk ketinggian/volume pakan dan load cell + HX711 untuk berat keluaran. Aktuator terdiri dari servo 360 dan servo 180. Kamera terhubung ke Raspberry Pi 4B. Pi memproses gambar lalu mengirimkannya ke backend.

##### 2. Platform Backend dan Database

Backend di-deploy pada Cloud Run. Data terstruktur disimpan di Firestore, sedangkan gambar/model di Cloud Storage. Platform menangani logika pemberian makan, penjadwalan, dan autentikasi.

##### 3. Aplikasi Mobile

Aplikasi terhubung ke backend via internet/Wi-Fi untuk menampilkan data sensor, status perangkat, dan citra kamera, serta menyediakan kontrol jarak jauh melalui antarmuka yang ringkas dan responsif.

#### E. Metode Pengujian

Metode pengujian terdiri dari tiga pengujian utama untuk mengukur fungsional dari backend, yaitu:

1. Uji Fungsional API, dilakukan untuk memastikan semua endpoint berjalan sesuai spesifikasi.
2. Uji Latensi, dilakukan untuk mengukur kinerja dan skalabilitas layanan di cloud.
3. Uji Penyimpanan, dilakukan untuk memastikan penyimpanan data/file benar, cepat, dan aman.

## IV. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah sistem Smart Pet Feeder yang mampu melakukan pemberian pakan kucing secara otomatis dan terjadwal serta menyediakan pemantauan real-time melalui aplikasi mobile. Dari sisi cloud computing, arsitektur backend dirancang cloud-native dengan layanan REST API pada Google Cloud Run yang terintegrasi dengan Cloud Firestore untuk data terstruktur dan Cloud Storage untuk file gambar dan model, serta observabilitas melalui Cloud Logging/Monitoring. Perangkat keras dan aplikasi mobile telah berhasil berkomunikasi dengan backend melalui internet. Kinerja arsitektur cloud dievaluasi melalui tiga

pengujian utama yaitu uji fungsional API, uji latensi, dan uji penyimpanan.

#### A. Hasil Implementasi Sistem

Sistem berhasil terhubung ke layanan cloud dan beroperasi stabil. Backend di Google Cloud Run terintegrasi dengan Cloud Firestore dan Cloud Storage melalui endpoint HTTPS, sehingga proses pengiriman serta penerimaan data dari Raspberry Pi dan aplikasi mobile berlangsung konsisten mulai dari autentikasi, pemrosesan permintaan, hingga penyimpanan serta pengambilan data.

#### B. Analisis Hasil Pengujian

##### 1. Penyimpanan Data di Firestore

Sistem telah berhasil menyimpan serta membaca data pengguna meliputi nama pengguna yang terregistrasi ke dalam aplikasi, profil kucing, jadwal pemberian makanan, dan riwayat pemberian makanan ke dalam Firestore.

GAMBAR 2  
Struktur Data Firestore

##### 2. Latensi Data

Rata-rata lantensi antara sistem ke layanan cloud dan berada pada kisaran 1 hingga 2 detik dipengaruhi oleh kualitas internet yang dapat dilihat pada tabel perbandingan waktu pengeluaran makanan berikut.

TABEL 1

Perbandingan Waktu Pengeluaran Makanan

Waktu yang Diatur pada Aplikasi	Waktu Makanan Dikeluarkan
17:42	17:42:01
17:45	17:45:01
17:50	17:50:01
17:53	17:53:02
17:57	17:57:02
18:01	18:01:01
18:02	18:02:01

##### 3. Penyimpanan File di Google Cloud Storage

Sistem telah berhasil menyimpan serta membaca File pengguna yang meliputi gambar kucing dan model deteksi *deep learning* di Google Cloud Storage. Proses unggah dan unduh dilakukan melalui endpoint terautentikasi pada Cloud Run dengan penggunaan signed URL dan kebijakan IAM untuk membatasi akses, sehingga integritas dan keamanan File tetap terjaga.

GAMBAR 3

Struktur Penyimpanan File Google Cloud Storage

## V. KESIMPULAN

Penelitian ini berhasil mewujudkan sistem Smart Pet Feeder dengan arsitektur cloud-native yang stabil dan skalabel. Backend REST API dijalankan di Google Cloud Run dan terintegrasi dengan Cloud Firestore untuk data terstruktur serta Cloud Storage untuk File gambar dan model, serta didukung observabilitas melalui Cloud Logging/Monitoring. Perangkat keras (Raspberry Pi 4B dengan sensor ultrasonik, load cell + HX711, kamera, dan servo) serta aplikasi mobile terbukti berkomunikasi lancar dengan backend melalui internet. Seluruh API yang dibangun bekerja sesuai harapan tanpa ada anomali. Uji coba menunjukkan layanan merespons cepat dan stabil, dengan waktu tanggap rata-rata sekitar 1–2 detik untuk permintaan umum. Penyimpanan data dan File pengguna berjalan lancar serta aman. Secara keseluruhan, tujuan penelitian tercapai: sistem responsif, dapat diandalkan, dan mudah diakses dari perangkat pengguna. Keterbatasan utama ada pada ketergantungan kualitas internet rumah dan potensi melambat saat mengunggah gambar berukuran besar.

## REFERENSI

- [1] S. Julianti, I. N. Qomariah, M. A. Anshari, and K. A. Widayati, "Aktivitas Harian dan Perilaku Makan Kucing Domestik Liar di Lingkungan Kantin IPB," *Al-Kauniyah: Jurnal Biologi*, vol. 14, no. 2, 2021, pp. 244–253.
- [2] P. Borra, "A Survey of Google Cloud Platform (GCP): Features, Services, and Applications," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 4, no. 3, pp. 191–199, June 2024.
- [3] J. Yang and A. Abraham, "Analyzing the Features, Usability, and Performance of Deploying a Containerized Mobile Web Application on Serverless Cloud Platforms," *Future Internet*, vol. 16, no. 12, 2024, p. 475.
- [4] D. Durner, V. Leis, and T. Neumann, "Exploiting cloud object storage for high-performance analytics," *Proceedings of the VLDB Endowment*, vol. 16, no. 11, 2023, pp. 2769–2782.
- [5] S. A. R. Shah, A. Waqas, M.-H. Kim, T.-H. Kim, H. Yoon, and S.-Y. Noh, "Benchmarking and Performance Evaluations on Various Configurations of Virtual Machine and Containers for Cloud-Based Scientific Workloads," *Applied Sciences*, vol. 11, no. 3, 2021, 993.