LAPORAN PRAKTIKUM TEKNOLOGI CLOUD COMPUTING

Men-deploy Front-End dan Back-End ke Cloud Run dan App Engine secara CI/CD dengan memanfaatkan Cloud Build.

Disusun oleh

Nama : Aziizah Intan Ramadhan N

NIM : 123220201

PROGRAM STUDI INFORMATIKA

JURUSAN INFORMATIKA

FAKULTAS TEKNIK INDUSTRI

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

YOGYAKARTA

2025

HALAMAN PENGESAHAN

LAPORAN PRAKTIKUM

Disusun Oleh :

Aziizah Intan Ramadhan N 123220201

Telah diperiksa dan disetujui oleh Asisten Praktikum……

Pada tanggal : ……

.

Menyetujui.

Asisten Praktikum	**Asisten Praktikum**
Berlyandica Alam F	**Faustina Chelloana T**
NIM 123210060	**NIM 123210139**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan laporan praktikum Teknologi Cloud Computing . Adapun laporan ini berisi tentang tugas Web Service (Rest Api).

Tidak lupa ucapan terima kasih kepada dosen mata kuliah Teknologi Cloud Computing, Bapak Dr. Awang Hendrianto P., S.T., M.T. serta asisten praktikum Teknologi Cloud Computing yang sudah membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terima kasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, 20 April 2025
Penulis

DAFTAR ISI

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, pengembangan aplikasi berbasis web memerlukan pendekatan yang efisien, cepat, dan andal, khususnya dalam hal deployment. Proses deploy tradisional yang dilakukan secara manual sering kali menyebabkan banyak kendala seperti human error, ketidakkonsistenan antar lingkungan, serta waktu yang tidak efisien. Hal ini menjadi tantangan utama, terutama ketika sebuah aplikasi terdiri dari arsitektur frontend dan backend yang terpisah.

Untuk mengatasi masalah tersebut, pendekatan Continuous Integration and Continuous Deployment (CI/CD) menjadi solusi efektif. CI/CD memungkinkan otomatisasi proses build, testing, dan deployment sehingga mempercepat siklus pengembangan sekaligus meningkatkan kualitas aplikasi. Dengan pipeline CI/CD yang baik, tim pengembang dapat memastikan bahwa setiap perubahan kode diuji secara otomatis dan diterapkan ke lingkungan produksi dengan konsisten tanpa intervensi manual.

Selain itu, penggunaan teknologi containerisasi seperti Docker serta orkestrasi menggunakan Kubernetes juga mendukung fleksibilitas dalam deployment aplikasi frontend maupun backend secara terpisah namun tetap terintegrasi dengan baik di lingkungan cloud.

1.2 Rumusan Masalah

Bagaimana cara menerapkan proses CI/CD secara otomatis dan terintegrasi untuk aplikasi frontend dan backend dengan menggunakan layanan Cloud Build, Cloud Run, serta App Engine di Google Cloud Platform?

1.3 Tujuan

- Mengotomatisasi proses build dan deployment untuk aplikasi frontend dan backend dengan memanfaatkan Cloud Build.
- Melakukan deployment backend secara otomatis ke Cloud Run menggunakan konfigurasi pada file cloudbuild.backend.yaml.
- Melakukan deployment frontend ke App Engine melalui pengaturan di cloudbuild.frontend.yaml dan file app.yaml.
- Menghubungkan backend dengan frontend dengan mengatur variabel BASE_URL yang sesuai dalam aplikasi frontend.
- Menyediakan akses publik ke aplikasi melalui URL yang disediakan oleh App Engine.

1.4 Manfaat

- Meningkatkan efisiensi dan konsistensi dalam proses deployment aplikasi.
- Meminimalkan kesalahan deployment yang disebabkan oleh proses manual.
- Memudahkan pengembangan berkelanjutan dan kolaboratif antar tim melalui otomatisasi CI/CD.
- Memberikan gambaran nyata mengenai penerapan teknologi cloud dalam proses DevOps modern.
- Menyediakan solusi praktis dan skalabel untuk deployment aplikasi berbasis arsitektur microservices.

BAB II

TINJAUAN LITERATUR

2.1 Google Cloud Platform

Google Cloud Platform (GCP) adalah layanan komputasi awan yang disediakan oleh Google, menawarkan infrastruktur, platform, dan layanan yang digunakan untuk mengembangkan, menguji, dan menjalankan aplikasi. Beberapa layanan GCP yang mendukung implementasi CI/CD antara lain:

2.1.1 Cloud Build

Cloud Build merupakan layanan dari Google Cloud Platform (GCP) yang berfungsi untuk secara otomatis membangun dan menguji aplikasi berdasarkan konfigurasi yang ditentukan dalam file `cloudbuild.yaml`. File tersebut memuat serangkaian langkah yang dijalankan saat pipeline diaktifkan, seperti pemasangan dependensi, proses build aplikasi, serta deployment. Cloud Build juga dapat terintegrasi dengan berbagai sistem pengelolaan versi seperti GitHub dan Cloud Source Repositories.

2.1.2 Cloud Run

Cloud Run adalah layanan serverless yang memungkinkan pengembang menjalankan container secara otomatis dengan kemampuan skalabilitas tinggi. Layanan ini sangat cocok untuk aplikasi backend yang dikemas dalam container dan dijalankan sesuai kebutuhan. Proses deployment ke Cloud Run dapat diotomatisasi menggunakan Cloud Build, sehingga memudahkan pengelolaan siklus hidup aplikasi.

2.1.3 App Engine

App Engine merupakan platform sebagai layanan (PaaS) yang memungkinkan penggelaran aplikasi web tanpa harus mengelola infrastruktur server secara langsung. Baik aplikasi frontend berbasis web statis maupun

dinamis dapat dijalankan di App Engine dengan konfigurasi yang mudah melalui file `app.yaml`.

2.2 Continuous Integration dan Continuous Deployment (CI/CD)

Continuous Integration (CI) merupakan praktik di mana tim pengembang secara rutin menggabungkan kode ke dalam repositori bersama. Setiap perubahan kode tersebut kemudian diuji dan dibangun secara otomatis untuk memastikan tidak ada konflik atau kesalahan yang terlewatkan. Selanjutnya, Continuous Deployment (CD) melanjutkan proses ini dengan secara otomatis menerapkan hasil build yang berhasil ke lingkungan produksi. Dengan mengadopsi CI/CD, proses pengembangan perangkat lunak menjadi lebih cepat, stabil, dan lebih mudah dikelola serta dipelihara.

2.3 MySQL dalam Cloud

MySQL di Cloud merupakan layanan basis data relasional yang dioperasikan dan dikelola dalam lingkungan komputasi awan, sehingga organisasi dapat memanfaatkan keunggulan MySQL tanpa harus mengurus infrastruktur fisik secara langsung. Layanan ini menyediakan kemampuan skalabilitas yang fleksibel, memungkinkan perusahaan untuk menyesuaikan kapasitas sesuai dengan kebutuhan bisnis yang berkembang tanpa perlu investasi besar pada perangkat keras. Penyedia layanan cloud utama seperti AWS (dengan Amazon RDS for MySQL), Google Cloud Platform (dengan Cloud SQL for MySQL), dan Microsoft Azure (dengan Azure Database for MySQL) menawarkan solusi managed MySQL yang mencakup fitur-fitur seperti backup otomatis, patching keamanan, pemantauan performa, serta replikasi untuk ketersediaan tinggi.

Dengan menggunakan MySQL dalam cloud, perusahaan dapat fokus pada pengembangan aplikasi dan analisis data tanpa terbebani oleh tugas-tugas operasional database sehari-hari. Selain itu, model pembayaran berbasis penggunaan memungkinkan efisiensi biaya sesuai dengan skala kebutuhan.

BAB III

METODOLOGI

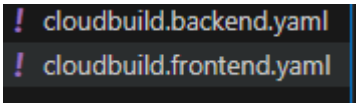
3.1 Analisis Permasalahan

Pada penugasan praktikum ini, permasalahan utama yang dihadapi adalah bagaimana menerapkan proses Continuous Integration dan Continuous Deployment (CI/CD) untuk aplikasi yang memiliki arsitektur terpisah antara frontend dan backend. Tujuan dari penugasan ini adalah untuk menyederhanakan dan mengotomatisasi proses deployment, sehingga setiap perubahan kode dapat langsung di-deploy ke lingkungan produksi tanpa proses manual.

3.2 Perancangan Solusi

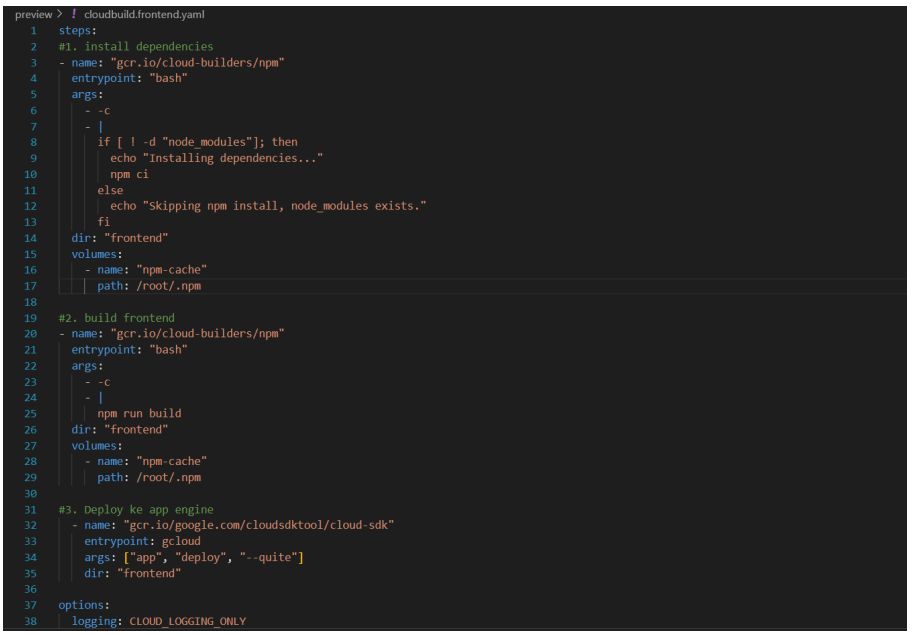
Berdasarkan hasil analisis permasalahan yang dijelaskan diatas bahwa hal yang perlu dipersiapkan yaitu :

1. Membuat dan menambahkan file cloudbuild.backend.yaml dan cloudbuild.frontend.yaml.



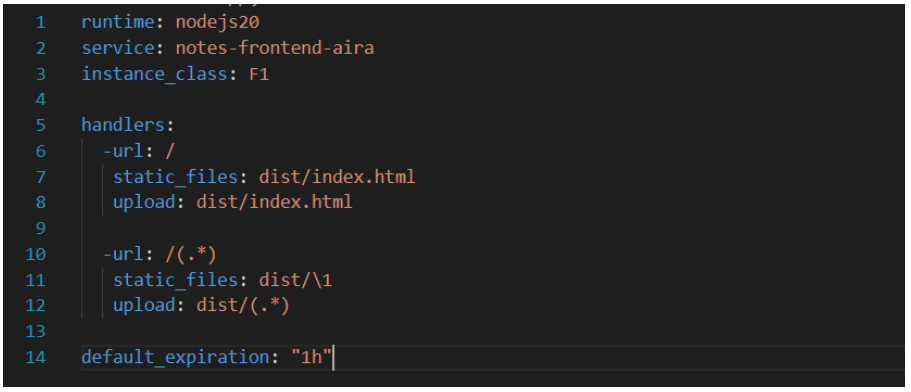
Gambar 3.1 File Cloudbuild

Gambar 3.2 File Cloudbuild Backend



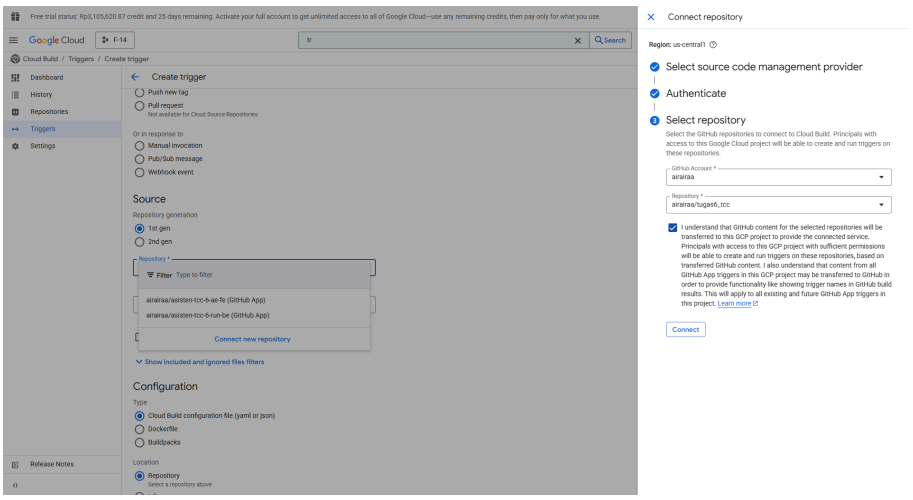
Gambar 3.3 File Cloudbuild Frontend

2. Membuat dan menambahkan file app.yaml

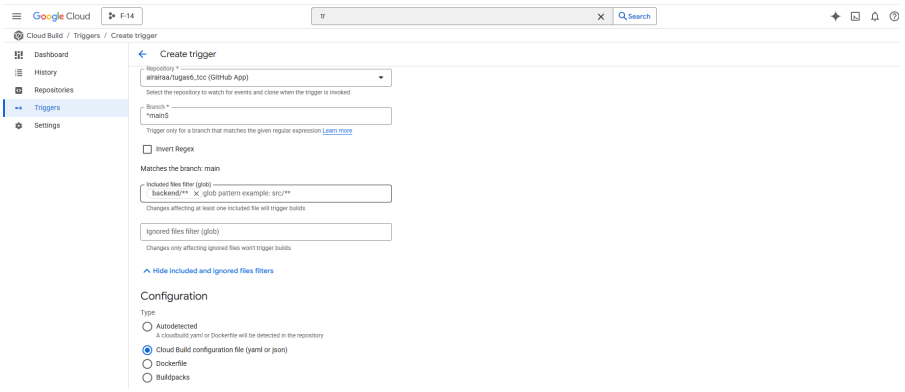


Gambar 3.4 File App.yaml

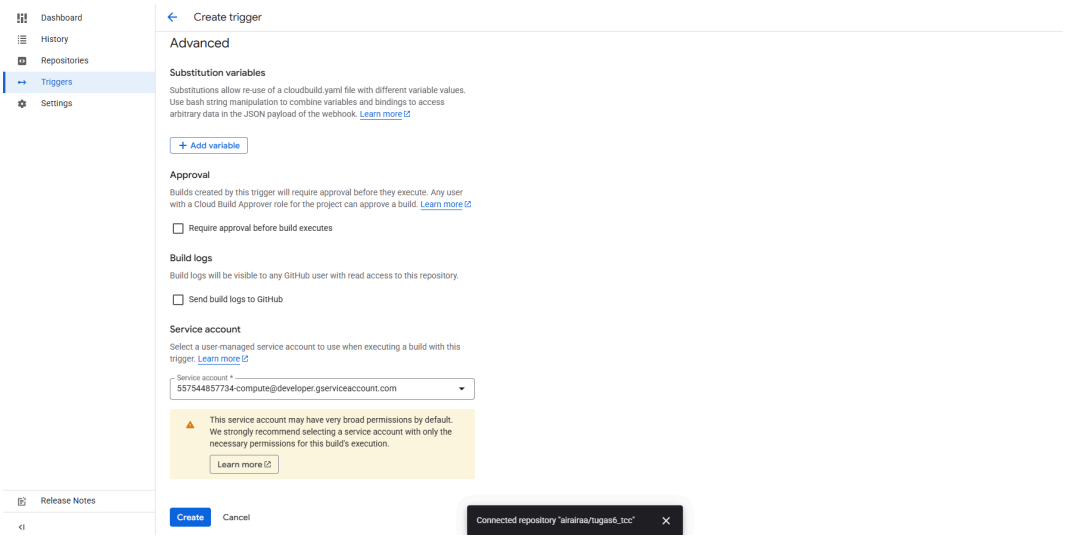
3. Membuat trigger



Gambar 3.5 Membuat Trigger

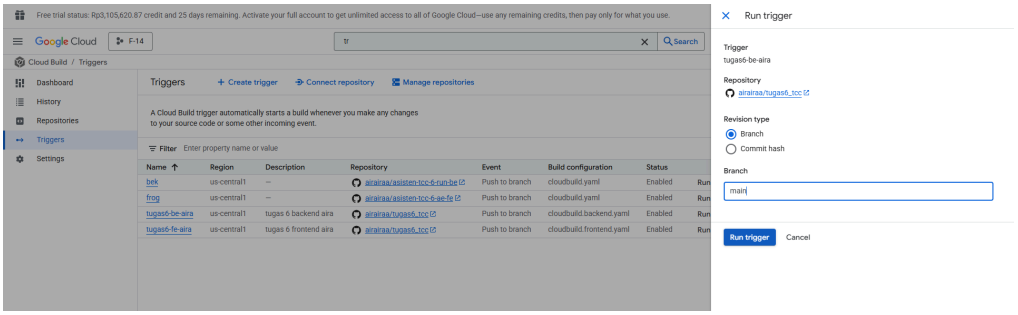


Gambar 3.6 Trigger backend



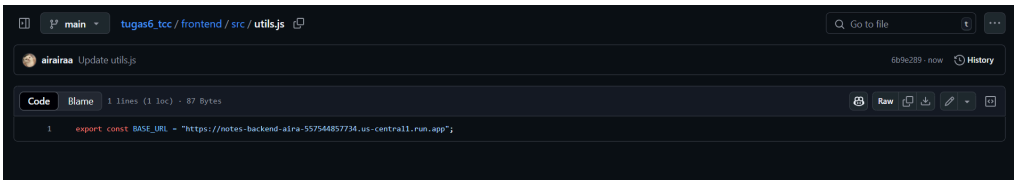
Gambar 3.7 Trigger Frontend

4. Jalankan



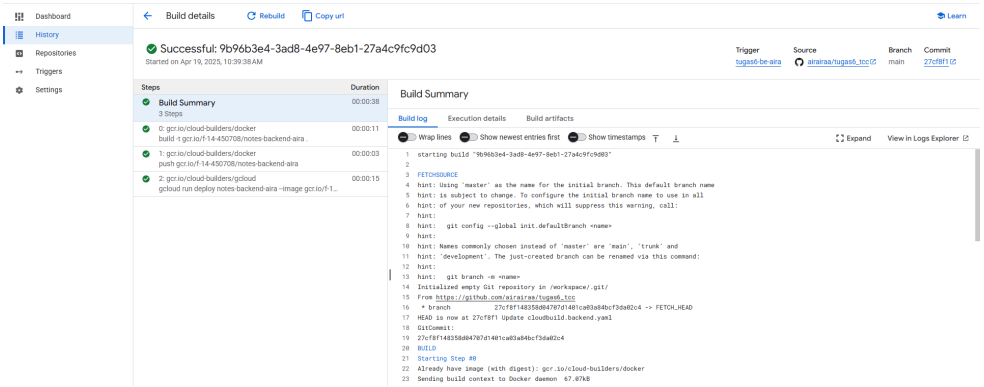
Gambar 3.8 Run Trigger

5. Mengganti BASE_URL dengan link backend yang sudah dijalankan

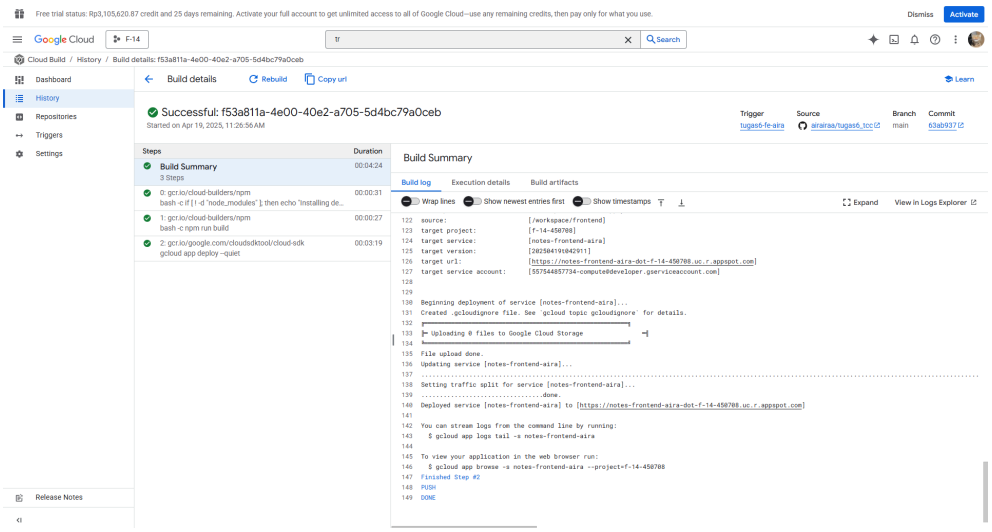


Gambar 3.9 Ganti BASE_URL

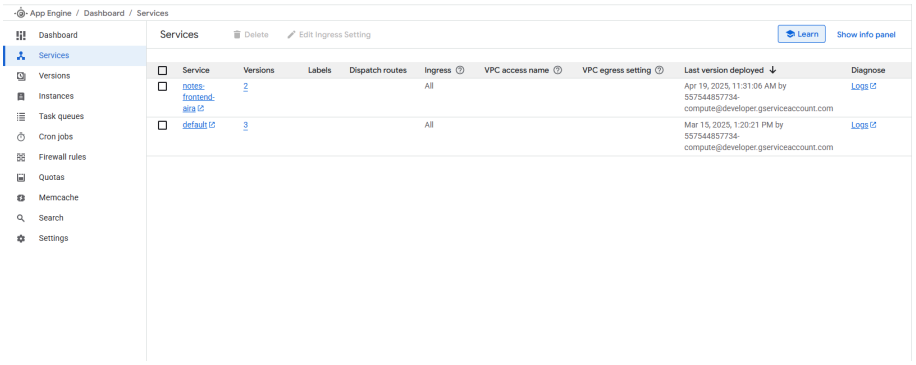
6. Mengakses link yang tersedia



Gambar 3.10 Build backend

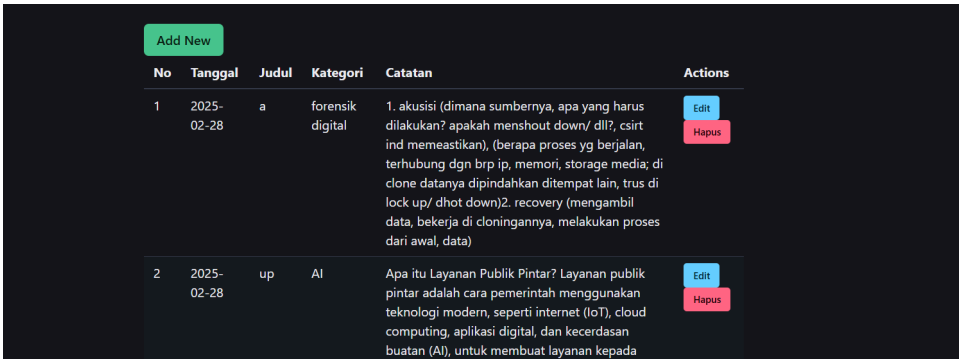


Gambar 3.11 Build Frontend



Gambar 3.12 Service

7. Hasil akhir



Gambar 3.12 Hasil

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil implementasi menunjukkan bahwa proses CI/CD berhasil dijalankan untuk frontend dan backend dengan memanfaatkan Cloud Build, Cloud Run, serta App Engine. Backend berhasil dideploy ke Cloud Run melalui pipeline yang diatur dalam file cloudbuild.backend.yaml, menghasilkan URL publik yang dapat diakses. Sementara itu, frontend juga sukses dideploy ke App Engine menggunakan konfigurasi pada cloudbuild.frontend.yaml dan app.yaml. Setelah backend aktif, URL tersebut dimasukkan ke dalam konfigurasi frontend sebagai nilai dari variabel BASE_URL, sehingga frontend dapat berkomunikasi dengan backend secara lancar.

4.2 Pembahasan

Implementasi ini membuktikan bahwa CI/CD mampu mempercepat serta menyederhanakan proses deployment. Cloud Build secara otomatis mengelola proses build dan deploy setiap kali terjadi perubahan pada repository. Cloud Run sangat efektif untuk backend karena mendukung containerisasi dan penskalaan otomatis, sedangkan App Engine cocok untuk frontend berkat kemudahan konfigurasi dan deployment-nya. Tantangan utama yang dihadapi adalah pengaturan BASE_URL yang saat ini masih dilakukan secara manual. Ke depannya, proses ini dapat diotomatisasi agar integrasi antara frontend dan backend berjalan lebih mulus dan efisien.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa penerapan CI/CD untuk proses deployment frontend dan backend dengan memanfaatkan layanan Google Cloud Build, Cloud Run, dan App Engine telah berhasil dilaksanakan dengan baik. Backend berhasil dibangun dan dideploy secara otomatis ke Cloud Run, sedangkan frontend berhasil dideploy ke App Engine. Sistem ini memungkinkan integrasi yang efisien antara frontend dan backend, serta meningkatkan kecepatan dan konsistensi dalam proses deployment. Proyek ini berhasil menjawab rumusan masalah serta mencapai tujuan yang telah ditetapkan, dan solusi yang dibangun terbukti layak untuk digunakan dalam pengembangan aplikasi secara modern dan profesional.

5.2 Saran

Sebagai langkah lanjutan dari hasil yang telah dicapai, disarankan agar proses penyesuaian BASE_URL diotomatisasi melalui integrasi dalam pipeline atau dengan menggunakan penyimpanan konfigurasi berbasis environment variables. Pendekatan ini akan meningkatkan fleksibilitas sekaligus mengurangi risiko kesalahan akibat pengaturan manual. Selain itu, perlu dilakukan pengujian lebih mendalam terkait performa dan keamanan sistem guna memastikan kestabilan aplikasi saat dijalankan pada skala produksi. Ke depannya, solusi ini dapat dikembangkan lebih lanjut dengan menambahkan fitur monitoring otomatis serta skalabilitas yang lebih baik untuk mendukung pertumbuhan pengguna dan kompleksitas aplikasi.

DAFTAR PUSTAKA

Sato, Y. (2020). Cloud Native DevOps with Kubernetes: Building, Deploying, and Scaling Modern Applications in the Cloud. O'Reilly Media.

Turnbull, J. (2014). The Docker Book: Containerization is the new virtualization. James Turnbull.

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A Software Architect's Perspective. Addison-Wesley Professional.