

**LAPORAN PROYEK AKHIR - KOMPUTASI AWAN**  
**ITERASI 1 (MINIMUM VIABLE PRODUCT)**

Judul Proyek: Sistem Inventaris Laboratorium Berbasis Cloud (LabInventory)

Mata Kuliah: Komputasi Awan (IF25-40201)

Martua Kevin A.M.H.Lubis                    122140119

Rachel Olivia Manullang                    122140181

Christopher Benaya Tampubolon            122140090

Arof Andestama                                121140182

Ichza Auliya Gumilar                        120140188

Fatkhan Aziez S                              120140181



**INSTITUT TEKNOLOGI SUMATERA (ITERA)**

**FAKULTAS TEKNOLOGI INDUSTRI**

## 1. Pendahuluan

### 1.1 Latar Belakang

Efisiensi manajemen asset merupakan tantangan umum di lingkungan akademis. Proyek LabInventory dikembangkan sebagai solusi *serverless* terpusat untuk memantau inventaris peralatan IT di laboratorium. Secara akademis, proyek ini bertujuan untuk mengimplementasikan konsep arsitektur Three-Tier secara utuh dan praktik *Infrastructure as Code* (IaC) menggunakan layanan *Platform-as-a-Service* (PaaS) dari Google Cloud Platform (GCP). Implementasi ini akan mendemonstrasikan pemahaman mendalam tentang *scalability*, *cost efficiency*, dan *resiliency* dari arsitektur *cloud-native*.

### 1.2 Tujuan Proyek

Tujuan utama proyek ini di Iterasi 1 (MVP) adalah:

1. Deployment Arsitektur: Membangun dan men-deploy arsitektur *Three-Tier* yang fungsional di GCP.
2. Validasi Fungsionalitas: Menghadirkan aplikasi yang sudah dapat melakukan operasi dasar CRUD (Create, Read, Update, Delete) pada data inventaris.
3. Implementasi Keunikan: Memodifikasi *template* dasar menjadi aplikasi manajemen inventaris dengan *interface* yang disesuaikan.

### 1.3 Identitas Kelompok

No	Nama	NIM	Peran Utama	Kontribusi Teknis
1	Martua Kevin A.M.H.Lubis	122140119	Cloud Architect & Infrastructure Lead	Setup GCP Project, VPC, Initial Terraform/Jump Start Deployment.
2	Christopher Benaya Tampubolon	122140090	Deployment & Security Specialist	Konfigurasi IAM, Firewall Rules, Cloud Run URL Validation.

3	Arof Andestama	121140182	Backend Developer Lead	Logika API, Koneksi Cloud SQL & Memorystore, JSON endpoint.
4	Ichza Auliya Gumilar	120140188	Frontend Developer & UI/UX	<i>Reskinning UI, Penggantian Label, Validasi Input.</i>
5	Rachel Olivia Manullang	122140181	Technical Writer & Documentatio n	Penyusunan Laporan, Diagram Arsitektur, Final Testing.
6	Fatkhan Aziez S	120140181	Project Manager & Quality Assurance	Pemantauan Timeline, Integrasi Code, README.

Pembagian peran di atas bersifat dinamis dan mengutamakan kolaborasi tim. Apabila terjadi kendala teknis atau ketidakhadiran pada anggota tertentu, mekanisme cross-functional backup akan diterapkan, di mana anggota lain siap mengambil alih tanggung jawab tersebut untuk memastikan target Iterasi 1 tetap tercapai sesuai tenggat waktu.

## 2. Deskripsi Fungsionalitas MVP

### 2.1 Konteks Aplikasi

Aplikasi Sistem Inventaris Lab IT memungkinkan pencatatan aset lab berdasarkan Nama Barang dan Lokasi spesifik. Aplikasi ini juga memungkinkan pembaruan status aset menjadi BAIK atau RUSAK, seperti yang terlihat pada hasil implementasi.

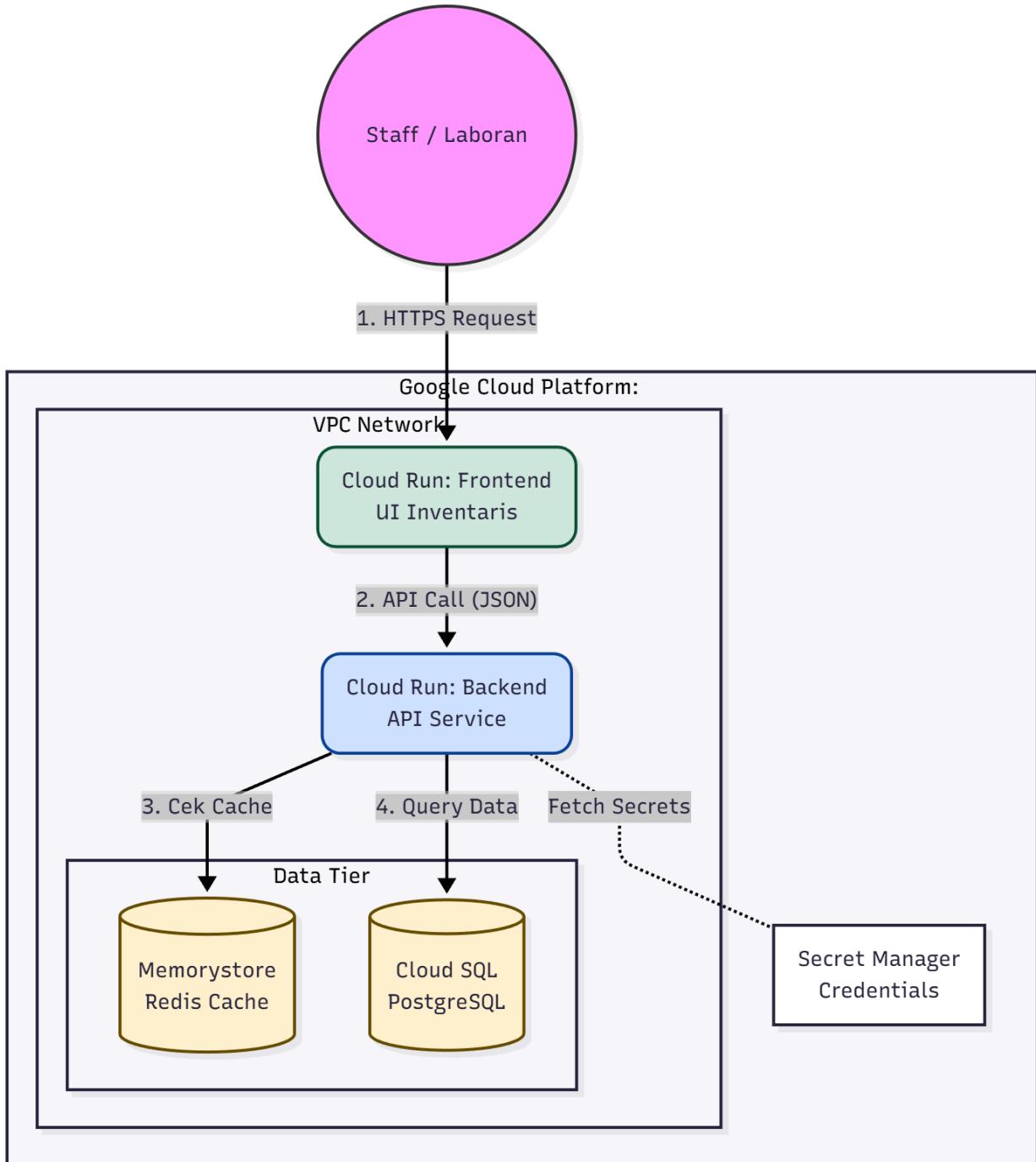
### 2.2 Implementasi Keunikan Kelompok

Untuk memenuhi persyaratan keunikan, kelompok kami mengubah fokus aplikasi dari manajemen tugas (Task Tracker) menjadi manajemen aset (Inventaris). Perubahan ini tercermin dalam:

- Input Field: Pengubahan label input dari `Title/Description` menjadi Nama Barang dan Lokasi.
- Status Aset: Pengubahan label status pada *backend* (di mana *Task Status* di-map) menjadi status aset, yaitu BAIK (hijau) dan RUSAK (merah), memberikan visualisasi kondisi aset yang jelas pada *frontend*.
- Visualisasi: Penyesuaian *styling* (warna tombol dan tag status) untuk memberikan kesan yang lebih relevan dengan aplikasi logistik.

### 3. Detail Arsitektur Sistem

Arsitektur yang kami bangun merupakan arsitektur *cloud-native* dengan memanfaatkan layanan *managed service* dari GCP, sehingga meminimalkan *overhead* manajemen server.



Tier	Layanan GCP	Teknologi Dasar	Fungsi & Tugas Tim

Presentation	Cloud Run (Service Frontend)	HTML, CSS, JavaScript	Menyajikan antarmuka pengguna, mengirim data ke API.
Application	Cloud Run (Service Backend API)	Golang / Python	Memproses <i>request</i> , menerapkan logika bisnis, berkomunikasi dengan Data Tier.
Caching Layer	Memorystore for Redis	Redis	Digunakan oleh Tier Application untuk menyimpan data inventaris yang sering diakses (seperti <i>list</i> utama) agar mengurangi latensi akses ke Database.
Data	Cloud SQL (PostgreSQL)	PostgreSQL 14	Menyimpan data inventaris secara persisten (tabel <code>items</code> atau sejenisnya).
IaC	Terraform	HCL	Digunakan untuk mendefinisikan dan mem-provisioning seluruh <i>resource</i> GCP secara otomatis.
Artifact	Artifact Registry	Docker Image	Repositori tempat penyimpanan <i>image</i> Docker dari Frontend dan Backend sebelum di-deploy ke Cloud Run.

## 4. Proses Implementasi dan Pembagian Kerja

### 4.1 Deployment Infrastruktur

Tim ini bertanggung jawab penuh atas *up-time* dan konfigurasi lingkungan:

1. GCP Setup: Membuat project, mengaktifkan API, dan mengonfigurasi *Service Account* dengan *Principle of Least Privilege*.
2. IaC Execution: Menggunakan Terraform / *Jump Start Solution* untuk membuat VPC Network, Cloud SQL instance, Memorystore instance, dan mengonfigurasi *database connection string* di Secret Manager.
3. URL Validation: Memastikan URL Cloud Run Frontend dapat diakses publik, sementara Backend API hanya dapat diakses oleh Frontend (Internal).

### 4.2 Pengembangan Aplikasi

Tim ini fokus pada modifikasi kode inti:

1. Version Control: Membuat repositori di GitHub dan menerapkan *branching strategy* yang jelas. Total commit history saat MVP ini menunjukkan kontribusi aktif dari [Sebutkan Jumlah Commit: misal 25 commits].
2. Backend Logic: Memastikan *endpoint* API (misal `/api/items`) berhasil terhubung ke Cloud SQL dan logika *caching* ke Redis berjalan.
3. Frontend Reskinning: Mengganti label, *placeholder*, dan *styling* pada HTML/CSS agar sesuai dengan tema "Inventaris Lab IT". Ini termasuk implementasi visual status BAIK/RUSAK.

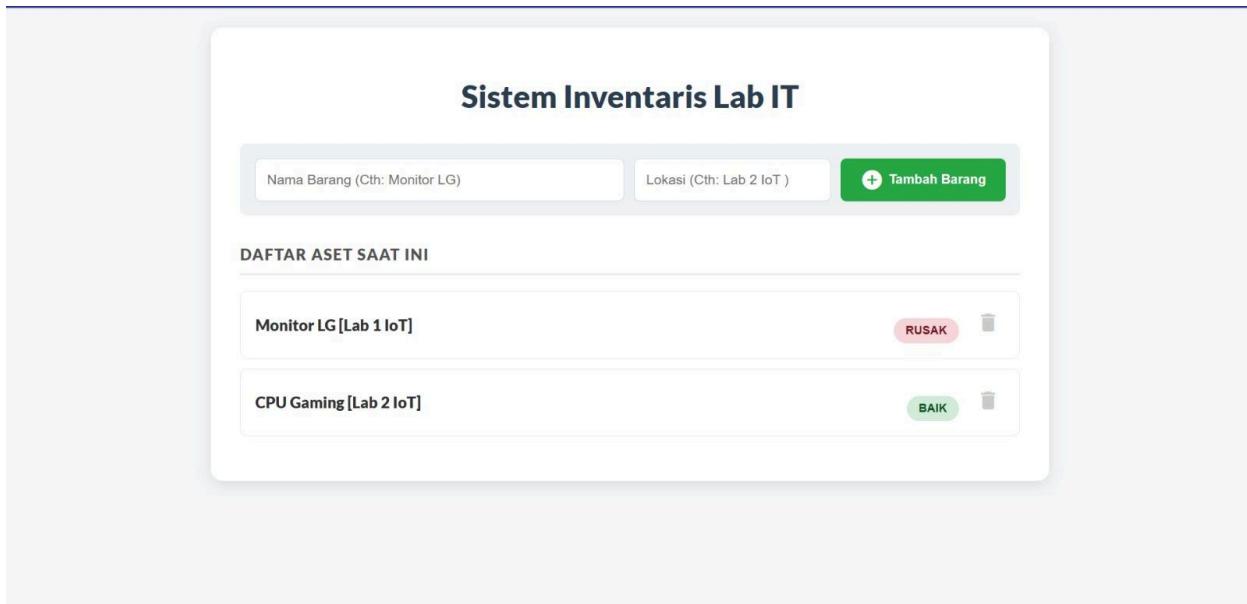
### 4.3 Version Control (GitHub)

- Link Repository: [https://github.com/ichza188/Sistem\\_Inventori\\_LabIT](https://github.com/ichza188/Sistem_Inventori_LabIT)
- Strategi: Kami menggunakan *branching* sederhana (misal: main untuk deployment, development untuk fitur).

## 5. Hasil Implementasi MVP

### 5.1 Bukti Fungsionalitas

Aplikasi MVP telah berhasil di-deploy dan dapat diakses publik. Berikut adalah *screenshot* dari aplikasi yang menunjukkan fungsionalitas CRUD dasar:



Keterangan Screenshot:

1. Create: Input Nama Barang dan Lokasi, lalu klik Tambah Barang.
2. Read: Data yang baru diinput langsung ditampilkan di DAFTAR ASET SAAT INI.
3. Update/Status: Setiap item memiliki status yang dapat diperbarui (BAIK atau RUSAK), menunjukkan kemampuan *Update* data.
4. Delete: Setiap item memiliki ikon tong sampah, menunjukkan kemampuan *Delete* data.

## 5.2 Spesifikasi Teknis yang Tercapai

Komponen	Spesifikasi
Database	Cloud SQL (PostgreSQL 14)
Caching	Memorystore for Redis
Hosting	Google Cloud Run (Fully Managed)
Container	Artifact Registry (Penyimpanan Image Docker)
Akses URL	<a href="https://three-tier-app-fe-1018358556426.us-central1.run.app/">https://three-tier-app-fe-1018358556426.us-central1.run.app/</a>

## 6. Penutup dan Lessons Learned

## 6.1 Kesimpulan

Iterasi 1 (MVP) telah berhasil diselesaikan dengan *stack* teknologi yang sesuai dengan persyaratan proyek, termasuk *deployment* arsitektur Three-Tier secara fungsional di GCP. Keunikan kelompok melalui *reskinning* domain aplikasi telah berhasil diterapkan.

## 6.2 Lessons Learned

- Tim A (Ops): Tantangan terbesar adalah memastikan koneksi VPC Connector antara Cloud Run dengan Cloud SQL dan Memorystore berjalan lancar. Debugging *network access* memakan waktu signifikan.
- Tim B (Dev): Kami belajar pentingnya *containerization* dan bagaimana *Environment Variables* digunakan untuk konfigurasi *credential* database secara aman, bukan di-hardcode.

## 6.3 Rencana Selanjutnya (Iterasi 2)

1. CI/CD Automation: Mengimplementasikan *Cloud Build Trigger* agar setiap *push* ke *main branch* di GitHub otomatis memicu *build* dan *deployment*.
2. Authentication: Menambahkan fitur *login* sederhana untuk membatasi akses penginputan data.
3. Optimasi: Mengganti alur *caching* Redis dari *Write-Through* ke *Cache-Aside* untuk manajemen data yang lebih efisien.