

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Transformasi digital telah menjadi salah satu pendorong utama kemajuan di berbagai sektor, termasuk pertanian. Dalam konteks ini, pemanfaatan teknologi berbasis Internet of Things (IoT) dan layanan cloud menjadi peluang besar untuk mengatasi tantangan konvensional, seperti pengelolaan sumber daya dan monitoring yang masih dilakukan secara manual.

Teknologi cloud-native kini menjadi solusi unggulan karena fleksibilitas dan skalabilitasnya dibandingkan dengan server on-premise. Arsitektur ini mendukung pengembangan sistem yang mampu menangani data besar, memproses informasi secara real-time, dan memberikan analitik yang dapat diakses dari berbagai lokasi secara efisien.

Dalam upaya meningkatkan produktivitas dan monitoring di bidang pertanian, proyek penerapan sistem smartfarming berbasis cloud-native menjadi langkah strategis. Sistem ini menggabungkan sensor IoT untuk memantau kondisi pertanian dan teknologi cloud untuk menyimpan serta mengelola data guna mendukung pengambilan keputusan berbasis informasi yang real-time.

Politeknik Negeri Malang sebagai pemilik proyek ini bertujuan untuk memberikan dampak nyata dalam dunia pendidikan sekaligus membantu masyarakat. Dengan menerapkan solusi berbasis AWS, proyek ini juga menjadi kesempatan strategis untuk mengaplikasikan teknologi modern guna memenuhi kebutuhan transformasi digital di era IoT.

1.2 Tujuan dan Manfaat

1.2.1 Tujuan Umum Magang

Tujuan Magang secara umum adalah meningkatkan pengetahuan dan keterampilan serta pengalaman kerja bagi mahasiswa mengenai kegiatan Perusahaan/industry/instansi/ dan/unit bisnis strategis lainnya yang layak dijadikan tempat Magang. Selain itu, tujuan magang adalah melatih mahasiswa agar lebih

kritis terhadap perbedaan atau kesenjangan (gap) yang mereka jumpai dilapangan dengan yang di bangku kuliah. Dengan demikian mahasiswa diharapkan mampu untuk mengembangkan keterampilan tertentu yang tidak peroleh dikampus.

1.2.2 Tujuan Khusus Magang

Tujuan khusus yang diharapkan dapat dicapai melalui kegiatan magang mencakup hal-hal berikut:

- a. Mengaplikasikan arsitektur cloud-native menggunakan AWS untuk menyediakan infrastruktur yang fleksibel dan efisien dalam mendukung sistem smartfarming.
- b. Meningkatkan produktivitas serta kemampuan monitoring jarak jauh melalui pemanfaatan teknologi IoT dan layanan cloud.
- c. Membangun sistem yang mampu memproses data secara real-time dan memberikan informasi yang relevan bagi pengguna.
- d. Memberikan kontribusi teknologi modern kepada Politeknik Negeri Malang sebagai bentuk inovasi pendidikan yang aplikatif dan berdampak pada masyarakat luas.

1.2.3 Manfaat Magang

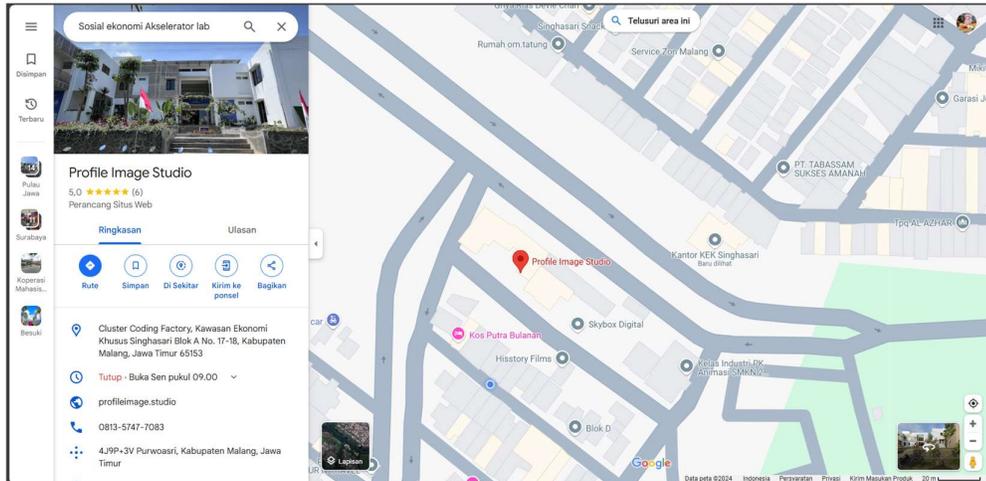
Pada kegiatan magang yang dilakukan memiliki beberapa manfaat antara lain:

- a. Mahasiswa terlatih untuk menangani tugas lapangan sekaligus melakukan serangkaian keterampilan yang sesuai dengan bidang keahliannya.
- b. Membuka peluang kerjasama yang lebih intensif serta gambaran perkembangan ipteks yang diharapkan di industri/instansi untuk menjaga mutu dan relevansi kurikulum.
- c. Mahasiswa memiliki kesempatan untuk memperkuat keterampilan dan pengetahuannya, yang akan berkontribusi pada peningkatan kepercayaan diri dan kedewasaannya.

1.3 Lokasi dan Jadwal Kerja

Lokasi kegiatan magang (PKL) ini dilaksanakan di SEAL (Sosial Ekonomi

Akselerator Lab) yang terletak di Cluster Coding Factory, Kawasan Ekonomi Khusus Singhasari Blok A No. 17-18, Kabupaten Malang, Jawa Timur 65153. Di bawah ini disajikan peta lokasi pelaksanaan magang (PKL).



Gambar 1.1 Peta Lokasi Magang

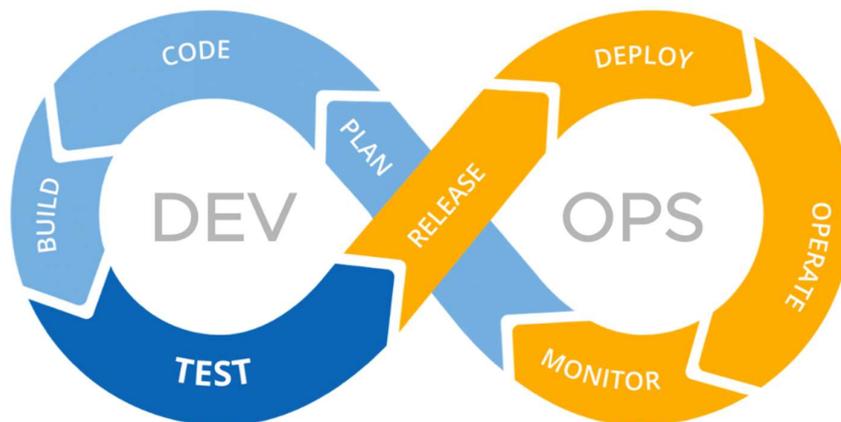
Adapun jadwal kegiatan magang ini dilaksanakan mulai tanggal 6 September 2024 sampai dengan tanggal 31 Desember 2024. Berikut merupakan jadwal kerja yang ditetapkan oleh SEAL selama empat bulan yang dituangkan dalam bentuk tabel.

Tabel 1.1 Jadwal Kerja Magang

No	Hari	Jam
1.	Senin	09.00 – 17.00 WIB
2.	Selasa	09.00 – 17.00 WIB
3.	Rabu	09.00 – 17.00 WIB
4.	Kamis	09.00 – 17.00 WIB
5.	Jumat	09.00 – 17.00 WIB
6.	Sabtu	Libur
7.	Minggu	Libur

1.4 Metode Pelaksanaan

Metode yang diterapkan dalam pelaksanaan Praktik Kerja Lapangan (PKL) dalam proses perancangan dan penerapan arsitektur cloud-native untuk sistem smartfarming menggunakan AWS adalah dengan menggunakan pendekatan DevOps Lifecycle Model. Metode ini mendukung pengembangan sistem dengan integrasi berkelanjutan (Continuous Integration) dan pengiriman berkelanjutan (Continuous Delivery), memungkinkan setiap proses pengembangan dapat berjalan secara lebih cepat, terstruktur, dan kolaboratif. Tahapan ini mencakup serangkaian siklus yang dilakukan secara berulang-ulang, mulai dari pengembangan hingga monitoring.



Gambar 1.2 DevOps Lifecycle

Pendekatan DevOps memastikan pengembangan sistem cloud-native berjalan fleksibel dengan memanfaatkan layanan AWS untuk efisiensi infrastruktur, penyimpanan data, dan pengelolaan proses IoT secara optimal. Berikut adalah penjelasan tahapan-tahapan metode yang diterapkan dalam pelaksanaan Praktik Kerja Lapangan ini:

1.4.1 Plan

Tahapan ini bertujuan untuk merencanakan kebutuhan sistem dan menetapkan tujuan implementasi arsitektur cloud-native. Aktivitas yang dilakukan meliputi:

1. Identifikasi Kebutuhan Sistem: Menganalisis kebutuhan perangkat IoT, alur data sensor, dan spesifikasi teknologi cloud yang akan digunakan.

2. Perencanaan Arsitektur: Menyusun rencana integrasi perangkat IoT dengan Eclipse mosquitto untuk Broker MQTT dan RDS PostgreSQL untuk manajemen penyimpanan data.

1.4.2 Develop

Pada tahap ini, dilakukan proses pengembangan sistem smartfarming secara iteratif dengan menggunakan teknologi cloud-native. Aktivitas yang dilakukan meliputi:

1. Pengembangan Backend API: Membuat layanan backend menggunakan AdonisJS untuk menangani data sensor yang dikirim oleh perangkat IoT ke cloud.
2. Pengembangan Frontend: Merancang antarmuka sistem menggunakan Angular.js untuk menampilkan data sensor seperti kadar NPK, suhu, kelembaban, dan intensitas cahaya.
3. Manajemen Kode: Menggunakan GitLab sebagai sistem version control untuk pengelolaan kode secara kolaboratif.

1.4.3 Build

Tahap ini melibatkan proses membangun sistem agar dapat diimplementasikan dan diuji dengan cepat menggunakan alat otomatisasi. Aktivitas yang dilakukan adalah:

1. Integrasi Kode dan Dependency: Menggunakan CI/CD Pipeline pada GitLab untuk mengotomatiskan proses build dari kode yang telah dikembangkan.
2. Dockerization: Mengemas backend API dan frontend ke dalam container menggunakan Docker untuk mempermudah deployment di berbagai lingkungan.
3. Validasi Build: Memastikan proses build berjalan tanpa error sebelum dilanjutkan ke tahap testing dan deployment.

1.4.4 Test

Tahap testing dilakukan untuk memeriksa fungsionalitas, performa, dan keamanan sistem sebelum di-deploy. Proses yang dilakukan meliputi:

1. Pengujian Unit dan Integrasi: Menguji backend API dan frontend untuk memastikan data sensor dapat diproses dan ditampilkan dengan benar.

2. Pengujian Unit dan Integrasi: Menguji backend API dan frontend untuk memastikan data sensor dapat diproses dan ditampilkan dengan benar.
3. Pengujian Keamanan: Memastikan API dan komunikasi data IoT aman dari kebocoran atau akses ilegal dengan menerapkan AWS IAM dan *security groups*.

1.4.5 Release

Setelah pengujian selesai dan sistem dinyatakan siap, tahapan ini berfokus pada deployment sistem ke lingkungan produksi. Aktivitas yang dilakukan meliputi:

1. Deploy Sistem di AWS:
 - Backend dan frontend di-deploy di AWS EC2 menggunakan container Docker.
 - penyimpanan file data sensor pada RDS PostgreSQL.
2. Otomatisasi Deployment: Menggunakan GitLab CI/CD untuk mengatur proses deployment berkelanjutan secara otomatis ke cloud AWS.

1.4.6 Operate

Tahap ini berfokus pada pengoperasian sistem cloud-native yang telah di-deploy, meliputi:

1. Monitoring Sistem: Menggunakan Amazon CloudWatch untuk memantau performa aplikasi, jaringan, dan data dari perangkat IoT.
2. Penanganan Log: Merekam aktivitas sistem untuk dianalisis dan menyelesaikan potensi gangguan menggunakan AWS CloudTrail dan AWS Logs.
3. Optimasi Infrastruktur: Melakukan pengaturan skala otomatis (auto-scaling) di EC2 untuk menyesuaikan kebutuhan sumber daya.

1.4.7 Monitor

Tahap terakhir adalah memantau sistem secara berkelanjutan untuk memastikan performa dan stabilitas operasional. Aktivitas pada tahap ini meliputi:

1. Pengumpulan Data Performa: Analisis metrik performa sistem, seperti uptime, response time, dan penggunaan sumber daya.
2. Evaluasi Log Sistem: Mengevaluasi log untuk mendeteksi kesalahan atau gangguan yang mungkin terjadi pada perangkat atau infrastruktur cloud

AWS.

3. Continuous Feedback: Memberikan laporan kepada stakeholder dan merumuskan peningkatan serta pembaruan untuk siklus DevOps berikutnya.