

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pengembangan perangkat lunak merupakan suatu kegiatan yang tidak akan bisa dipisahkan dalam kehidupan masyarakat khususnya di dalam perusahaan. Di setiap perusahaan yang bergerak dalam beragam sektor bisnis tentu memerlukan aplikasi, yaitu system informasi *enterprise* (Inggrid, 2019). Sistem *enterprise* adalah sebuah sistem yang mendukung organisasi untuk menciptakan budaya yang lebih disiplin. Menurut Munawar & Hodijah (2018), sistem informasi perusahaan biasanya dibangun dengan pendekatan monolitik, dimana aplikasi terbungkus dalam paket besar dan ketika diperlukan perubahan pada beberapa bagian kode program, hal ini akan berdampak besar pada kode program lain.

PT. Enerren Technologies yang selanjutnya dalam penulisan akan disebut sebagai Enerren merupakan perusahaan yang bergerak di bidang jasa pembuatan aplikasi berbasis *web* untuk solusi transportasi dan logistik dengan perangkat keras *Global Positioning System* (GPS), diantara aplikasi andalannya adalah *Transportation Management System* (TMS). Secara umum aplikasi ini bertujuan untuk memantau pengiriman barang menggunakan armada kendaraan yang telah dipasang perangkat GPS.

TMS merupakan aplikasi buatan Enerren yang termasuk menggunakan arsitektur monolitik, baik yang dikelola secara internal dengan konsep bisnis *Software as a Service* (SaaS) yaitu sebuah layanan berlangganan bulanan dengan semua layanan baik aplikasi dan basis data disediakan secara penuh oleh Enerren, ataupun yang menggunakan konsep bisnis *on premise* yaitu aplikasi yang instalasinya berada pada server milik pelanggan.

Permasalahan utama yang umumnya terdapat pada aplikasi berarsitektur monolitik seperti TMS adalah proses pengembangan yang memerlukan waktu yang relatif lama disebabkan karena setiap adanya implementasi pada pelanggan baru dengan aplikasi yang sama diperlukan *copy-paste* dari *source* program yang sudah ada lalu dilakukan perubahan pada kode aplikasi tersebut secara manual.

Selain itu, TMS saat ini juga memiliki ketergantungan terhadap komponen pemrograman satu sama lain yang erat sehingga bersifat *tightly coupled*, hal ini tentu membuat aplikasi bersifat kaku dan cukup sulit ketika diperlukan integrasi dengan modul aplikasi Enerren lainnya. Ditambah lagi, saat ini implementasi dilakukan pada banyak *server database* mengakibatkan tingginya kemungkinan terjadi *human error* ketika melakukan perubahan skrip secara berurutan pada setiap *database* TMS.

1.2. Identifikasi Masalah

Dari penjelasan kontekstual di atas, dikemukakan beberapa masalah identifikasi, yaitu:

1. Belum adanya mekanisme di dalam TMS untuk mencegah terjadinya *human error* ketika proses *update* pada *database*.
2. Belum adanya arsitektur aplikasi yang tidak *tightly coupled* dan independen.
3. Belum adanya solusi agar proses *update* versi *database* dapat dipastikan berurutan dan seragam.

Dari uraian identifikasi permasalahan dapat dirumuskan sebagai berikut :

1. Bagaimana cara untuk membuat mekanisme di dalam TMS untuk mencegah terjadinya *human error* ketika proses *update* pada *database*?
2. Bagaimana cara untuk membuat aplikasi agar tidak *tightly coupled* dan independen?
3. Bagaimana cara untuk membuat proses *update* versi *database* secara berurutan dan seragam pada aplikasi TMS?

1.3. Ruang Lingkup dan Batasan Masalah

Ruang lingkup dan batasan masalah dalam tugas akhir ini adalah analisis dan perancangan aplikasi sistem manajemen transportasi dengan menggunakan *Dependency Injection* (DI) pada *framework* aplikasinya dan penggunaan *Code First* (CF) pada perancangan *database*.

1.4. Tujuan Penelitian

Tujuan Penelitian adalah sebagai berikut :

1. Sebagai salah satu syarat untuk mendapatkan gelar sarjana di Universitas Pembangunan Jaya (UPJ).
2. Untuk membuat aplikasi TMS secara *component-based modules*, tidak *tightly coupled* dan independen dengan penerapan *Dependency Injection (DI)* sebagai sarana untuk menyokong alur proses komponen aplikasi.
3. Untuk mengurangi kemungkinan terjadinya *human error* yang sering terjadi pada saat proses *update database*.

1.5. Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut :

1. Menjadi referensi pembuatan aplikasi *component-based modules*, tidak *tightly coupled* dan independen dengan menggunakan *Dependency Injection (DI)* dan komponen secara sistematis di dalam aplikasi.
2. Mengurangi *human error* ketika melakukan *update database* pada server dengan penerapan *Code First (CF)*.

1.6. Sistematika Penulisan

Sistematika penulisan skripsi terdiri atas :

1. Bagian awal skripsi

Bagian awal skripsi terdiri atas halaman sampul depan, judul, halaman pengesahan, halaman pernyataan orinasinalitas, halaman kata pengantar, abstrak, daftar isi, daftar tabel, daftar gambar dan daftar lampiran.

2. Bagian utama skripsi

Bagian utama skripsi / Tugas Akhir terdiri atas 5 bagian diantaranya :

BAB I. Pendahuluan

Pendahuluan pada Bab I meliputi dasar-dasar, perumusan/pendefinisian tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

BAB II. Tinjauan Pustaka

Pada Bab II memuat teori dasar dan tinjauan studi. berisikan uraian mengenai teori-teori dan tinjauan studi sebelumnya yang mendukung skripsi / Tugas Akhir.

BAB III. Metode Penelitian

Bab ini memuat objek penelitian, analisis sistem yang telah berjalan dan analisis kebutuhan dari sistem yang ada.

BAB IV. Hasil dan Analisis Penelitian

Bab ini terdiri atas rangkaian pembahasan dari penelitian yang telah dilakukan. Adapun uraiannya terdiri atas :

1. Analisis Perancangan Sistem
2. Perancangan Diagram Sistem Usulan
3. Perancangan Antar Muka Pengguna
4. Perancangan Implementasi

BAB V. Penutup

Bab terakhir dari bagian utama skripsi ini memuat kesimpulan dan saran atas pembahasan dan hasil penelitian yang telah dilakukan.

3. Bagian akhir skripsi

Pada bagian akhir skripsi / Tugas Akhir terdiri atas daftar pustaka dan lampiran.