

BAB I

PENDAHULUAN

I.1 Latar Belakang Masalah

Berkembangnya teknologi di Indonesia membuat para penggerak usaha logistik mulai mengembangkan sayapnya di area Teknologi dan Informasi. Pada tahun 2017 BPS mencatat bahwa sektor bisnis agregator logistik tumbuh sebesar 8,25%, melampaui PDB nasional sebesar 5,06% [1]. Sedangkan pada tahun 2018, bisnis ini diestimasi tumbuh sebesar 10% [1]. Dengan pertumbuhan 20% per tahun, aplikasi agregator logistik diperkirakan mencapai 7%-8% pada tahun 2021 atau meningkat 2,08 kali lipat [2]. Itulah mengapa banyak perusahaan logistik di tanah air membangun aplikasi agregator logistik [3]. Hal ini menimbulkan efek kepada *developer* aplikasi di Indonesia khususnya yang membangun aplikasi agregator logistik dan para *developer* dituntut untuk cepat dan akurat dalam proses pembuatannya untuk memenuhi kebutuhan pasar yang cepat berubah [4].

Berdasarkan hasil wawancara dengan beberapa *developer* PT. Sangkuriang Internasional serta observasi penulis yang saat ini masih sebagai salah satu *developer* perangkat lunak di PT. Sangkuriang Internasional. Ditemukan kendala dalam membuat perangkat lunak, yaitu pembangunan perangkat lunak yang selalu dimulai dari nol (0) ketika membuat program/aplikasi pada domain yang sama dikarenakan sulitnya penggunaan ulang. Hal ini merupakan suatu masalah yang krusial karena dalam pembangunan aplikasi, *developer* dituntut untuk cepat dan akurat [4]. Berdasarkan hasil observasi terhadap tiga aplikasi agregator logistik terdapat, kesamaan dari ketiga aplikasi tersebut. Kesamaan itu ialah kesamaan fungsionalitas. Hal ini seharusnya dapat digunakan *developer* untuk mempermudah pekerjaannya karena fungsionalitas dasar yang sudah jelas sehingga fungsionalitas dasar tersebut dapat digunakan kembali

apabila membangun perangkat lunak pada domain yang sama khususnya aplikasi agregator logistik.

Berdasarkan hal tersebut, maka solusi yang dapat digunakan adalah membangun sebuah fondasi pada aplikasi agregator logistik. Fondasi tersebut adalah sekumpulan kode yang berupa *Class Library*. *Class Library* apabila diterjemahkan ke dalam Bahasa Indonesia dapat diartikan sebagai perpustakaan kelas. Namun, maksud dari perpustakaan kelas itu sendiri ialah *template* yang terdiri dari kelas-kelas yang dipakai dalam pemrograman berbasis objek (OOP) yang berisi kode program yang dapat digunakan kembali untuk mempermudah pekerjaan yang repetitif [5] [6] . Menurut studi kasus tentang *class library* terdapat terdapat domain-domain yang sudah diteliti, namun belum ada yang mengulas secara khusus tentang *class library* pada domain agregator logistik [7] [8] [9]. Mengacu pada beberapa penelitian, terdapat metode analisis yang cocok untuk pembuatan *class library* yaitu FODA [8] [10] [11] [12].

Berangkat dari masalah-masalah tersebut, maka perlu dilakukan pembangunan *class library* untuk domain manajemen angkutan barang di aplikasi agregator logistik. Dengan menggunakan *class library* diharapkan dapat mempermudah serta mempercepat para *developer* dalam membangun serta menggunakan fungsionalitas dasar pada domain manajemen angkutan barang di aplikasi agregator logistik.

I.1 Perumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, maka dirumuskan masalah pada penelitian ini yaitu bagaimana membangun sebuah *class library* untuk domain manajemen angkutan barang di aplikasi agregator logistik.

I.2 Maksud dan Tujuan

Maksud dari penulisan tugas akhir ini adalah untuk membangun *class library* untuk domain manajemen angkutan barang di aplikasi agregator logistik. Sedangkan tujuan yang ingin dicapai dari penelitian ini adalah:

1. Mempermudah serta mempercepat *developer* dalam membangun aplikasi dan/atau *service* agregator logistik.
2. Menyediakan fungsionalitas dasar yang dapat digunakan kembali dalam membuat aplikasi dan/atau *service* agregator logistik.

I.1 Batasan Masalah

Berdasarkan permasalahan serta maksud dan tujuan yang telah dijelaskan sebelumnya, maka penelitian ini dibatasi dengan masalah sebagai berikut:

1. *Class Library* yang dibangun hanya pada domain manajemen angkutan barang di aplikasi agregator logistik.
2. Bahasa pemrograman yang digunakan untuk pembangunan *class library* adalah C# dengan teknologi .NET CORE.
3. *Class Library* yang dibangun digunakan pada lingkungan Entity Framework.
4. *Class Library* yang dibangun hanya untuk *backend*.
5. Pendekatan analisis dan perancangan menggunakan pendekatan analisis dan perancangan perangkat lunak berorientasi objek (OOP).
6. Metode penelitian yang digunakan adalah *Domain Engineering*.
7. Metode analisis menggunakan FODA (*Feature Oriented Domain Analysis*).

I.2 Metodologi Penelitian

Metodologi penelitian yang digunakan adalah menggunakan metode pengumpulan data dan metode penyelesaian masalah.

I.5.1 Metode Pengumpulan Data

Metode Pengumpulan data yang digunakan dalam penelitian ini antara lain adalah:

1. Metode Studi Literatur

Studi literatur adalah metode pengumpulan data yang dilakukan dengan cara mengumpulkan literatur, jurnal, *paper*, bacaan-bacaan dan website yang berkaitan dengan judul penelitian [13].

1. Metode Observasi

Metode Observasi merupakan metode yang dilakukan dengan cara mengamati objek yang dibutuhkan selama penelitian [13].

2. Metode Wawancara

Metode wawancara adalah metode pengumpulan data yang dilakukan dengan bertanya kepada pihak yang terkait dengan penelitian, dalam hal ini adalah *developer* [13].

I.5.2 Metode Penyelesaian Masalah

Metode penyelesaian masalah dari penelitian ini menggunakan pendekatan *Domain Engineering*. Berikut ini adalah tahapan yang dilakukan:

1. Analisis Domain

Pada penelitian ini domain yang dianalisis adalah domain aplikasi. Dengan menggunakan metode FODA (*Feature Oriented Domain Analysis*), dilakukan analisis terhadap seluruh fungsionalitas yang terdapat pada masing-masing aplikasi. Sehingga didapatkan fungsionalitas apa saja yang akan digunakan untuk bagian selanjutnya. FODA terbagi menjadi dua tahap yaitu *Context Analysis* dan *Domain Modeling* [14].

2. Perancangan Domain

Setelah melakukan analisis terhadap domain, maka pada tahap ini dilakukan perancangan terhadap *class library* yang terdiri dari perancangan kelas dan *package*.

3. Implementasi Domain

Pada tahap ini dilakukan pembangunan terhadap *class library* yang dibangun dan melakukan apa yang telah dirancang sebelumnya. Setelah itu dilakukan integrasi dan dilakukan pengujian untuk mengetahui hasil dari apa yang telah diimplementasikan serta memastikan bahwa apa yang telah dibangun sesuai dengan kebutuhan. Pengujian terhadap *class library* dilakukan menggunakan *unit testing*, *integration testing* dan/atau *user acceptance testing*.



Gambar I.1 – Metode Penyelesaian Masalah

I.1 Sistematika Penulisan

BAB 1 : PENDAHULUAN

Dalam bab ini berisikan latar belakang masalah, identifikasi masalah, batasan masalah, dan sistematika penulisan yang ada pada penelitian ini.

BAB II : LANDASAN TEORI

Dalam bab ini berisikan teori-teori yang digunakan yang mendukung dalam penyelesaian penelitian pada domain agregator logistik.

BAB III : ANALISIS DAN PERANCANGAN

Dalam bab ini berisikan analisis sistem serta perancangan sistem guna membangun *class library* pada domain agregator logistik seperti yang diharapkan.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Dalam bab ini berisikan implementasi dari sistem yang telah dirancang pada bab 3. Implementasi yang dilakukan meliputi, implementasi basis data, perangkat lunak serta keras. Tahapan pengujian berupa *unit testing* dilakukan pada perangkat lunak yang telah diimplementasi.

BAB V : KESIMPULAN DAN SARAN

Dalam bab ini berisikan tentang kesimpulan dan saran dari proses penelitian yang sudah dilakukan.

