

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Vodea merupakan perusahaan yang bergerak di bidang Produksi Digital dan Strategi. Vodea berfokus membantu bisnis dalam mengambil langkah bisnis yang tepat dengan menyediakan inovasi melalui teknologi seperti pengembangan produk sistem aplikasi baik *mobile apps* dan *web apps*. Vodea bergerak pada bidang pengembangan produk digital berbasis teknologi informasi yang dapat membantu kebutuhan bisnis untuk mendukung dalam kelancaran proses bisnis digital mereka.

Kampanye melalui media email masih menjadi salah satu alternatif platform utama untuk meningkatkan pemasaran produk dan membangun *customer engagement* secara personal dengan para pelanggan. Pemasaran melalui media email dapat memberikan informasi kepada pelanggan bahwa bisnis sedang memberlakukan promosi terhadap produk atau layanan jasa yang ditawarkan. Jumlah pengiriman email dari tahun 2017 sampai dengan tahun 2020 mengalami peningkatan  $\pm 3.47\%$  tiap tahunnya dan juga prediksi pada akan ada total 376,4 juta email yang akan dikirim pada tahun 2025 (Johnson, 2021).

*Microservices architecture* merupakan solusi dari masalah arsitektur tradisional yang telah ada sebelumnya. *Microservices architecture* adalah gaya pengembangan aplikasi dengan membaginya menjadi beberapa layanan yang dibatasi oleh fungsional bisnis dan memungkinkan untuk mendistribusikannya terlebih dahulu sebelum layanan lain terselesaikan (Newman, 2015). *Microservices architecture* memungkinkan untuk setiap layanan fokus pada penyelesaian satu masalah, setiap layanan tidak bergantung pada layanan lain dan tidak memiliki ketergantungan terhadap teknologi yang digunakan (Jaramillo et al., 2016). *Microservices architecture* sangat berbeda dengan *Monolith architecture*, yang dimana *Monolith architecture* menempatkan seluruh fungsi bisnis di dalam satu

aplikasi besar (Chris Richardson & Smith, 2016). *Monolithic architecture* dapat memunculkan masalah kompleksitas yang dapat menghambat proses pengembangan dan pemeliharaan aplikasi serta memiliki ketergantungan pada teknologi yang telah digunakan (Mustafin, 2017).

Menerapkan *Microservices architecture* sebagai arsitektur aplikasi memiliki beberapa kekurangan serta tantangan, diantaranya munculnya kesulitan dalam manajemen *dependency* dari setiap layanan mungkin akan menggunakan berbagai macam teknologi untuk menyesuaikan dengan kebutuhan, banyaknya layanan yang akan di distribusi atau integrasi ke dalam lingkungan produksi, memonitor status dari layanan yang berjalan, dan juga tidak adanya aturan pasti dalam memisahkan layanan dari aplikasi yang akan dibangun.

Membangun aplikasi dengan menggunakan *Microservices architecture* memiliki kelemahan karena tidak adanya aturan pasti dalam hal memisahkan layanan yang ada. Oleh karena itu diperlukan suatu pendekatan untuk memudahkan dalam memecahkan setiap modul dari aplikasi ke dalam layanan yang memiliki fungsi spesifik. Dalam buku yang berjudul "*Domain-Driven Design: Tackling Complexity in the Heart of Software, (2003)*", *Domain Driven Design* dapat membantu dalam membatasi dan mendesain setiap layanan mikro sehingga setiap layanan nantinya akan sesuai dengan kebutuhan fungsional bisnis (Evans, 2003). *Domain Driven Design (DDD)* dimulai dengan memodelkan domain bisnis dan membuat model domain. Model domain merupakan model abstrak dari domain bisnis. Proses ini untuk memilih dan mengatur pengetahuan domain, dan menyediakan bahasa yang sama antara *business analyst* dan pengembang (Evans, 2003).

Selain itu, apabila pengembangan aplikasi telah menggunakan berbagai macam bahasa pemrograman ataupun teknologi lainnya dapat menimbulkan kesulitan dalam *deployment* dan *integration*. Hal ini dapat menjadi hambatan untuk pengembang karena diperlukan Langkah tambahan untuk memasang semua *tools* ataupun *dependency* agar layanan yang telah dibuat

dapat dijalankan. *Docker container* hadir sebagai platform untuk membantu pengembang dalam mengatur *dependency* pada *Microservices architecture* (Jaramillo et al., 2016). *Docker* merupakan teknologi platform *container* yang memungkinkan untuk membangun, menguji dan menyebarkan aplikasi terdistribusi dalam *environment* yang terisolasi (Jaramillo et al., 2016).

Untuk manajemen *container* pada lingkungan produksi, dibutuhkan sebuah *container orchestration* agar memudahkan dalam pendistribusian aplikasi dan juga *scaling* aplikasi apabila beban kerja yang diberikan sudah terlalu banyak. *Kubernetes (K8s)* adalah sebuah platform *container orchestration* yang digunakan sebagai alat untuk manajemen *container* dengan *automation*. *Kubernetes* dapat membantu dalam hal penjadwalan, pendistribusian beban kerja, dan *scaling* (Vohra, 2016).

Salah satu kelemahan implementasi *Microservices architecture* yaitu proses *integration* dan *deployment* layanan ke lingkungan produksi bergerak sangat cepat, karena tidak diperlukan menunggu layanan lainnya selesai, apabila layanan yang dibuat telah siap, maka layanan langsung dapat di kirim ke lingkungan produksi, proses ini dapat menyulitkan karena banyaknya layanan yang perlu didistribusikan ke lingkungan produksi. *DevOps* dapat membantu untuk mengotomatiskan proses antara *development team* dan *operation team* agar dapat melakukan proses *build*, *testing*, sampai *release* dengan otomatis (Kang et al., 2016). Selain itu *DevOps* mengurangi campur tangan manusia dalam pengujian dan pendistribusian agar seluruh proses pengujian dan pendistribusian dapat menjadi lebih efektif.

Berdasarkan uraian permasalahan diatas, aplikasi Kampanye Email Marketing ini perlu dibangun menggunakan *Microservices architecture* dan pendekatan *Domain Driven Design* untuk memecahkan modul menjadi layanan yang mampu bekerja secara independen. Selain itu dibutuhkan pula *Docker Container*, *Kubernetes*, dan *DevOps* untuk menutupi kekurangan yang ada pada *Microservices architecture*.

## 1.2 Identifikasi Masalah

Dengan meninjau permasalahan yang telah dikemukakan di atas, identifikasi masalah yang akan dibahas pada penelitian ini sebagai berikut:

1. Bagaimana membangun aplikasi Kampanye Email Marketing dengan menggunakan *Microservices Architecture* dan pendekatan *Domain Driven Design*?
2. Bagaimana menggunakan *Docker Container* dan *Kubernetes* sebagai platform untuk memajemen setiap layanan dari aplikasi Kampanye Email Marketing?
3. Bagaimana menerapkan pendekatan *DevOps* untuk otomatisasi dalam integrasi dan distribusi ke dalam lingkungan produksi pada aplikasi Kampanye Email Marketing?

## 1.3 Manfaat Tugas Akhir

Adapun manfaat yang didapat dari tugas akhir ini, antara lain sebagai berikut:

1. Aplikasi Kampanye Email Marketing dapat membantu para pemangku kepentingan untuk melakukan kegiatan promosi melalui media digital dan laporan dengan bantuan satu aplikasi.
2. Mengenalkan penggunaan *Microservices Architecture*, *Domain Driven Design*, *Docker Container*, *Kubernetes*, dan *DevOps* untuk *Software Developer*.

## 1.4 Tujuan Tugas Akhir

Adapun tujuan penelitian tugas akhir ini adalah:

1. Membangun aplikasi Kampanye Email Marketing dengan menggunakan *Microservices Architecture* dan *Domain Driven Design* untuk mendesain batasan fungsi bisnis yang besar menjadi layanan -layanan kecil yang independen untuk diterapkan di aplikasi Kampanye Email Marketing di PT. Vodea Dwi Karsa.
2. Menggunakan *Docker Container* dan *Kubernetes* sebagai platform untuk membantu mengatur dan mengelola setiap layanan pada aplikasi Kampanye Email Marketing.

3. Mengimplementasikan pendekatan *DevOps* untuk mengurangi waktu proses integrasi dan distribusi ke dalam lingkungan produksi.

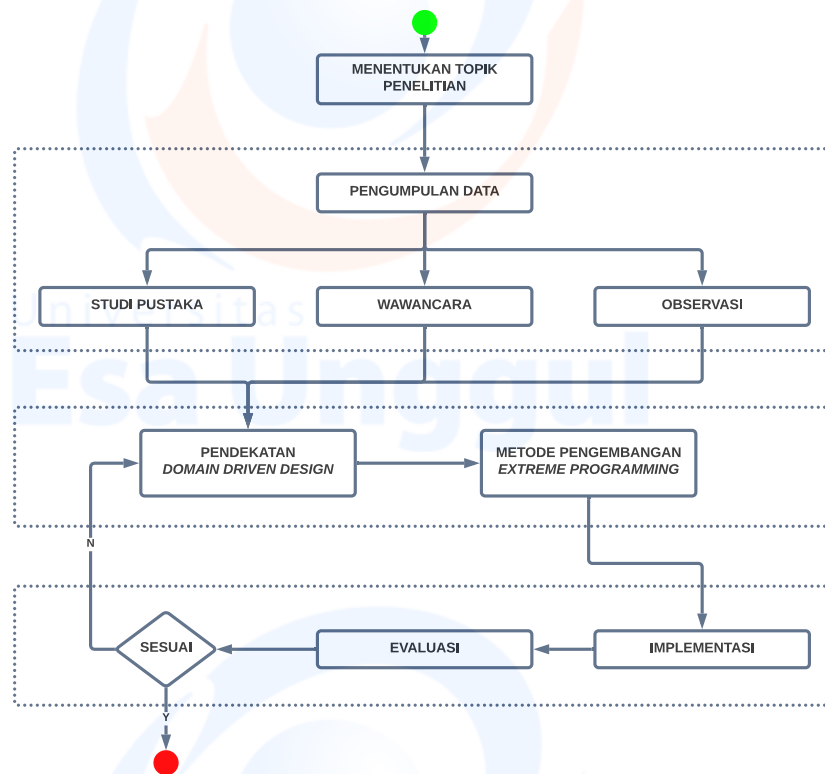
### 1.5 Lingkup Tugas Akhir

Agar permasalahan yang tercakup dalam latar belakang tidak menyimpang dan terlalu jauh, maka dibuat ruang lingkup dari penelitian ini, diantaranya:

1. Pembahasan dan pembangunan hanya pada aplikasi Kampanye Email Marketing.
2. Aplikasi ini tidak mencakup untuk mengatasi masalah email yang terkirim masuk ke dalam folder *spam*.
3. Pembangunan menggunakan kerangka kerja Nest.js sebagai *Back-end* dan React.js sebagai *Front-end*.
4. Pembahasan penelitian ini tidak mencakup perbandingan mengenai *Microservice Architecture* dan *Service Oriented Architecture*.

### 1.6 Kerangka Berpikir

Kerangka berpikir merupakan uraian atau gambaran mengenai kerangka konsep pemecahan masalah yang telah didefinisikan atau dirumuskan seperti gambar dibawah ini:



Gambar 1.1: Kerangka Berpikir

### 1.6.1 Pengumpulan Data

Pada tahapan ini dilakukan pengumpulan data untuk mendukung dan menjadi tujuan penelitian ini, pengumpulan data dilakukan dengan melakukan beberapa tahapan, diantaranya sebagai berikut:

1. Studi Pustaka

Pada tahapan ini dilakukan pengumpulan data dan informasi dari beberapa karya ilmiah maupun non-ilmiah yang masih berkaitan dengan penelitian untuk dijadikan pedoman dari setiap permasalahan yang ada.

2. Wawancara

Kegiatan wawancara dilakukan dengan *stakeholder* perusahaan untuk menggali kebutuhan fungsional sampai dengan kebutuhan non-fungsional.

3. Observasi



Pengamatan dilakukan dengan mengamati sistem yang sudah ada beberapa perusahaan yang menyediakan aplikasi sejenis untuk memperoleh informasi mengenai fitur, kelebihan, serta kekurangan untuk menjadi masukan kepada *stakeholder*.

### 1.6.2 *Domain Driven Design*

*Domain Driven Design* digunakan untuk merancang pengembangan aplikasi perangkat lunak yang kompleks dengan menerapkan *bounded context* disetiap domain bisnis. *Domain Driven Design* membantu memecahkan setiap bisnis proses berdasarkan fungsinya untuk kemudian dijadikan sebagai layanan dari *Microservices Architecture*.

Adapun tahapan yang akan dilakukan dalam menggunakan pendekatan *Domain Driven Design*, yaitu:

1. Menganalisis fungsional proses bisnis dari aplikasi yang akan dibangun.
2. Mengelompokan fungsional sistem untuk mengetahui fungsi-fungsi yang memiliki persamaan antara proses bisnis.
3. Mengidentifikasi hubungan fungsional suatu sistem untuk mengetahui keterhubungan antara satu proses bisnis dengan proses bisnis yang lain.
4. Mendefinisikan fungsional menjadi *microservices*.

### 1.6.3 *Extreme Programming*

Metode pengembangan *Extreme Programming* membagi tahap pengembangan menjadi 4 bagian, yaitu:

#### 1.6.3.1 *Perencanaan (Planning)*

Pada tahap ini akan dilakukan perencanaan pembangunan aplikasi berdasarkan data yang telah dikumpulkan pada kegiatan indentifikasi masalah serta

menyusun cara penyelesaian masalah yang telah berdasarkan data yang telah dikumpulkan.

#### **1.6.3.2 Perancangan (*Design*)**

Pada tahap ini akan dilakukan perancangan aplikasi dengan menggunakan metode *Domain Driven Design* dan menggambarannya dalam bentuk *Unified Model Language*.

#### **1.6.3.3 Pengkodean (*Coding*)**

Pada tahap ini akan dilakukan pembuatan aplikasi dan menyusun struktur basis data berdasarkan desain yang telah disepakati.

#### **1.6.3.4 Pengujian (*Test*)**

Pada tahap ini akan dilakukan pengujian dengan metode *Black Box Testing* untuk memastikan sistem tersebut telah berjalan sebagaimana mestinya. Pengujian dilakukan secara otomatis pada saat proses *Continuous Integration* dari *DevOps*.

#### **1.6.4 Implementasi**

Pada tahapan ini dilakukan pengimplementasian atau pengiriman aplikasi ke dalam lingkungan produksi. Proses ini akan secara otomatis dilakukan pada *Continuous Deployment* apabila proses *Continuous Integration* pada tahap sebelumnya berhasil.

### **1.7 Sistematika Penulisan Tugas Akhir**

Sistematika penulisan proposal tugas akhir ini dibagi menjadi 5 (lima) bab dan mengenai isi-isi bab diuraikan sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini merupakan bagian yang berisi mengenai latar belakang, identifikasi masalah, tujuan, manfaat, ruang lingkup serta sistematika penulisan tugas akhir yang diperlukan dalam menyusun Tugas Akhir.



## **BAB II LANDASAN TEORI**

Bab ini berisikan uraian teoritis dan penelitian terdahulu yang menjadi dasar dan mendukung dalam argumentasi dan rancangan tugas akhir ini.

## **BAB III METODOLOGI PENELITIAN**

Bab ini berisikan rencana dan objek penelitian, metode yang digunakan, data yang diperlukan, sumber data, teknik pengumpulan data, dan teknik analisis data.

## **BAB IV HASIL DAN PEMBAHASAN**

Bab ini berisikan tentang hasil data penelitian, pembahasan hasil analisis, dan penarikan kesimpulan dari penelitian yang dilakukan berdasarkan kerangka pemikiran yang telah dibuat.

## **BAB V KESIMPULAN DAN SARAN**

Bab ini berisikan kesimpulan mengenai hasil tentang dari penelitian dan perancangan yang telah dilakukan serta saran mengenai sistem yang akan dikembangkan dimasa yang akan datang.