

**ABSTRAK**

Judul : Rancang Bangun Aplikasi Kampanye Email Marketing Berbasis Arsitektur *Microservices* dan *Container* (Studi Kasus: PT. Vodea Dwi Karsa)

Nama : Yudi Hertanto

Program Studi : Teknik Informatika

Aplikasi kampanye email marketing yang dibutuhkan PT. Vodea Dwi Karsa berguna untuk melakukan kampanye marketing melalui email dengan fungsi utama mengirimkan dan melakukan analisa terhadap keberhasilan email marketing yang dikirimkan ke konsumen terdaftar. Saat ini pengembangan aplikasi di PT. Vodea Dwi Karsa masih menggunakan *Monolith Architecture*. Aplikasi yang telah menjadi besar atau kompleks namun masih menggunakan *Monolithic Architecture* dapat memberikan masalah serius pada proses pengembangan dan pemeliharaan kedepan. *Microservices Architecture* hadir sebagai solusi dari masalah pada *Monolithic Architecture*. *Microservices Architecture* merupakan arsitektur dalam pengembangan perangkat lunak yang dimana sebuah sistem aplikasi merupakan susunan dari banyak layanan kecil dan independen namun tetap dapat saling berinteraksi satu sama lain melalui *light communication*. *Microservices Architecture* dibangun dari komponen kecil dan independen yang mampu dibuat dan didistribusikan ke dalam lingkungan produksi tanpa harus menunggu layanan lainnya selesai dan juga mampu menskalakan aplikasi di level layanan yang memberikan tingkat efisiensi yang sangat tinggi pada pemakaian sumber daya server. Dalam implementasinya *Microservices Architecture* digunakan metode *Domain Driven Design* sebagai panduan dalam membatasi fungsi dari setiap layanan. Hasil dari pembagian sistem dengan metode *Domain Driven Design* adalah *API Gateway*, *Account Service*, *Audience Service*, *Campaign Service*, *Email Service* dan *Public Service*. *Container (Docker)* merupakan virtualisasi sistem operasi yang membungkus suatu aplikasi beserta *dependency*-nya. *Container* berbeda dengan *Virtual Machine*, dimana proses yang dilakukan oleh *Container* terisolir dari *Host OS* ataupun *container* lainnya. *Container Orchestration (Kubernetes)* digunakan sebagai platform *container* untuk manajemen *workloads* serta menyediakan konfigurasi otomatisasi secara deklaratif. *DevOps* diterapkan sebagai langkah dalam meminimalisir waktu integrasi dan distribusi ke dalam lingkungan produksi.

**Kata Kunci:** *Microservices*, *Domain Driven Design*, *Container*, *Docker*, *Container Orchestration*, *Kubernetes*, *DevOps*.

**ABSTRACT**

*Title : Design And Build Email Marketing Campaign Application Based On Microservices Architecture And Container Technology (Case Study: PT. Vodea Dwi Karsa)*

*Name : Yudi Hertanto*

*Study Program : Computer Science*

*Marketing Email Campaign application needed by PT. Vodea Dwi Karsa is useful for managing campaigns via email with the main function of sending and analyzing the success of campaigns sent to registered consumers. Applications that have been built at PT. Vodea Dwi Karsa still uses Monolith Architecture. The Applications that have become large or complex but still use Monolithic Architecture can give serious problems for future development and maintenance processes. Microservices Architecture comes as a solution in Monolithic Architecture. Microservices Architecture is a style in software development where an application system is an arrangement of many small and independent services but can still interact with each other through light communication. Microservices Architecture is built from small and independent components that can be created and distributed into a production environment without having to wait for other services to be completed and also Microservices Architecture is able to scale applications at the service level which provides a very high level of efficiency in server resource usage. In implementing Microservices Architecture, the Domain Driven Design method is used as a guide in limiting the function of each service. The results of sharing the system using the Domain Driven Design method are API Gateway, Account Service, Audience Service, Campaign Service, Email Service and Public Service. Container (Docker) is an operating system virtualization that wraps an application and its dependencies. Containers are different from Virtual Machines, where the process carried out by the Container is isolated from the Host OS or other containers. Container Orchestration (Kubernetes) is used as a container platform for managing workloads as well as providing declarative automation configurations. DevOps is implemented as a step in minimizing integration and distribution time into the production environment.*

**Keyword:** *Microservices, Domain Driven Design, Container, Docker, Container Orchestration, Kubernetes, DevOps.*