

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang semakin berkembang, pengelolaan layanan aplikasi secara efisien dan skalabel menjadi kebutuhan penting bagi banyak organisasi. Aplikasi Netpro yang dirancang untuk memberikan kemudahan dan efisiensi dalam pengelolaan *Wi-Fi (Wireless Fidelity)* di berbagai jenis usaha saat ini menghadapi berbagai tantangan dalam memastikan kualitas layanan tetap optimal dan sesuai dengan kebutuhan pengguna. Salah satu tantangan utama adalah tingginya penggunaan memori ketika banyak layanan berjalan bersamaan. Selain itu, menjalankan berbagai layanan secara bersamaan sering menimbulkan konflik konfigurasi, seperti tabrakan *port* yang menyulitkan pengelolaan sistem. Di sisi lain, proses *deployment* aplikasi yang masih manual dan kompleks juga menghambat konsistensi lingkungan aplikasi di berbagai server serta memperlambat waktu pengembangan dan pemeliharaan.

Untuk mengatasi permasalahan tersebut, sistem manajemen layanan berbasis *Docker* menawarkan solusi yang inovatif dan relevan. *Docker* memungkinkan aplikasi dan dependensinya dikemas dalam *container* yang memungkinkan penyimpanan terisolasi, sehingga sumber daya dapat dikelola lebih efisien serta dapat menjalankan berbagai layanan secara bersamaan tanpa konflik konfigurasi. Penggunaan *Docker* juga terbukti mampu meningkatkan efisiensi migrasi dan *deployment* aplikasi. Misalnya, sistem migrasi berbasis *Docker* seperti *Sledge* dapat mengurangi *overhead* migrasi hingga 57%, waktu migrasi gambar *container* sebesar 55%, dan *downtime* hingga 70%, menjadikannya solusi ideal untuk *load balancing* dan pemeliharaan *host* dalam skala besar [7]. Selain itu, penerapan *Docker* dalam proses *Continuous Integration/Continuous Delivery (CI/CD)* telah mempercepat waktu *deployment* rata-rata menjadi hanya 17,9 detik, sehingga mengurangi *downtime* dan meningkatkan produktivitas pengembang [1]. Kombinasi *Docker* dan *Kubernetes* juga memberikan efisiensi biaya operasional hingga 44,44% dengan elastisitas tinggi untuk menangani beban dinamis [26].

Data ini menunjukkan bahwa teknologi *Docker* telah menjadi pilihan strategis untuk meningkatkan efisiensi dan skalabilitas aplikasi. Oleh karena itu, implementasi sistem manajemen layanan berbasis *Docker* pada aplikasi Netpro diharapkan mampu mengatasi permasalahan yang ada, sekaligus mendukung pengelolaan layanan yang lebih modern, fleksibel, dan efisien.

1.2 Rumusan Masalah dan Solusi

Berdasarkan latar belakang yang telah dijelaskan, beberapa permasalahan dapat dirumuskan sebagai berikut:

1. Bagaimana cara mengelola penggunaan memori pada aplikasi NetPro agar terhindar dari penggunaan sumber daya yang berlebihan dan memastikan aplikasi tetap berjalan stabil meskipun banyak layanan berjalan secara bersamaan?
2. Bagaimana cara menjalankan layanan berbeda secara bersamaan dalam sistem aplikasi NetPro tanpa terjadi konflik konfigurasi seperti tabrakan port?
3. Bagaimana cara memanfaatkan teknologi *Docker* untuk menyederhanakan proses *deployment* aplikasi NetPro dan memastikan lingkungan aplikasi tetap konsisten di berbagai server?

Untuk menjawab permasalahan di atas, berikut adalah solusi yang ditawarkan:

1. Pengelolaan Penggunaan Memori dengan Docker

Docker memungkinkan pengelolaan sumber daya yang lebih terkontrol dengan cara membatasi memori yang dapat digunakan oleh setiap *container*. Dengan menggunakan fitur pembatasan memori, administrator dapat menentukan batas maksimum memori yang diperbolehkan untuk setiap *container*. Hal ini memastikan bahwa tidak ada layanan yang menghabiskan sumber daya secara berlebihan, sehingga memori tetap terjaga untuk layanan lainnya.

2. Penerapan *Docker* untuk Menjalankan Layanan Berbeda Secara Bersamaan

Dengan teknologi *Docker*, berbagai layanan dapat dijalankan secara bersamaan dalam *container* yang terisolasi. Sebagai contoh, layanan seperti *Nginx* dan *Apache* dapat berjalan dalam *container* terpisah dengan konfigurasi masing-masing, termasuk pengaturan *port* yang berbeda. Hal ini memungkinkan aplikasi NetPro untuk menjalankan berbagai layanan secara bersamaan tanpa konflik, seperti tabrakan *port* yang umum terjadi pada sistem berbasis *host Linux*. *Docker Compose* mempermudah pengelolaan layanan ini dengan mendefinisikan pengaturan untuk setiap *container* dalam satu file.

3. Penyederhanaan *Deployment* dengan *Docker Compose*

Docker Compose memungkinkan semua dependensi aplikasi NetPro didefinisikan dalam satu file *YAML*, termasuk konfigurasi layanan seperti *database*, web server, dan aplikasi utama. Proses *deployment* menjadi lebih sederhana hanya dengan menarik *Image-Docker* dari *Docker Hub* dan menjalankan *container*. Pendekatan ini memastikan lingkungan aplikasi tetap konsisten di berbagai server, menghilangkan kebutuhan untuk instalasi manual, dan menghemat waktu serta usaha dalam proses *deploy*.

1.3 Tujuan

Alasan utama pekerjaan ini dilakukan adalah untuk menyelesaikan berbagai permasalahan yang dihadapi aplikasi NetPro terkait pengelolaan layanan dengan memanfaatkan teknologi *Docker*. Berikut rincian tujuan yang hendak dicapai:

1. Mengelola Penggunaan Memori Secara Efisien

Menerapkan teknologi *Docker* untuk membatasi penggunaan memori pada setiap *container*, sehingga mencegah layanan tertentu menghabiskan sumber daya secara berlebihan. Pendekatan ini memastikan stabilitas aplikasi NetPro meskipun banyak layanan berjalan secara bersamaan, dengan memanfaatkan isolasi *container* untuk pengelolaan sumber daya yang lebih terkontrol.

2. Mengoperasikan Layanan Berbeda Secara Bersamaan dalam Satu Sistem

Teknologi *Docker* memberikan kemampuan untuk menjalankan beberapa layanan dengan konfigurasi yang berbeda, seperti *Nginx* dan *Apache*, secara bersamaan tanpa saling mengganggu. Dengan isolasi *container*, setiap layanan dapat dijalankan dengan pengaturan *port* masing-masing, sehingga potensi tabrakan *port* dapat dihindari. Pendekatan ini memastikan fleksibilitas dan efisiensi dalam mengelola berbagai layanan tanpa perlu menghentikan layanan lain yang sedang berjalan.

3. Mengurangi Kompleksitas *Deployment* Aplikasi

Dengan menggunakan teknologi *Docker*, proses *deploy* aplikasi menjadi lebih sederhana karena hanya membutuhkan langkah menarik *Image-Docker* dari *Docker Hub* dan menjalankan *container*. Semua dependensi aplikasi sudah termasuk dalam *Image-Docker*, sehingga tidak perlu instalasi *manual* pada server. Pendekatan ini menghemat waktu dan mempermudah proses *deploy*, asalkan *Docker* telah terinstal di server.

1.4 Batasan Masalah

1. Ruang Lingkup Proyek Akhir

Proyek Akhir ini fokus pada perancangan dan implementasi sistem manajemen layanan berbasis *Docker* pada aplikasi NetPro untuk mengatasi permasalahan ketidaksesuaian lingkungan, keterbatasan skalabilitas, dan kompleksitas pengelolaan layanan. Lingkup pekerjaan meliputi:

- a. Analisis kebutuhan sistem aplikasi NetPro.
- b. Perancangan arsitektur berbasis *Docker* yang mencakup *Docker Compose*, *Dockerfile*, *DockerHub*, dan *Nginx*.
- c. Implementasi solusi berbasis *Docker* untuk memastikan efisiensi penggunaan memori, skalabilitas, dan efisiensi operasional aplikasi.
- d. Pengujian dan evaluasi sistem yang telah diimplementasikan.

2. Ruang Lingkup dan Constraint Produk

- a. Solusi yang dibangun menggunakan *Docker* dengan fokus pada *Docker Compose*, *Dockerfile*, *DockerHub*, dan *Nginx* sebagai *reverse proxy*. Tidak mencakup penerapan teknologi lain seperti *Kubernetes* atau *CI/CD*.
- b. Menggunakan teknologi *Docker* untuk kontainerisasi aplikasi NetPro dengan tujuan meningkatkan efisiensi penggunaan sumber daya dan skalabilitas, tidak menggunakan teknologi *container* lainnya seperti *Podman* atau *LXC (Linux Containers)*.
- c. Tidak mencakup pengembangan ulang kode aplikasi NetPro, hanya memastikan kompatibilitas dalam lingkungan *Docker*.
- d. Fokus hanya pada pengaturan konfigurasi dan tidak mencakup modifikasi algoritma aplikasi utama.

