

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, perusahaan-perusahaan menghadapi tuntutan untuk memperkuat infrastruktur teknologi informasi mereka agar dapat mengelola data dengan lebih efisien, aman, dan terukur. Salah satu elemen krusial dalam infrastruktur teknologi informasi ini adalah *data center*. Dalam konteks ini, PT. Pelayaran Nasional Indonesia (PELNI) memiliki *data center* yang menyimpan dan mengelola berbagai data penting, termasuk informasi pelayaran, jadwal kapal, data penumpang, dan logistik. Oleh karena itu, pengelolaan dan pemantauan *data center* yang efektif sangat penting untuk meminimalkan risiko *downtime* dan memastikan performa yang optimal.

Dengan memanfaatkan arsitektur *microservices* dan *platform* orkestrasi kontainer seperti Kubernetes, PT. Pelayaran Nasional Indonesia (PELNI) telah memperbaharui sistem manajemen pusat data mereka ke tingkat yang lebih tinggi. Arsitektur *microservices* memungkinkan aplikasi DCIM untuk dibangun sebagai serangkaian layanan terpisah, yang memfasilitasi pengembangan, implementasi, dan perubahan dengan lebih cepat dan fleksibel. Menurut Tanuwijaya dkk. (2021) "Arsitektur *microservices* berarti setiap layanan beroperasi secara independen, sehingga sistem menjadi lebih tahan terhadap kegagalan layanan individual. Dengan penerapan *microservices*, setiap layanan memiliki infrastruktur sendiri dan bisa berjalan di server yang berbeda. Akibatnya, jika satu layanan mengalami gangguan atau *server down*, layanan lain tetap bisa diakses oleh pengguna. Mengubah sistem dari arsitektur monolitik ke *microservices* memungkinkan pemenuhan kebutuhan skalabilitas untuk penggunaan sumber daya masing-masing layanan, serta mempermudah pemeliharaan dan pengujian karena setiap layanan terpisah satu sama lain." [1]

Dengan Kubernetes sebagai *platform* orkestrasi, PT. Pelayaran Nasional Indonesia (PELNI) dapat mengoptimalkan manajemen kontainer secara terpusat, penskalaan otomatis, dan alokasi sumber daya secara dinamis. Hal ini memungkinkan perusahaan untuk mendukung operasi DCIM dengan lebih baik, meningkatkan efisiensi operasional, serta responsif terhadap perubahan lingkungan dan kebutuhan bisnis yang terus berkembang. Dengan demikian, investasi dalam teknologi ini tidak hanya

meningkatkan efektivitas operasional, tetapi juga memungkinkan PT. Pelayaran Nasional Indonesia (PELNI) untuk tetap kompetitif di pasar yang terus berubah dan menuntut.

Menurut Huda dan Kusumawardani (2022) " Kubernetes dirancang untuk mengotomatisasi penyebaran, penskalaan, dan pengoperasian aplikasi dalam *container*. Teknologi Kubernetes memungkinkan otomatisasi *container* diimplementasikan sesuai dengan jumlah pengguna yang mengaksesnya secara bersamaan. Tiga alasan utama menggunakan Kubernetes untuk pengelolaan server adalah ketersediaan, skalabilitas, dan keandalan." [2]

Dalam mengelola infrastruktur data center yang kompleks, PT. Pelayaran Nasional Indonesia (PELNI) membutuhkan sistem pemantauan yang canggih dan terintegrasi untuk memantau kesehatan dan kinerja infrastruktur mereka secara *real-time*. Sistem pemantauan ini harus mampu mendeteksi perubahan atau masalah dengan cepat, sehingga PELNI dapat merespons dengan segera untuk mencegah terjadinya gangguan yang lebih serius. Pemantauan yang efektif akan mengurangi risiko downtime, yang dapat berdampak negatif pada operasional perusahaan dan kepuasan pelanggan.

Salah satu solusi yang dapat diimplementasikan adalah pengembangan Dashboard Monitoring Data Center Infrastructure Management (DCIM) yang terintegrasi dengan Grafana, Alert Manager, dan Bot Telegram menggunakan arsitektur Kubernetes. Grafana adalah platform *open-source* untuk memantau dan memvisualisasikan data kinerja infrastruktur secara real-time. Dengan menggunakan Grafana, PELNI dapat membuat dashboard yang informatif dan interaktif untuk memantau berbagai metrik penting dari data center mereka. Alert Manager akan berperan dalam mengelola notifikasi dan peringatan jika terjadi anomali atau masalah pada infrastruktur. Integrasi dengan Bot Telegram memungkinkan pengiriman notifikasi instan kepada tim teknis PELNI melalui aplikasi Telegram, sehingga mereka dapat segera mengambil tindakan yang diperlukan. Dengan demikian, tim teknis dapat merespons masalah dengan cepat dan efisien, mengurangi risiko downtime dan memastikan ketersediaan sistem yang optimal.

Melalui proyek akhir ini, diharapkan akan menghasilkan sebuah dashboard monitoring yang tidak hanya dapat memberikan informasi real-time tentang kesehatan dan kinerja infrastruktur data center PT. Pelayaran Nasional Indonesia (PELNI), tetapi

juga memiliki kemampuan untuk memberikan notifikasi instan melalui Bot Telegram jika terjadi perubahan signifikan atau masalah dalam infrastruktur. Dengan demikian, proyek ini diharapkan dapat meningkatkan efisiensi, ketersediaan, dan keandalan operasional infrastruktur data center PELNI, mendukung operasional perusahaan secara keseluruhan, dan pada akhirnya meningkatkan kepuasan pelanggan PELNI.

1.2 Tujuan dan Manfaat

Adapun tujuan dari penulisan Proyek Akhir ini, sebagai berikut.

1. Merancang *dashboard* untuk memberikan informasi *real-time* tentang kinerja dan ketersediaan sistem.
2. Mengintegrasikan Grafana dan *alert manager* dengan telegram untuk memberikan notifikasi instan kepada tim operasional ketika terjadi perubahan signifikan dalam kinerja atau keadaan sistem.
3. Mengimplementasikan *dashboard monitoring* untuk secara efektif memantau kesehatan dan kinerja infrastruktur *data center* PT. Pelayaran Nasional Indonesia (PELNI).

Manfaat dari penulisan Proyek Akhir ini, sebagai berikut.

1. Memungkinkan PT. Pelayaran Nasional Indonesia (PELNI) untuk memantau secara proaktif ketersediaan layanan mereka, sehingga dapat mengurangi *downtime* dan memastikan kontinuitas operasional.
2. Mempermudah tim operasional PT. Pelayaran Nasional Indonesia (PELNI) dalam memantau dan mengelola infrastruktur *data center* mereka, sehingga meningkatkan efisiensi operasional dan mengurangi waktu penanganan masalah.
3. Memungkinkan tim operasional untuk merespon dengan cepat terhadap peringatan atau pemberitahuan yang diterima melalui telegram, Sehingga dapat mengambil langkah-langkah yang diperlukan untuk mengatasi masalah dengan lebih efisien.

1.3 Rumusan Masalah

Adapun rumusan masalah dari Proyek Akhir ini, sebagai berikut.

1. Apa saja fitur-fitur yang diperlukan dalam merancang *dashboard* untuk memberikan informasi *real-time* tentang kinerja dan ketersediaan sistem?
2. Bagaimana sistem *alert manager* pada Grafana dapat diintegrasikan ke dalam bot telegram agar memberikan notifikasi instan kepada tim operasional ketika terjadi perubahan signifikan dalam kinerja atau keadaan sistem?
3. Bagaimana mengukur efektivitas *dashboard monitoring* dalam membantu PT. Pelayaran Nasional Indonesia (PELNI) memantau kesehatan dan kinerja infrastruktur *data center* mereka, serta memperbaiki atau meningkatkan *dashboard* berdasarkan hasil evaluasi kinerja?

1.4 Batasan Masalah

Adapun batasan masalah dari Proyek Akhir ini, sebagai berikut.

1. Desain dan implementasi *dashboard monitoring* untuk infrastruktur Data Center PT. Pelayaran Nasional Indonesia (PELNI). Ini mencakup semua komponen yang relevan dengan manajemen infrastruktur *data center*, termasuk server, jaringan, penyimpanan, dan aplikasi terkait.
2. Penggunaan arsitektur *microservices* untuk mendesain dan membangun *dashboard monitoring*. Ini mencakup pemilihan teknologi, pola desain, dan implementasi layanan mikro yang diperlukan untuk mendukung fleksibilitas, skalabilitas, dan isolasi antar komponen.
3. Penggunaan Kubernetes sebagai *platform* orkestrasi untuk menjalankan dan mengelola layanan *micro* yang digunakan dalam *dashboard monitoring*. Ini akan mempertimbangkan konfigurasi, manajemen, dan otomatisasi Kubernetes dalam konteks infrastruktur *data center* PT. Pelayaran Nasional Indonesia (PELNI).
4. Mempertimbangkan kebutuhan dan tantangan khusus yang dihadapi oleh PT. Pelayaran Nasional Indonesia (PELNI) dalam manajemen infrastruktur *data center* mereka. Ini termasuk pertimbangan seperti skala operasional, kebutuhan bisnis, dan regulasi industri yang berlaku.

1.5 Metodologi

Adapun metodologi pada penelitian Proyek Akhir ini, sebagai berikut.

1. Studi Literatur

Tahapan awal yang melibatkan penelitian dan analisis terhadap literatur-literatur dan kajian-kajian yang berkaitan dengan permasalahan yang ada pada penelitian Proyek Akhir ini, baik berupa buku, e-journal maupun sumber informasi lain yang relevan dengan topik penelitian Proyek Akhir.

2. Perancangan Sistem

Perancangan sistem dilakukan secara menyeluruh. Ini meliputi perancangan antarmuka pengguna yang intuitif dan responsif, perancangan struktur *database* untuk menyimpan metrik dan log yang dikumpulkan, serta perancangan arsitektur teknis yang mencakup integrasi Prometheus untuk pengumpulan data, Grafana untuk visualisasi, dan Telegram untuk notifikasi *alert*. Rancangan sistem ini akan menjadi panduan dalam tahap pengembangan proyek.

3. Implementasi dan Pengembangan

Tahap ini melibatkan implementasi rancangan sistem ke dalam kode nyata menggunakan *platform* dan teknologi yang sesuai, seperti Kubernetes untuk orkestrasi kontainer, Prometheus untuk *monitoring*, dan Grafana untuk *dashboard*. Pengembangan juga mencakup pemodelan *database*, integrasi Prometheus dengan sumber data yang relevan, pengembangan fungsi *alert* di Grafana, dan konfigurasi bot Telegram untuk notifikasi.

4. Pengujian dan Evaluasi

Pengujian dilakukan untuk memastikan bahwa *dashboard monitoring* DCIM berfungsi sesuai perancangan. Pengujian mencakup kinerja sistem, verifikasi data Prometheus, tampilan dan fungsionalitas Grafana, serta keandalan notifikasi Telegram. Evaluasi menilai kesesuaian solusi dengan tujuan proyek dan mengidentifikasi area yang perlu perbaikan.

1.6 Sistematika Penulisan

Dalam penulisan Proyek Akhir terdiri atas lima bab, dengan keterangan sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisi latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian, serta sistematika penulisan.

BAB II DASAR TEORI

Pada bab ini membahas tentang teori pendukung yang menjadi dasar pengetahuan yang digunakan dalam menyusun Proyek Akhir. Ini mencakup konsep-konsep dasar tentang pemanfaatan Kubernetes untuk orkestrasi *container*, *monitoring* infrastruktur *data center*, Prometheus untuk pengumpulan metrik, penggunaan Grafana untuk visualisasi data, dan integrasi dengan Telegram untuk notifikasi *alert*.

BAB III PERENCANAAN DAN MODEL SISTEM

Pada bab ini membahas tentang blok diagram sistem yang digunakan, proses pengerjaan Proyek Akhir, dan proses instalasi serta konfigurasi sistem.

BAB IV HASIL DAN PENGUJIAN

Pada bab ini membahas tentang proses implementasi sistem sesuai dengan perencanaan yang telah dibuat, termasuk konfigurasi Prometheus, Grafana, dan bot Telegram dalam lingkungan Kubernetes.

BAB V PENUTUP

Pada bab ini membahas tentang kesimpulan dari pengerjaan Proyek Akhir dan saran untuk pembaca yang akan mengambil penelitian dengan topik yang sama.