

Implementasi Aplikasi Data Center Infrastructure Management (Dcim) Berdasarkan Arsitektur Microservices Menggunakan Kubernetes Studi Kasus Pt. Pelayaran Nasional Indonesia (Pelni)

1st Bintang Elizabeth Margaretha Sitorus

Fakultas Ilmu Terapan
Univeristas Telkom
Bandung, Indonesia

bintangems@students.telkomuniversity.ac.id

2nd Muhammad Iqbal

Fakultas Ilmu Terapan
Univeristas Telkom
Bandung, Indonesia

miqbal@telkomuniversity.ac.id

3rd Chadra Mahendra Putra

Pelayaran Nasional Indonesia
Bandung, Indonesia
chandra@pelni.co.id

Abstrak - Di era digital ini, perusahaan-perusahaan menghadapi tekanan untuk meningkatkan infrastruktur IT mereka guna mengelola data secara lebih efektif, aman, dan dapat diukur. Infrastruktur data tersebut menjadi kunci untuk menjaga kelancaran layanan dan operasional, terutama bagi perusahaan besar dengan data center yang kompleks dan luas. PT. Pelayaran Nasional Indonesia (PELNI), sebagai perusahaan terkemuka di industri pelayaran, juga menghadapi tantangan serupa dalam mengelola infrastruktur data center mereka. Penelitian ini 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. Untuk mengelola infrastruktur *data center* tersebut, PT. Pelayaran Nasional Indonesia (PELNI) memerlukan sistem pemantauan yang maju dan terintegrasi guna memantau pendataan serta kinerja infrastruktur secara *real-time*. Karena itu, integrasi aplikasi *Data Center Infrastructure Management* (DCIM) menggunakan Arsitektur Kubernetes menjadi keharusan bagi PT. Pelayaran Nasional Indonesia (PELNI). Diharapkan bahwa proyek akhir ini dapat merencanakan, mengimplementasikan, dan memantau infrastruktur menggunakan Kubernetes untuk aplikasi *Data Center Infrastructure Management* (DCIM) di PT. Pelayaran Nasional Indonesia (PELNI). Langkah-langkah ini tidak hanya meningkatkan efisiensi operasional, tetapi juga memungkinkan PT. Pelni dapat mengatasi kebutuhan dan tantangan manajemen pusat data yang lebih baik. Dengan menerapkan Kubernetes, PT. Pelayaran Nasional Indonesia (PELNI) dapat meningkatkan dokumentasi dan pengelolaan jaringan, serta memperkuat keamanan layanan infrastruktur data layanan DCIM.

Kata kunci: kubernetes, DCIM, arsitektur *microservices*, netbox

Dalam era digital saat ini, perusahaan-perusahaan menghadapi tuntutan untuk memperkuat infrastruktur Teknologi Informasi (TI) mereka agar dapat mengelola data dengan lebih efisien, aman, dan terukur. Infrastruktur data tersebut menjadi kunci untuk menjaga ketersediaan layanan dan operasional yang lancar. Hal ini menjadi krusial terutama bagi Perusahaan besar yang memiliki *data center* yang kompleks dan banyak. Dalam konteks ini, PT. Pelayaran Nasional Indonesia (PELNI) sebagai perusahaan besar yang bergerak di bidang pelayaran, memiliki kebutuhan yang serupa dalam mengelola infrastruktur *data center* mereka. Seiring dengan berkembangnya teknologi, perusahaan harus terus berinovasi dalam mengelola dan mengoptimalkan infrastruktur IT mereka. Salah satu pendekatan yang telah terbukti efektif adalah dengan memanfaatkan arsitektur *microservices* dan *platform orkestrasi kontainer* seperti Kubernetes. Arsitektur *microservices* memungkinkan aplikasi *Data Center Infrastructure Management* (DCIM) dibangun sebagai layanan terpisah, yang memungkinkan pengembangan, penerapan, dan perubahan dengan lebih cepat.

Tujuan dari penulisan proyek akhir ini bertujuan untuk mengimplementasikan aplikasi tersebut agar dapat lebih efektif mengelola perkembangan infrastruktur pusat data. Selain itu proyek ini bertujuan untuk mengotomatiskan seluruh proses pengelolaan infrastruktur pusat data melalui kubernetes, sehingga meningkatkan kinerja pusat data.

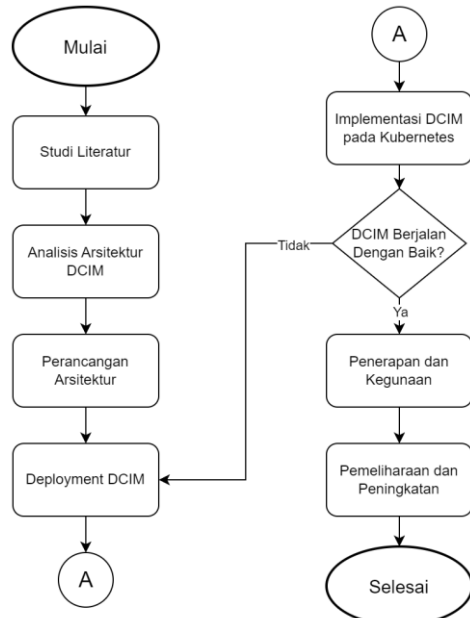
Rumusan masalah dalam proyek akhir ini mencakup beberapa aspek utama terkait penerapan kubernetes adalah bagaimana merancang dan mengintegrasikan infrastruktur pusat data dengan menggunakan kuberntes untuk mendukung aplikasi DCIM. Sehingga aplikasi yang dibangun dapat menampilkan fitur-fitur yang dibutuhkan dan dapat digunakan dalam aplikasi pusat data.

I. PENDAHULUAN

II. MATERIAL DAN METODOLOGI/PERANCANGAN

Dalam tahapan perancangan aplikasi DCIM dilakukan dengan menggunakan kubernetes dan netbox untuk menyimpan dan mengelola informasi dan proses dapat dilihat pada tahapan pembuatannya adalah sebagai berikut:

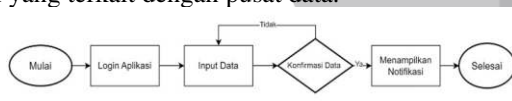
A. Tahapan Perancangan



Gambar 2. 1
Flowchart Perancangan Sistem

B. Flowchart Sistem Aplikasi

Tujuan dari pembuatan aplikasi netbox adalah untuk mendokumentasikan dan mengelola infrastruktur fisik dan virtual dalam pusat data. Aplikasi tersebut dapat memudahkan pengguna untuk membantu mengelola berbagai aspek dari pusat data termasuk dengan perangkat keras jaringan, server, rak, kabel dan elemen lain yang terkait dengan pusat data.



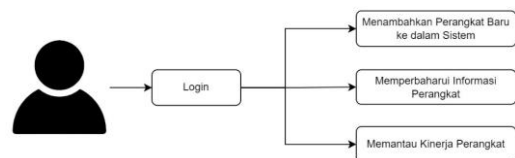
Gambar 2. 2
Flowchart Sistem Aplikasi

1. Tahapan awal mulai, pada tahapan ini dipersiapkan koneksi jaringan *internet* untuk menggunakan aplikasi dalam jarak dekat. Jika ingin menggunakan aplikasi dari jarak jauh, maka perlu menggunakan VPN.
2. Tahapan kedua adalah pengguna diminta *login* pada aplikasi. Hal ini memastikan bahwa pengguna memiliki izin yang benar untuk mengakses dan mengelola data dalam aplikasi.
3. Setelah berhasil *login*, pengguna memasukkan informasi data yang diperlukan sesuai kebutuhan aplikasi. Data ini berupa data yang ada di *data center*.

4. Setelah pengguna memasukkan informasi data, langkah selanjutnya adalah memverifikasi kebenaran informasi data yang dimasukkan.
5. Jika data yang dimasukkan benar, aplikasi menampilkan notifikasi kepada pengguna tentang keberhasilan proses pemasukan data. Jika data yang dimasukkan tidak benar, pengguna kembali ke langkah memasukkan kembali data.

C. Use Case

Perancangan *Use Case* diagram ini bertujuan untuk mendefinisikan interaksi antara pengguna dengan fungsionalitas dasar aplikasi. Dalam diagram tersebut, pengguna akan menjadi *administrator* pusat data. *Use Case* dapat menambahkan perangkat baru ke inventaris, memperbarui informasi perangkat, dan memantau aktivitas perangkat. Setiap *Use Case* terhubung dengan pengguna yang sudah ditetapkan, hal tersebut menentukan siapa yang dapat mengakses aplikasi. Diagram ini dapat memberikan gambaran yang jelas tentang bagaimana pengguna berinteraksi dengan aplikasi pengelolaan infrastruktur pusat data.



Gambar 2. 3
Use Case

D. Perancangan

Perancangan aplikasi ini dirancang aplikasi berdasarkan DCIM untuk pengelolaan pusat data dengan menggunakan database postgresql dan redis.

Adapun beberapa fitur yang ada dalam aplikasi ini, yaitu:

1. **Dashboard:** Menampilkan status dan kelengkapan infrastruktur pusat data, termasuk perangkat keras, server dan elemen lainnya.
2. **Organization:** Struktur organisasi untuk mengelola infrastruktur pusat data. Fitur ini memungkinkan pengguna membuat dan mengelola unit organisasi, seperti lokasi fisik yang terkait dengan infrastruktur. Hal ini juga memungkinkan pengguna untuk mengelompokkan perangkat keras dan koneksi jaringan berdasarkan kelompok organisasi pusat data.
3. **Devices:** Menyediakan manajemen perangkat lengkap dalam infrastruktur pusat data. Fitur ini memungkinkan pengguna untuk membuat dan mengelola informasi terkait perangkat jaringan seperti *switch*, *router*, *server*. Fitur ini juga memungkinkan pengguna memberikan informasi detail mengenai perangkat, seperti model, merek, nomor seri perangkat, serta informasi konfigurasi dan lokasi fisik. Sehingga, pengguna dapat memantau status dan ketersediaan perangkat, serta memantau kinerja perangkat.
4. **Connections:** Menampilkan pemantauan dan dokumentasi terperinci terkait koneksi fisik

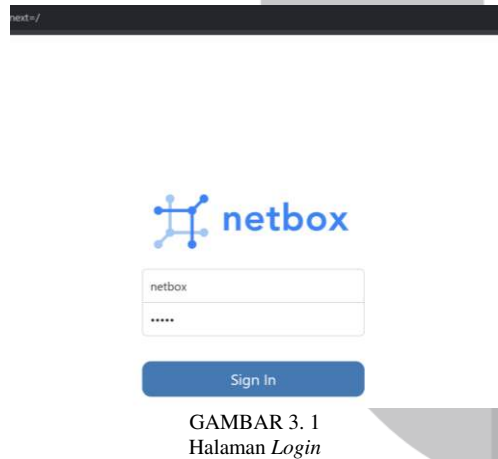
perangkat keras infrastruktur pusat data. Fitur ini juga memungkinkan pengguna untuk mengelola koneksi jaringan dan mengkonfigurasi koneksi fisik perangkat. Sehingga, pengguna dapat memeriksa status dan ketersediaan koneksi, serta melacak potensi titik kegagalan atau gangguan dalam jaringan.

III. HASIL PENELITIAN DAN PENGUJIAN

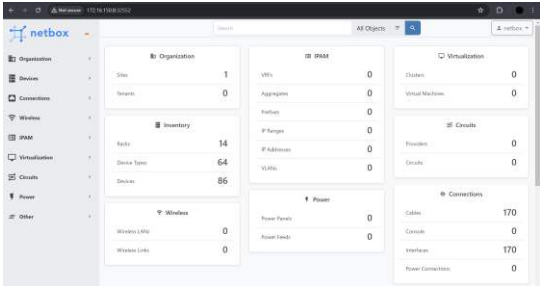
Hasil simulasi perancangan yang telah dilakukan pada metode sebelumnya. Simulasi tersebut mencakup beberapa aspek, antara lain penggunaan sistem dan fungsi sistem untuk memudahkan pengelolaan otomatis. Hasil ini bertujuan untuk **mengevaluasi** keberhasilan perancangan dan implementasi sistem serta mengidentifikasi informasi yang memerlukan perbaikan atau penyempurnaan. Dengan memahami hasilnya, maka dapat mengoptimalkan kinerja sistem secara keseluruhan.

A. Hasil Aplikasi DCIM

Pengguna membuka *browser* untuk mengakses aplikasi dengan memasukkan URL `http://SERVER-IP:32552`. Server merespon dengan menampilkan halaman *login* yang memungkinkan pengguna mengelola aplikasi. Setiap pengguna harus *login* untuk menggunakan aplikasi dengan memasukkan *username* dan *password* untuk memverifikasi identitas pengguna.



Pada halaman *dashboard*, menyediakan struktur organisasi untuk pengelolaan infrastruktur pusat data. Fitur ini memungkinkan pengguna membuat dan mengelola unit organisasi, seperti lokasi fisik yang terkait dengan infrastruktur.



GAMBAR 3.2
Halaman Dashboard

B. Pengujian Sistem

Pengujian ini dilakukan untuk memastikan bahwa aplikasi telah terinstal dengan benar di *cluster* kubernetes dan node jaringan berfungsi dengan benar. Perintah “`kubectl get deployments`” digunakan untuk memeriksa status dari semua hasil deployments, objek tersebut dapat mengelola dan memperbaharui aplikasi yang berjalan di dalam pod.

TABEL 3.1
Hasil Deployment

Deployment Status			
orangepi@worker01:~\$ kubectl get deployments -n netbox-system			
NAME	READY	UP-TO-DATE	AVAILABLE
AGE			
netbox-worker	1/1	1	1
12d			
netbox	1/1	1	1
12d			

Perintah “`kubectl get services`” digunakan untuk menampilkan daftar semua *services* yang ada di dalam *cluster*. *Services* merupakan objek yang mendefenisikan satu set pod dan kebijakan untuk mengaksesnya.

TABEL 3.2
Hasil Services

Services Status		
orangepi@worker01:~\$ kubectl get services -n netbox-system		
NAME	EXTERNAL-IP	TYPE
CLUSTER-IP	PORT (S)	
AGE		
netbox-postgresql-headless		ClusterIP
5432/TCP	12d	None <none>
netbox-redis-headless		ClusterIP
6379/TCP	12d	None <none>
netbox		NodePort
10.152.183.128		<none>
80:32552/TCP	12d	
netbox-redis-master		ClusterIP
10.152.183.52		<none>
6379/TCP	12d	
netbox-postgresql		ClusterIP
10.152.183.113		<none>
5432/TCP	12d	

```

netbox-redis-replicas
ClusterIP    10.152.183.27    <none>
6379/TCP    12d

```

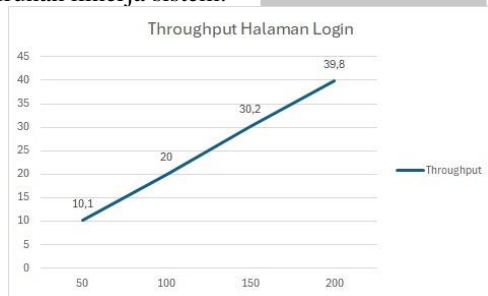
C. Pengujian Beban Sistem

Hasil pengujian dengan menggunakan JMeter telah menunjukkan bahwa ketiga endpoint, yaitu; halaman *login*, halaman *racks*, halaman *devices*, menunjukkan kapasitas penanganan permintaan yang tinggi dan stabilitas kinerja operasional yang baik. Terjadi peningkatan *throughput* secara linier dengan peningkatan jumlah threads pada setiap *endpoint*. Volume data yang diterima dan dikirim juga meningkat seiring dengan beban dan tidak terdapat *error* yang terjadi pada setiap level beban.

TABEL 3. 3
Pengujian Beban Sistem

No	Endpoint	Parameter	Test Plan Load Testing (Number of Thread)			
			50	100	150	200
1	Halaman Login	Throughput (KB/s)	10,1	20,0	30,2	39,8
		Error (%)	0,00	0,00	0,00	0,00
2	Halaman Racks	Throughput (KB/s)	10,1	17,4	30,7	33,1
		Error (%)	0,00	0,00	0,00	0,00
3	Halaman Devices	Throughput (KB/s)	10,1	14,5	19,0	32,3
		Error (%)	0,00	0,00	0,00	0,00

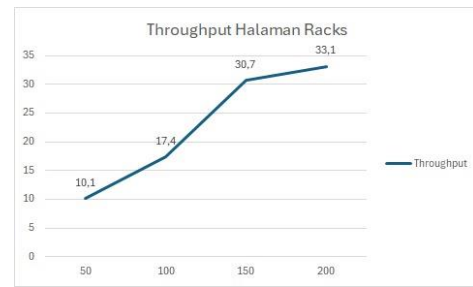
Grafik *throughput* halaman *login* menunjukkan peningkatan yang stabil dan linier seiring bertambahnya jumlah *thread*, dari 10,1 KB/s dengan 50 *thread* hingga 39,8 KB/s dengan 200 *thread*. Hal ini menunjukkan bahwa halaman *login* memiliki kinerja yang baik dengan mengalami peningkatan beban yang relatif, tanpa adanya penurunan kinerja sistem.



GAMBAR 3. 3

Grafik Hasil Pengujian Halaman Login

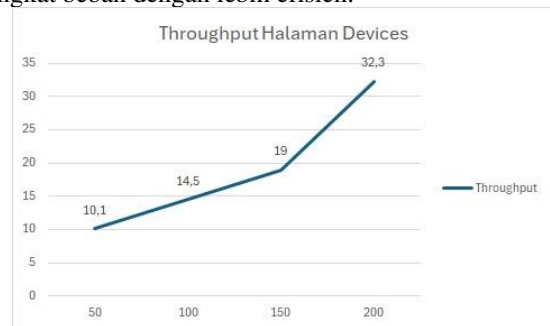
Grafik *throughput* halaman *racks* juga meningkat seiring dengan bertambahnya jumlah *thread*, namun tidak secara linier. *Throughput* meningkat dari 10,1 KB/s pada 50 *thread* menjadi 33,1 KB/s pada 200 *thread*, dengan menampilkan peningkatan yang signifikan pada 150 *thread*. Namun, peningkatan dari 150 *thread* ke 200 *thread* lebih meningkat, sehingga menunjukkan kemungkinan kendala kinerja pada halaman *racks* saat mencapai beban tinggi.



GAMBAR 3. 4

Grafik Hasil Pengujian Halaman Racks

Halaman *devices* menampilkan grafik *throughput* yang berbeda dengan peningkatan *throughput* yang lebih lambat hingga 150 *thread*. *Throughput* naik dari 10,1 KB/s pada 50 *thread* ke 19,0 KB/s pada 150 *thread*, namun pada 200 *thread* meningkat secara signifikan menjadi 32,3 KB/s. Hal ini menunjukkan bahwa halaman *devices* dapat mengalami hambatan atau keterbatasan kinerja pada beban rendah, namun dapat meningkat secara signifikan ketika beban tertentu tercapai. Secara menyeluruh, setiap halaman memiliki metrik kinerja karena peningkatan beban, sehingga memberikan informasi penting untuk mengoptimalkan dan meningkatkan sistem agar dapat menangani berbagai tingkat beban dengan lebih efisien.



GAMBAR 3. 5

Grafik Hasil Pengujian Halaman Devices

IV. KESIMPULAN

Berdasarkan hasil pengujian fungsionalitas terhadap fitur-fitur pada sistem aplikasi web yang telah dibuat, dapat disimpulkan bahwa seluruh fitur dapat berjalan 100% sebagaimana mestinya. Selain itu, penerapan sistem dengan Kubernetes menyediakan fitur untuk menerapkan dan mengelola aplikasi dalam *container* secara otomatis. Pengujian dan penerapan aplikasi DCIM menggunakan Kubernetes juga memfasilitasi pemantauan, pengelolaan, dan dokumentasi seluruh sumber daya pusat data, sehingga meningkatkan keandalan dalam pengelolaan infrastruktur pusat data. Hasil pengujian beban sistem menggunakan JMeter menunjukkan bahwa ketiga endpoint utama memiliki kapasitas penanganan permintaan yang tinggi dan stabilitas operasional yang baik, serta kinerja meningkat secara linier seiring dengan bertambahnya jumlah *thread* per 50 KB/s. Tidak ada error pada seluruh tingkat beban, dengan error rate 0,00% pada setiap *thread*.

REFERENSI

- [1] A. Tanuwijaya, H. Novianus Palit, and A. Noertjahyana, "Penerapan Microservices dan Amazon Elastic Container Service untuk Mendukung Scalability."
- [2] K. Yedutun, A. Noertjahyana, and H. Novianus Palit, "Implementasi Container Kubernetes untuk Mendukung Scalability."
- [3] A. Nurul Huda and S. Kusumawardani, "Huda, Kusumawardani-Kubernetes Cluster Management for Cloud Computing Platform: A Systematic Literature Review 75 KUBERNETES CLUSTER MANAGEMENT FOR CLOUD COMPUTING PLATFORM: A SYSTEMATIC LITERATURE REVIEW MANAJEMEN KLASER KUBERNETES PADA PLATFORM KOMPUTASI AWAN: TINJAUAN LITERATUR SISTEMATIK."
- [4] Editor, "Tentang PT Pelni,." pelni.co.id, 2019. [Online]. Available <https://www.pelni.co.id/tentang-kami>. [Accessed 16 02 2024]
- [5] T. Sinta Peringkat, berdasarkan S. Dirjen Penguatan RisBang Kemenristekdikti, A. Firmansyah, and N. Merlina, "PREDIKSI POLA PENJUALAN TIKET KAPAL PT. PELNI CABANG MAKASSAR MENGGUNAKAN METODE ALGORITMA APRIORI".
- [6] Editor, "Apa itu Kubernetes?"
- [7] L. Widyawati, H. Santoso, and H. Budiman, "ANALISA PENERAPAN SERVER DEPLOYMENT MENGGUNAKAN KUBERNETES UNTUK MENGHINDARI SINGLE OF FAILURE," 2021.
- [8] M. Harris, "Data Center InfrastruCture ManageMent," 2015. [Online]. Available: <http://www.wiley.com/go/datacenterhandbook>
- [9] "Aplikasi_Sistem_Monitoring_Server_Menggunakan_Device_Orange_Pi_berbasis_Web_Service".
- [10] M. Dicky Syahputra Lubis *et al.*, "MEMBANGUN ROUTER PADA JARINGAN KOMPUTER MENGGUNAKAN UBUNTU OS," *Jurnal Teknik Informatika Kaputama (JTIK)*, vol. 4, no. 2, 2020.
- [11] L. M. Alchuluq and F. Nurzaman, "ANALISIS PADA ARSITEKTUR MICROSERVICE UNTUK LAYANAN BISNIS TOKO ONLINE," 2021.
- [12] S. Saidah and R. Syaban, "IMPLEMENTASI ARSITEKTUR MICROSERVICES PADA APLIKASI POINT OF SALE TOKO FLYOVER STIKER," 2023. [Online]. Available: <https://flyoverstiker.online/>, Editor, "NetBox Documentation." Kubernetes.io, 2020. [Online]. Available: <https://kubernetes.io/id/docs/concepts/overview/what-is-kubernetes/> [Accessed 16 02 2024]
- [14] M. Fadlulloh and R. Bik, "IMPLEMENTASI DOCKER UNTUK PENGELOLAAN BANYAK APLIKASI WEB (Studi Kasus : Jurusan Teknik Informatika UNESA)," 2017.
- [15] K. Thias Widagdo, I. Bayu, and Y. A. Susetyo, "Pemodelan Sistem Monitoring Sensor Curah Hujan Menggunakan Grafana".
- [16] D. Rahman and H. Amnur, "Indri Rahmayuni 133 Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram Jurnal Ilmiah Teknologi Sistem Informasi," 2020. [Online]. Available: <http://jurnal-itsi.org>
- [17] R. Normadhoni, S. Putri Dewanti, W. Cahyo Namaskara, D. Yusfi Akhadi, and R. Fauzi, "Penggunaan Bot Telegram sebagai Announcemnt System dalam Dunia Parenting." [Online]. Available: <http://jurnalilmiah.org/journal/index.php/jet>