

Perancangan Dan Implementasi Sistem Manajemen Layanan Jaringan Internet Berbasis Docker Pada Aplikasi Netpro

1st Ghefira La-Agnasrasya Gracianette

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

graceghefira@student.telkomuniversity.ac.id

2nd Suci Aulia

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

sucia@tass.telkomuniversity.ac.id

Abstrak — Pengelolaan layanan aplikasi NetPro menghadapi tantangan penggunaan memori tinggi, konflik konfigurasi, dan kompleksitas deployment. Tugas akhir ini menggunakan teknologi Docker untuk meningkatkan efisiensi memori, menyederhanakan deployment, dan mengatasi konflik layanan. Metode mencakup konfigurasi Dockerfile, Docker Compose, dan integrasi source code, menggunakan alat seperti Docker, GitHub, dan Grafana. Hasil menunjukkan peningkatan efisiensi memori 4%, penyimpanan 8,5%, eliminasi konflik port, dan penyederhanaan deployment melalui Docker Hub. Dengan demikian, Docker meningkatkan efisiensi, fleksibilitas, dan stabilitas layanan NetPro.

Kata kunci — docker, container, deployment, NetPro

I. PENDAHULUAN

Dalam era digital, pengelolaan layanan aplikasi yang efisien menjadi kebutuhan utama. Aplikasi NetPro menghadapi tantangan seperti penggunaan memori tinggi, konflik konfigurasi, dan proses deployment yang kompleks. Teknologi Docker menawarkan solusi berupa isolasi container untuk meningkatkan efisiensi sumber daya, menjalankan layanan tanpa konflik, serta menyederhanakan deployment.

Docker memungkinkan pengelolaan memori terkontrol, layanan terisolasi, dan deployment otomatis dengan Docker Compose, yang mendukung konsistensi lingkungan aplikasi. Data menunjukkan bahwa penggunaan Docker mampu mengurangi overhead hingga 57% dan downtime hingga 70%, serta meningkatkan fleksibilitas layanan.

Penelitian ini bertujuan untuk:

1. Mengelola penggunaan memori secara efisien melalui pembatasan sumber daya container.
2. Menjalankan berbagai layanan secara bersamaan tanpa konflik konfigurasi.
3. Menyederhanakan proses deployment dengan Docker Compose.

II. KAJIAN TEORI

A. Docker

Platform containerization yang memungkinkan aplikasi dikemas dengan dependensi ke dalam container untuk eksekusi konsisten di berbagai lingkungan.

B. PuTTY

Perangkat lunak klien SSH/Telnet untuk mengelola server jarak jauh dengan fitur enkripsi, port forwarding, dan emulasi terminal.

C. Ubuntu

Sistem operasi berbasis Debian yang open-source, menyediakan lingkungan kerja gratis dan stabil untuk pengembangan aplikasi.

D. GitHub

Platform kolaborasi dan penyimpanan kode berbasis web yang mendukung pengelolaan proyek perangkat lunak secara efisien.

E. Prometheus

Alat pemantauan open-source untuk mengumpulkan data metrik berbasis waktu, dilengkapi bahasa kueri PromQL dan fitur peringatan.

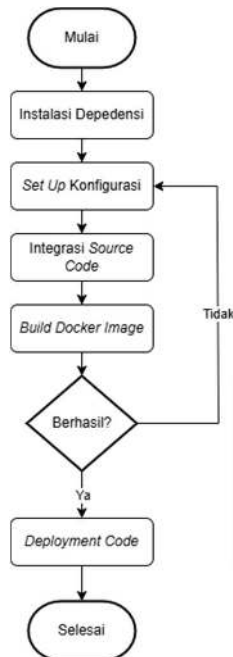
F. Grafana

Alat visualisasi open-source untuk data time series, memungkinkan pembuatan dashboard dinamis untuk analisis data.

G. Apache JMeter

Alat pengujian beban dan kinerja aplikasi, mensimulasikan akses pengguna untuk menganalisis respon server secara statistik.

III. METODE



GAMBAR 1
Flowchart Perancangan Aplikasi

Penelitian ini dilakukan untuk memastikan sistem berjalan optimal melalui tahapan berikut:

1. Instalasi perangkat lunak seperti Docker, Docker Compose, Nginx, Certbot, dan Git untuk menyiapkan lingkungan kerja.
2. Konfigurasi Nginx sebagai web server/reverse proxy, pembuatan Dockerfile dan Docker Compose untuk mengelola container, serta pengaturan jaringan dan variabel lingkungan agar layanan saling terhubung dengan optimal.
3. Sinkronisasi kode dari GitHub, pengujian integrasi, dan validasi untuk memastikan aplikasi berjalan tanpa error.
4. Pembuatan Image-Docker dari source code menggunakan Dockerfile untuk menghasilkan versi aplikasi yang konsisten dan portable.
5. Deploy Image-Docker ke server atau lingkungan produksi menggunakan container Docker, dilanjutkan dengan monitoring layanan untuk memastikan stabilitas dan kinerja sistem.

Setiap tahap mencakup evaluasi untuk memastikan keberhasilan atau melakukan perbaikan jika terjadi kendala.

IV. HASIL DAN PEMBAHASAN

A. Hasil Perancangan Sistem Aplikasi NetPro
Luaran pengembangan mencakup file konfigurasi seperti Dockerfile, docker-compose.yml, dan nginx.conf, yang mendukung operasi sistem secara optimal.

B. Pengujian Sistem Aplikasi NetPro

Pengujian melibatkan:

- Status Container: Semua container (laravel-app, nginx, db) berjalan dengan status "Up".
- Verifikasi Laravel-App: PHP 8.2, Composer 2.3.5, dan Laravel 11.44.2 terdeteksi sesuai konfigurasi.
- Verifikasi Nginx: Versi 1.27.4 berjalan optimal sebagai web server.
- Verifikasi Database: MySQL 5.7.44 berfungsi sebagai penyimpanan data.

Log menunjukkan semua permintaan diproses dengan kode HTTP 200, menandakan integrasi layanan berjalan lancar. Monitoring menggunakan Prometheus dan Grafana memastikan server stabil dengan sumber daya rendah.

C. Pengujian Beban

Uji beban dengan JMeter menunjukkan:

- Pada 10–50 pengguna: Sistem stabil dan efisien.



GAMBAR 2
Traffic Server 10-50 Pengguna

- Pada 100 pengguna: Sistem tetap stabil dengan waktu respons meningkat.



GAMBAR 3
Traffic Server 100 Pengguna

- Pada 500 pengguna: Lonjakan waktu respons mulai terlihat, namun semua permintaan tetap diproses.



GAMBAR 4
Traffic Server 500 Pengguna

D. Perbandingan Sistem Host Linux dan Docker

- Efisiensi Sumber Daya: Docker lebih hemat memori (25.8% vs 29.4%) dan penyimpanan (41.5% vs 49.8%) dibanding host Linux.



GAMBAR 5
Penggunaan Memori Sistem Host Linux



GAMBAR 6
Penggunaan Memori Sistem Docker

- Isolasi Layanan: Docker memungkinkan pengaturan port fleksibel untuk beberapa layanan di satu server.
- Kemudahan Deployment: Proses migrasi dan pengaturan aplikasi lebih cepat dengan Docker dibandingkan instalasi manual pada host Linux.

V. KESIMPULAN

Penerapan teknologi Docker dalam pengelolaan layanan aplikasi NetPro berhasil meningkatkan efisiensi, fleksibilitas, dan kemudahan pengelolaan. Pengujian menunjukkan bahwa Docker lebih hemat memori hingga 4% (25.8% vs 29.4%) dan penyimpanan hingga 8.5% (41.5% vs 49.8%) dibandingkan sistem host Linux. Keunggulan isolasi container memungkinkan layanan dengan konfigurasi berbeda, seperti Nginx dan Apache, berjalan bersamaan tanpa konflik port, yang sulit dicapai pada sistem host Linux. Selain itu, proses deployment aplikasi menjadi lebih sederhana dan cepat dengan penggunaan Image-Docker dari Docker Hub, tanpa perlu konfigurasi manual. Teknologi Docker terbukti memberikan stabilitas dan efisiensi dalam pengelolaan layanan aplikasi NetPro.

REFERENSI

- [1] A. Dhany Widiyanto, B. Anindito, and M. Noor Al Azam, "Implementation of Docker and Continuous Integration / Continuous Delivery for Management Information System Development," *International Journal of Electrical Engineering and Information Technology*, vol. 03, 2020.
- [2] A. M. Potdar, N. D. G. S. Kengond, and M. M. Mulla, "Performance Evaluation of Docker Container and Virtual Machine," *Procedia Comput Sci*, vol. 171, pp. 1419–1428, 2020, doi: 10.1016/j.procs.2020.04.152.
- [3] A. R. Ekaputra and A. S. Affandi, "Pemanfaatan layanan cloud computing dan docker container untuk meningkatkan kinerja aplikasi web," *Journal of Information System and Application Development*, vol. 1, no. 2, pp. 138–147, Sep. 2023, doi: 10.26905/jisad.v1i2.11084.
- [4] A. Sunyoto, "Uji Kinerja Sistem Web Service Pembayaran Mahasiswa Menggunakan Apache JMeter (Studi Kasus: Universitas AMIKOM Yogyakarta)," *Jurnal Teknologi Informasi*, vol. Vol. XIII No. 1, pp. 44–52, Apr. 2018.
- [5] Ahmed and G. Pierre, "Docker Container Deployment in Fog Computing Infrastructures," in *2018 IEEE International Conference on Edge Computing (EDGE)*, IEEE, Jul. 2018, pp. 1–8. doi: 10.1109/EDGE.2018.00008.
- [6] B. G. Pratama and M. F. Qodri, "Sistem Pemantauan Limbah Cair Berbasis Internet of Things dan Terproteksi Wireguard," *KURVATEK*, vol. 8, no. 1, pp. 99–108, Apr. 2023, doi: 10.33579/krvt.v8i1.4028.
- [7] B. Xu *et al.*, "Sledge: Towards Efficient Live Migration of Docker Containers," in *2020 IEEE 13th International Conference on Cloud Computing (CLOUD)*, IEEE, Oct. 2020, pp. 321–328. doi: 10.1109/CLOUD49709.2020.00052.
- [8] C. Chen, M. Hung, K. Lai, and Y. Lin, "Docker and Kubernetes," in *Industry 4.1*, Wiley, 2021, pp. 169–213. doi: 10.1002/9781119739920.ch5.
- [9] D. Rahman, H. Amnur, and I. Rahmayuni, "Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram," *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 1, no. 4, pp. 133–138, Dec. 2020, doi: 10.62527/jitsi.1.4.19.
- [10] G. Y. Saputra, A. D. Afrizal, F. K. R. Mahfud, F. A. Pribadi, and F. J. Pamungkas, "Penerapan Protokol Mqtt pada Teknologi Wan (Studi Kasus Sistem Parkir Universitas Brawijaya)," Sep. 14, 2017. doi: 10.31219/osf.io/anbhf.
- [11] H. Mubarak and H. Saptono, "Analisis dan Perancangan Platform Virtualisasi Berbasis Docker," *Jurnal Informatika Terpadu*, vol. 5, no. 1, pp. 18–23, Mar. 2019, doi: 10.54914/jit.v5i1.175.
- [12] I. Vasireddy, G. Ramya, and P. Kandi, "Kubernetes and Docker Load Balancing: State-of-the-Art Techniques and Challenges," *International Journal of Innovative Research in Engineering and Management*, vol. 10, no. 6, pp. 49–54, Dec. 2023, doi: 10.55524/ijirem.2023.10.6.7.
- [13] J. Cito, G. Schermann, J. E. Wittern, P. Leitner, S. Zumberi, and H. C. Gall, "An Empirical Analysis of the Docker Container Ecosystem on GitHub," in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, IEEE, May 2017, pp. 323–333. doi: 10.1109/MSR.2017.67.

- [14] J. Mahboob and J. Coffman, "A Kubernetes CI/CD Pipeline with Asylo as a Trusted Execution Environment Abstraction Framework," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, Jan. 2021, pp. 0529–0535. doi: 10.1109/CCWC51732.2021.9376148.
- [15] M. Fadlulloh and R. Bik, "Implementasi Docker untuk Pengelolaan Banyak Aplikasi Web (Studi Kasus : Jurusan Teknik Informatika Unesa)," 2017.
- [16] M. H. Saputra, Ferdiayansah, and F. Dristyan, "Pemanfaatan Virtualisasi Di Ubuntu Untuk Optimalisasi Sumber Daya: Manfaat, Tantangan, Dan Solusi," *Journal Of Computer Science And Technology (JOCSTEC)*, vol. 3, no. 1, pp. 49–54, Jan. 2025, doi: 10.59435/jocstec.v3i1.431.
- [17] M. Rodriguez and R. Buyya, "Container Orchestration With Cost-Efficient Autoscaling in Cloud Computing Environments," 2020, pp. 190–213. doi: 10.4018/978-1-7998-2701-6.ch010.
- [18] M. Sureshkumar and P. Rajesh, "Optimizing The Docker Container Usage Based on Load Scheduling," in *2017 2nd International Conference on Computing and Communications Technologies (ICCCCT)*, IEEE, Feb. 2017, pp. 165–168. doi: 10.1109/ICCCCT2.2017.7972269.
- [19] N. Putu, V. D. Saraswati, N. Yudistira, and P. P. Adikara, "Analisis Sentimen terhadap Perundungan Siber pada Twitter menggunakan Algoritma Bidirectional Encoder Representations from Transformer (BERT)," 2023. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [20] R. A. Megantara, F. Alzami, R. A. Pramunendar, and D. P. Prabowo, "Pengembangan dan Implementasi Docker untuk Memaksimalkan Utilitas Server Universitas pada Masa Covid-19," *Transmisi*, vol. 24, no. 2, pp. 48–54, May 2022, doi: 10.14710/transmisi.24.2.48-54.
- [21] R. Buyya, M. A. Rodriguez, A. N. Toosi, and J. Park, "Cost-Efficient Orchestration of Containers in Clouds: A Vision, Architectural Elements, and Future Directions," *J Phys Conf Ser*, vol. 1108, p. 012001, Nov. 2018, doi: 10.1088/1742-6596/1108/1/012001.
- [22] S. Amgothu, "An End-to-End CI/CD Pipeline Solution Using Jenkins and Kubernetes," *International Journal of Science and Research (IJSR)*, vol. 13, no. 8, pp. 1576–1578, Aug. 2024, doi: 10.21275/SR24826231120.
- [23] S. Dwiyanho, E. Rakhmat, and O. Gustiawan, "Implementasi Virtualisasi Server Berbasis Docker Container," vol. 7, no. 2, 2020.
- [24] V. Chaitanya and S. S. Kirubakaran, "Online Notice Board Using Raspberries Pi Using Putty Software and Internet of Things," 2022, p. 030055. doi: 10.1063/5.0111067.
- [25] W. M. C. J. T. Kithulwatta, K. P. N. Jayasena, B. T. G. S. Kumara, and R. M. K. T. Rathnayaka, "Integration With Docker Container Technologies for Distributed and Microservices Applications," *International Journal of Systems and Service-Oriented Engineering*, vol. 12, no. 1, pp. 1–22, Apr. 2022, doi: 10.4018/IJSSOE.297136.
- [26] Y. Aldwyhan, R. O. Sinnott, and G. T. Jayaputera, "Elastic Deployment of Container Clusters Across Geographically Distributed Cloud Data Centers for Web Applications," *Concurr Comput*, vol. 33, no. 21, Nov. 2021, doi: 10.1002/cpe.6436.
- [27] Z. Arif, A. Alim Murtopo, and S. Aris Munandar, "Penerapan Squirrelmail pada Mail Server Berbasis Linux Ubuntu."