

OPTIMASI BUILD DAN DEPLOYMENT APLIKASI BUN-FASTIFY DAN NEXT.JS MENGGUNAKAN DOCKER MULTI-STAGE BUILD PADA SISTEM AKADEMIK UMPO

Titan Apriliyan N. A, Adi Fajaryanto C., Ismail Abdurrozzaq Z.

Fakultas Teknik, Program Studi Teknik Informatika, Universitas Muhammadiyah
Ponorogo

e-mail: titanaprilian73@gmail.com

ABSTRAK

Sistem Informasi Akademik (SIMAK) Universitas Muhammadiyah Ponorogo yang terdiri dari layanan *frontend admin*, *frontend student*, dan *backend* menghadapi permasalahan ukuran *image* Docker yang besar, sehingga berdampak pada lambatnya proses *deployment* serta pemborosan sumber daya penyimpanan dan bandwidth saat distribusi *image*. Penelitian ini bertujuan untuk mengoptimalkan ukuran *image* Docker pada ketiga layanan tersebut menggunakan pendekatan *multi-stage build* guna meningkatkan efisiensi *build* dan *deployment* aplikasi berbasis *Bun-Fastify* dan *Next.js*. Penelitian dilakukan secara kuantitatif dengan membangun dua versi *image* untuk masing-masing layanan, yaitu versi *unoptimized* (satu *stage* tunggal) dan versi *optimized* (*multi-stage build* yang memisahkan tahap *builder* dan *runner*), kemudian membandingkan ukuran *image*, jumlah *layer*, dan ukuran *build context* menggunakan Docker Engine dan Portainer sebagai alat manajemen. Hasil penelitian menunjukkan bahwa pendekatan *multi-stage build* berhasil menurunkan ukuran *image backend* dari 709 MB menjadi 347 MB (51%), *frontend admin* dari 1,92 GB menjadi 373 MB (83,5%), dan *frontend student* dari 1,46 GB menjadi 386 MB (71,3%). Reduksi signifikan pada layanan *frontend* diperoleh dari pemanfaatan fitur *standalone output* Next.js yang melakukan *tree-shaking* dependensi, sedangkan pada *backend* diperoleh dari pemisahan artefak hasil kompilasi terhadap *source code* dan *devDependencies*. Penelitian ini menyimpulkan bahwa penerapan *multi-stage build* efektif digunakan sebagai strategi optimasi *image* Docker pada sistem akademik berbasis Bun dan Next.js.

Kata Kunci: *Containerization, Docker, Multi-stage Build, Optimasi Image.*

DAFTAR PUSTAKA

- [1] O. Bentaleb, A. S. Z. Belloum, A. Sebaa, and A. El-Maouhab, "Containerization technologies: taxonomies, applications and challenges," *Journal of Supercomputing*, vol. 78, no. 1, pp. 1144–1181, Jan. 2022, doi: 10.1007/s11227-021-03914-1.
- [2] Stack Overflow, "2025 Developer Survey," 2025. Accessed: Jul. 16, 2026. [Online]. Available: <https://survey.stackoverflow.co/2025/developers>
- [3] N. Zhao *et al.*, "Large-Scale Analysis of Docker Images and Performance Implications for Container Storage Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 4, pp. 918–930, Apr. 2021, doi: 10.1109/TPDS.2020.3034517.
- [4] K. Eng and A. Hindle, "Revisiting Dockerfiles in Open Source Software Over Time," Mar. 2021, doi: <https://doi.org/10.48550/arXiv.2103.12298>.
- [5] T. Durieux, "Empirical Study of the Docker Smells Impact on the Image Size," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2024, pp. 2568–2579. doi: 10.1145/3597503.3639143.
- [6] L. Gu, D. Zeng, J. Hu, H. Jin, S. Guo, and A. Y. Zomaya, "Exploring layered container structure for cost efficient microservice deployment," in *Proceedings - IEEE INFOCOM*, Institute of Electrical and Electronics Engineers Inc., May 2021. doi: 10.1109/INFOCOM42981.2021.9488918.
- [7] B. Kaur, M. Dugré, A. Hanna, and T. Glatard, "An analysis of security vulnerabilities in container images for scientific data analysis," *Gigascience*, vol. 10, no. 6, Jun. 2021, doi: 10.1093/gigascience/giab025.
- [8] J. Park, D. Kim, and K. Yeom, "An Approach for Reconstructing Applications to Develop Container-Based Microservices," *Mobile Information Systems*, vol. 2020, 2020, doi: 10.1155/2020/4295937.
- [9] M. R. Fachrudin and A. S. Affandi, "Implementation and Analysis of Container Image Optimization Using Alpine Linux and Multi-Stage Builds,"

- Teknika*, vol. 14, no. 1, pp. 9–18, Mar. 2025, doi: 10.34148/teknika.v14i1.1118.
- [10] 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 *IEEE International Working Conference on Mining Software Repositories*, IEEE Computer Society, Jun. 2017, pp. 323–333. doi: 10.1109/MSR.2017.67.
- [11] Docker Inc., “Multi-stage builds,” 2025. Accessed: Jul. 16, 2026. [Online]. Available: <https://docs.docker.com/get-started/docker-concepts/building-images/multi-stage-builds/>
- [12] Bun Team, “Welcome to Bun,” Bun Documentation. Accessed: Jun. 10, 2026. [Online]. Available: <https://bun.com/docs>
- [13] D. Reis, B. Piedade, F. F. Correia, J. P. Dias, and A. Aguiar, “Developing Docker and Docker-Compose Specifications: A Developers’ Survey,” *IEEE Access*, vol. 10, 2022, doi: 10.1109/ACCESS.2021.3137671.