

INTEGRASI *KUBERNETES* DENGAN *LOAD BALANCING* DAN SKALABILITAS DALAM PENGEMBANGAN APLIKASI MANAJEMEN LINGKUNGAN BERBASIS ANDROID UNTUK PERUMAHAN KOMERSIL

Feri Winarta^{1*}, Rino²

^{1,2}Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Buddhi Dharma
Email: feriwnarta26@gmail.com*

Abstrak

Penelitian ini berfokus pada pengembangan aplikasi manajemen lingkungan berbasis *Android* untuk perumahan komersil dengan mengintegrasikan *Kubernetes*, *load balancing*, dan skalabilitas. Latar belakang penelitian ini didasari oleh kebutuhan akan sistem yang efisien dalam menangani laporan permasalahan lingkungan yang sering kali tidak terkelola dengan baik. Tujuan dari penelitian ini adalah untuk menciptakan aplikasi yang dapat meningkatkan responsivitas dan ketersediaan layanan, serta memudahkan warga dalam melaporkan isu lingkungan secara *real-time*. Metode penelitian yang digunakan meliputi perancangan sistem, pengembangan aplikasi menggunakan *framework Flutter*, dan penerapan *Kubernetes* untuk pengelolaan kontainer. Hasil penelitian menunjukkan bahwa aplikasi yang dikembangkan mampu menangani lonjakan permintaan dengan baik, berkat penerapan *load balancing* dan *horizontal scaling* yang dilakukan oleh *Kubernetes*. Temuan ini mengindikasikan bahwa integrasi teknologi ini tidak hanya meningkatkan performa aplikasi, tetapi juga memberikan pengalaman pengguna yang lebih baik. Kesimpulan dari penelitian ini adalah bahwa penggunaan *Kubernetes* dalam pengembangan aplikasi manajemen lingkungan sangat efektif dan dapat menjadi model untuk aplikasi serupa di masa depan. Penelitian lebih lanjut diperlukan untuk mengeksplorasi fitur tambahan dan peningkatan sistem agar dapat memenuhi kebutuhan pengguna yang terus berkembang.

Kata kunci: *Kubernetes*, *Load Balancing*, Skalabilitas, *Flutter*, Manajemen Perumahan.

Pendahuluan

Perumahan adalah kumpulan rumah sebagai bagian dari permukiman, baik perkotaan maupun perdesaan, yang dilengkapi dengan prasarana, sarana, dan utilitas umum sebagai hasil upaya pemenuhan rumah yang layak huni (UURI Nomor 1/2011 tentang Perumahan dan Kawasan Permukiman) (Dr. Ir. M. Amir Salipu et al., 2023). RW 05 Bukit Golf Mediterania PIK memiliki divisi tambahan seperti *estate manager* yang bertugas mengelola lingkungan perumahan. Tugas-tugas tersebut meliputi perawatan taman, kebersihan lingkungan, kesehatan lingkungan, perawatan mekanikal dan elektrikal, perawatan *clubhouse*,

pengendalian bangunan, perawatan kolam renang, pengurusan administrasi, dan permasalahan keamanan.

Sebelumnya, RW 05 telah menerapkan pelaporan permasalahan lingkungan melalui aplikasi perpesanan daring. Namun, cara ini masih tergolong manual dan memiliki beberapa kelemahan, seperti kesulitan bagi pekerja untuk mendapatkan lokasi terkini permasalahan (Handoyo et al., 2022). Untuk mengatasi permasalahan tersebut, salah satu solusi yang dapat diterapkan adalah pengembangan aplikasi manajemen lingkungan berbasis *Android*. *Flutter* sendiri adalah sebuah *framework* pemrograman dengan bahasa *Dart* yang memudahkan pengembang untuk membuat aplikasi yang dapat berjalan di lebih dari satu jenis (Nelly Sofi & Riza Dharmawan, 2022),

Pada saat ini, *Flutter* dapat digunakan untuk aplikasi *mobile* (*iOS* dan *Android*), *web*, dan *desktop*. Namun, seiring dengan berkembangnya aplikasi dan meningkatnya jumlah pengguna, diperlukan infrastruktur yang mampu menangani beban kerja yang tinggi dan memastikan ketersediaan layanan secara terus-menerus. Masalah yang dihadapi adalah bahwa beban lalu lintas yang tinggi sering menyebabkan server mengalami *downtime* dan kelebihan beban (Ridha & Suhatman, 2022). *Load balancing* adalah metodologi strategis yang digunakan untuk mengalokasikan lalu lintas data dengan cara yang mendukung distribusi beban kerja yang merata di berbagai jaringan (Odhi Dwi Putra et al., 2020).

Teknik ini bertujuan untuk meningkatkan pemanfaatan sumber daya dan meningkatkan kinerja sistem secara keseluruhan (Khoiru Sabila et al., 2024). Dengan mengoptimalkan distribusi beban, *load balancing* bukan hanya meningkatkan kinerja jaringan saja, tetapi juga memastikan layanan yang ada memiliki ketersediaan yang tinggi serta mengurangi risiko kegagalan sistem (Tanuwijaya et al., 2021). Selain itu, *horizontal* skalabilitas pun diperlukan.

Horizontal skalabilitas adalah kemampuan suatu layanan untuk memperluas kapasitas sumber daya yang dimilikinya dalam menghadapi peningkatan beban kerja. Dengan kata lain, layanan yang memiliki skalabilitas yang baik dapat menyesuaikan diri secara efektif terhadap permintaan yang meningkat, baik melalui penambahan sumber daya secara vertikal, misalnya, meningkatkan spesifikasi *server*, maupun *horizontal*, misalnya, menambah jumlah *server* atau meningkatkan

jumlah *instance* (Subhi Basit et al., 2023). Skala *horizontal* merujuk pada kemampuan sebuah orkestrator atau *platform* manajemen kontainer untuk meningkatkan jumlah *instance* atau layanan dengan menambahkan lebih banyak *node* ke dalam klaster (Rina & Ridha, 2021).

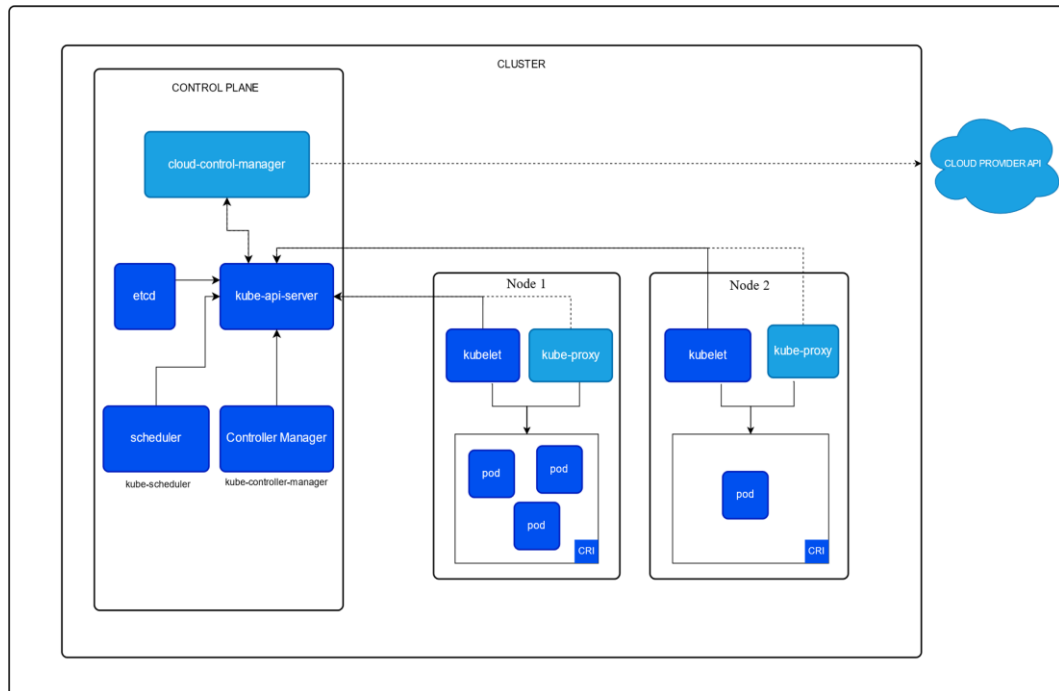
Kubernetes adalah kerangka kerja sumber terbuka yang dirancang untuk mengelola beban kerja aplikasi yang terkontainerisasi, yang menawarkan fitur-fitur seperti konfigurasi deklaratif dan otomatisasi (Magdalena et al., 2024). Kerangka kerja ini hadir dalam ekosistem yang luas dan berkembang pesat. Berbagai layanan, dukungan, dan alat yang terkait dengan *Kubernetes* dapat diakses dengan mudah (Cahyani et al., 2020).

Dengan menerapkan *Kubernetes*, aplikasi dapat di-*deploy* dalam bentuk kontainer yang terisolasi, sehingga memudahkan pengelolaan dan pengembangan lebih lanjut. *Kubernetes* berfungsi sebagai *platform* orkestrasi *kontainer* yang memungkinkan pengelolaan aplikasi yang terdistribusi dengan efisien, serta menawarkan fitur-fitur seperti konfigurasi deklaratif dan otomatisasi (Rina Noviana, 2022).

Metodologi

Kubernetes adalah platform sumber terbuka yang digunakan untuk mengelola beban kerja aplikasi yang berada dalam kontainer, serta menawarkan pengaturan dan otomatisasi dengan pendekatan deklaratif (Kubernetes, 2024).

Kubernetes berfungsi sebagai platform orkestrasi kontainer dan menawarkan kerangka kerja untuk memfasilitasi berbagai fungsinya. Arsitekturnya terdiri dari komponen-komponen penting, termasuk *Pod*, *Label*, dan *Selector* (Ridha & Suhatman, 2022)



Gambar 1. Arsitektur Kubernetes

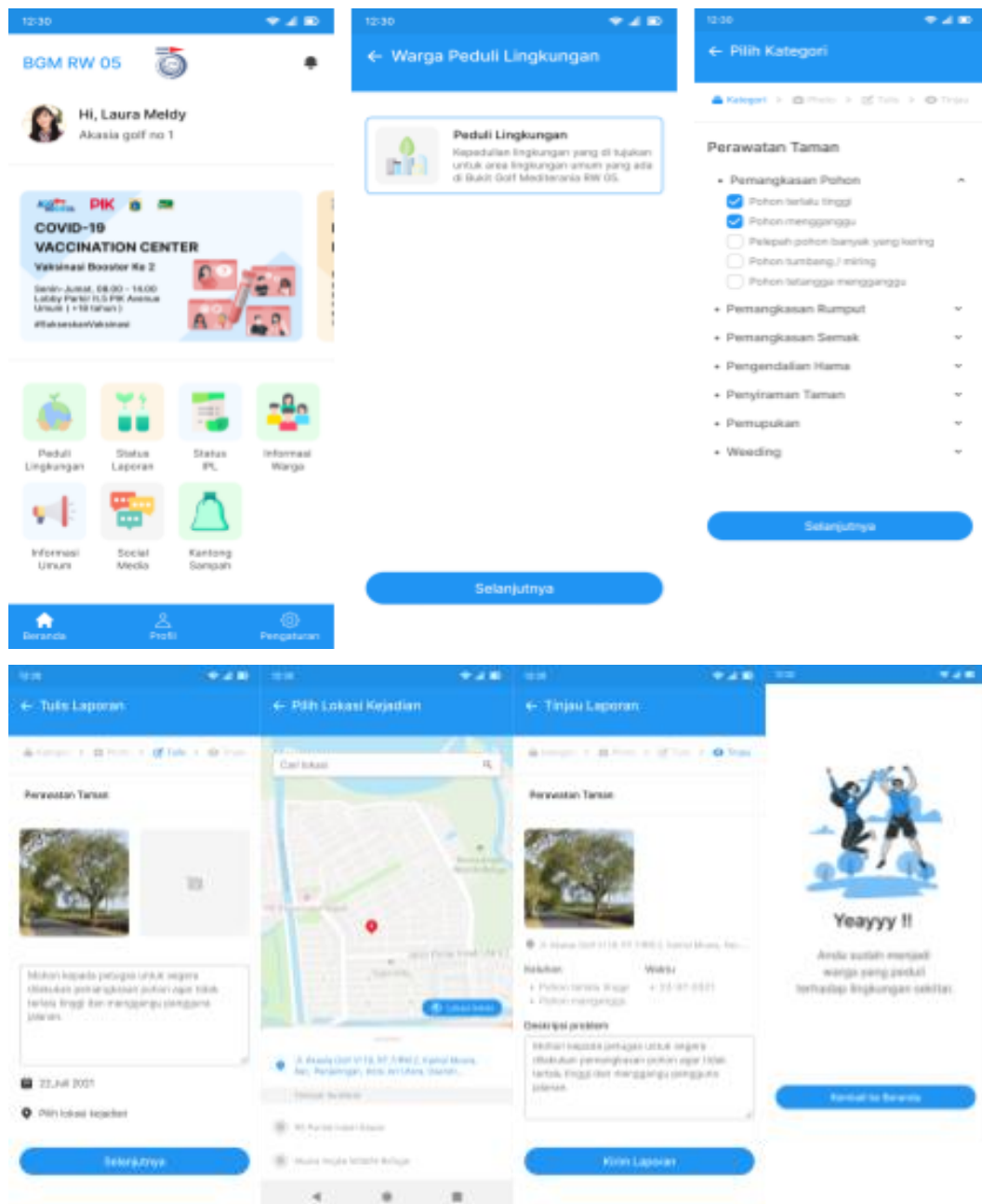
Kluster Kubernetes terdiri dari satu bidang kontrol dan sejumlah mesin pekerja, yang dikenal sebagai *node*, yang menjalankan aplikasi dalam kontainer. Setiap kluster membutuhkan minimal satu *node* pekerja untuk mengoperasikan *Pod*. *Node* pekerja berfungsi sebagai tuan rumah bagi *Pod*, yang merupakan elemen dari beban kerja aplikasi. Bidang kontrol bertugas mengelola *node* pekerja dan *Pod* yang terdapat dalam kluster. Dalam konfigurasi produksi, bidang kontrol umumnya dijalankan di beberapa komputer, dan kluster biasanya dapat menjalankan beberapa *node* untuk memastikan toleransi terhadap kesalahan dan tinggi ketersediaan (Sari et al., 2023)

Arsitektur *Kubernetes* terdiri dari dua komponen utama, *control plane* dan *node worker*. *Control plane* bertanggung jawab untuk pengambilan keputusan strategis dan penanganan peristiwa dalam kluster. Komponen utama dalam *control plane* meliputi *Server API*, yang menyediakan akses ke *API Kubernetes* dan bertindak sebagai antarmuka depan. *Etc*, yang berfungsi sebagai penyimpanan nilai kunci yang konsisten dan andal untuk seluruh data kluster, *Kube Scheduler*, yang memantau *Pod* baru tanpa penugasan *node* dan memilih *node* untuk mengeksekusinya. serta *Kube Controller Manager*, yang mengelola berbagai

kontroler untuk memastikan keadaan klaster sesuai dengan yang diinginkan pengguna, seperti menambah atau menghapus *Pod* (M Ramadhan Muttakin, 2021).

Hasil dan Pembahasan

Pembuatan aplikasi manajemen lingkungan berbasis *Android* untuk perumahan komersial tentunya perlu mengedepankan tampilan *UI* yang interaktif dan *user-friendly*, sehingga pengguna dapat dengan mudah mengakses berbagai fitur dan informasi yang diperlukan.



Gambar 2. Tampilan Komplain Warga didalam Aplikasi

Dengan antarmuka yang intuitif, pengguna dapat mengisi formulir Komplain dengan *detail* yang diperlukan, termasuk deskripsi masalah dan lokasi, serta melampirkan foto jika diperlukan, Tampilan ini tidak hanya memfasilitasi pengumpulan informasi yang diperlukan untuk penanganan masalah, tetapi juga memberdayakan warga untuk berpartisipasi aktif dalam menjaga dan meningkatkan kualitas lingkungan mereka.

Performa *load balancing web server* dapat diukur menggunakan beberapa parameter, di antaranya adalah *CPU Utilization*, *Memory Usage*, *Response Time*, *throughput*, dan *error rate*. Skenario pengujian ini untuk mendapatkan hasil dari kinerja *kubernetes* ingress controller yang mengimplementasikan *load balancing*. Tools ataupun aplikasi yang digunakan untuk pengujian ini ialah *Apache Jmeter* agar mendapatkan nilai parameter *Throughput*, *Response Time*, dan *Error rate*

Untuk menjalankan pengujian dapat dilakukan dengan beberapa uji coba melalui skenario pengujian dengan jumlah koneksi yang berbeda pada tingkat pengujian 100 – 1600 *Thread*, maka ditentukan jumlah *thread* yang digunakan yaitu 100, 400, 800, 1200 1600 *Request rate* yang rata rata setiap detik adalah 10 *request* per detik disetiap tingkatan *thread* dengan pengulangan 10 Pengujian akan dilakukan dengan skenario 3 *pod*.

Tabel 1. Hasil pengujian load balancing

<i>Thread</i>	<i>Response time (ms)</i>	<i>Error Rate</i>	<i>Throughput</i>
100	162,91	0,1	25,4
400	216,35	0,18	33,8
800	292,06	9,85	48,4
1200	508,16	8,09	83.3
1500	863,26	12,53	14,5

Pengujian skalabilitas ini bertujuan untuk menjalankan *HPA (Horizontal pod autoscaler)* dalam mengelola sumber daya aplikasi secara dinamis, memastikan bahwa aplikasi dapat beradaptasi dengan perubahan permintaan tanpa mengorbankan kinerja. Pengujian skalabilitas ini bertujuan untuk menjalankan *HPA (Horizontal pod autoscaler)* dalam mengelola sumber daya aplikasi secara dinamis, memastikan bahwa aplikasi dapat beradaptasi dengan perubahan permintaan tanpa mengorbankan kinerja.

Tabel 2. Hasil pengujian skalabilitas

<i>Thread</i>	<i>Target CPU (%)</i>	<i>Min Pod</i>	<i>Max Pod</i>	<i>Replicas</i>
0	0% 50%	2	5	2
1500	108% 50%	2	5	5

Simpulan

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa aplikasi manajemen lingkungan berbasis *Android* yang terintegrasi dengan *Kubernetes* berhasil memenuhi tujuan yang ditetapkan, yaitu meningkatkan efisiensi dan responsivitas dalam pengelolaan masalah lingkungan. Penerapan *Docker* untuk pembuatan *image* dan penggunaan *Kubernetes* untuk pengelolaan kontainer memberikan fleksibilitas dan skalabilitas yang diperlukan untuk menangani beban kerja yang dinamis.

Integrasi *Prometheus* dan *Grafana* juga memberikan keuntungan tambahan dalam hal pemantauan dan analisis kinerja sistem, memungkinkan pengembang untuk mengidentifikasi dan mengatasi masalah secara proaktif. Dengan demikian, penelitian ini menunjukkan bahwa teknologi kontainer dan alat pemantauan dapat berkontribusi secara signifikan terhadap pengelolaan aplikasi yang lebih baik, terutama dalam konteks spesifikasi *server* yang terbatas. Penelitian lebih lanjut disarankan untuk mengeksplorasi fitur tambahan dan peningkatan sistem agar dapat memenuhi kebutuhan pengguna yang terus berkembang.

Daftar Pustaka

- Cahyani, A., Rahayu, W. I., & Fatonah, R. N. S. (2020). *Panduan Pembuatan dan Penggunaan Aplikasi Implementasi Metode Servaqual Untuk Mengetahui Kepuasan Pelanggan Berdasarkan Hasil Penanganan Menggunakan Metode Wighted Product Pada PT. CDA*. Kreatif. <https://books.google.co.id/books?id=HND9DwAAQBAJ>
- Dr. Ir. M. Amir Salipu, M. T., Anggia R. Nurmaningtyas, S. T. M. S., Musfira, S. T. M. S., Dewi Angraeni, S. T. M. T., Alfred Benjamin Alfons, S. T. M. T., & Sugito Utomo, S. T. M. S. (2023). *Pengantar Perumahan Dan Permukiman Tinjauan Tentang Standar Dan Aturan Dalam Perencanaan Pembangunan Permukiman Modern Dan Tradisional*. Deepublish. <https://books.google.co.id/books?id=m6nXEAAAQBAJ>
- Handoyo, E. D., Santoso, S., & Surjawan, D. J. (2022). Pengembangan Aplikasi Mobile Pemesanan dan Pembayaran Makanan Berbasis Cloud Storage. *Jurnal Teknik Informatika Dan Sistem Informasi*, 8(1), 161–174. <https://doi.org/10.28932/jutisi.v8i1.4393>
- Khoiru Sabila, Siti Rahayu, & Titin Sumarni. (2024). Peningkatan Efisiensi

- Penggunaan Sumber Daya Jaringan Melalui Teknik Load Balancing. *CEMERLANG: Jurnal Manajemen Dan Ekonomi Bisnis*, 4(3), 31–41. <https://doi.org/10.55606/cemerlang.v4i3.2989>
- Kubernetes. (2024). *Kubernetes Web Dokumentasi*. <https://kubernetes.io/id/docs/concepts/overview/>
- M Ramadhan Muttakin, M. A. F. R. (2021). Implementasi Kubernetes Cluster Menggunakan Vagrant. *Applied Business and Engineering Conference*, 9, 218–227. <https://abecindonesia.org/proceeding/index.php/abec/article/view/45/150>
- Magdalena, L., Subagyo, M. H. A., Iskandar, Y. W., Bahtiar, U. H., Eliyaturohman, P. S., Efitra, E., & Uzma, I. (2024). *Design Pattern: Membangun Aplikasi Software yang Unggul dari Capstone Proyek*. PT. Sonpedia Publishing Indonesia. https://books.google.co.id/books?id=j_T8EAAAQBAJ
- Nelly Sofi, & Riza Dharmawan. (2022). Perancangan Aplikasi Bengkel Csm Berbasis Android Menggunakan Framework Flutter (Bahasa Dart). *Jurnal Teknik Dan Science*, 1(2), 53–64. <https://doi.org/10.56127/jts.v1i2.125>
- Odhi Dwi Putra, T., Widiarto, W., & Wiharto. (2020). Implementasi Algoritma Load Balancing PLBA Komputasi Grid pada Lab Environment Menggunakan PVM3. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, 4(6), 1190–1197. <https://doi.org/10.29207/resti.v4i6.2606>
- Ridha, M. A. F., & Suhatman, R. (2022). Perbandingan Kinerja Kubernetes Cluster dengan Virtualisasi KVM, Vagrant dan LXD. *Jurnal Komputer Terapan*, 8(1), 151–157. <https://doi.org/10.35143/jkt.v8i1.5231>
- Rina, N. K. S., & Ridha, M. A. F. (2021). The Implementasi Kubernetes Cluster Menggunakan KVM. *ABEC Indonesia*, 209–217. <https://abecindonesia.org/proceeding/index.php/abec/article/view/151%0Ahttps://abecindonesia.org/proceeding/index.php/abec/article/download/151/149>
- Rina Noviana. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql. *Jurnal Teknik Dan Science*, 1(2), 112–124. <https://doi.org/10.56127/jts.v1i2.128>
- Sari, W. T. W., Purwantoro, S., & Ridha, M. A. F. (2023). Jurnal Politeknik Caltex Riau Pengukuran Kinerja Kubernetes Cluster pada Nested Virtualization Berbasis KVM. *Jurnal Politeknik Caltex Riau*, 9(1), 11–18.
- Subhi Basit, M., Pratama, A., Firizqi, J. D., & Indrajit, R. E. (2023). Implementation of Container Orchestrator Management in Learning Management System. *Jurnal Teknik Informatika (Jutif)*, 4(4), 941–951. <https://doi.org/10.52436/1.jutif.2023.4.4.1289>
- Tanuwijaya, A., Palit, H. N., & Noertjahyana, A. (2021). Penerapan Microservices dan Amazon Elastic Container Service untuk Mendukung Scalability. *Jurnal Infra*, 9(2), 220–226.