

Pengembangan Back-End Service Aplikasi Automatisasi Ci/Cd Dan Dev-Ops Untuk Klod.Id

1st Mohammad Nowaf
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

znowaf@student.telkomuniversity.ac.id

2nd Muhammad Rizqy Alfari
Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

mrizkyalfari@telkomuniversity.ac.id

3rd Muhammad Purwadi
Chief Operating Officer
PT Tabel Data Informatika
Bandung, Indonesia

purwadi@tabeldata.com

Abstrak — Klod.id, sebuah aplikasi *Software as a Service* (SaaS), menghadapi tantangan pengembangan produk dan layanan yang terus berkembang di era cloud. Untuk mengatasi tantangan ini, Klod.id menerapkan pendekatan yang efisien dan efektif pada produknya dengan mengimplementasikan sistem otomatisasi CI/CD dan menerapkan metode Dev-Ops. Sistem otomatisasi CI/CD dan metode Dev-Ops memainkan peran kunci dalam membuat Klod.id menjadi SaaS yang efisien dan dapat diandalkan. Pemantauan dan analisis digunakan untuk meningkatkan efisiensi dan efektivitas selama tahap pengembangan produk di Klod.id, sambil juga mengurangi biaya operasional di divisi Dev-Ops. Kunci keberhasilan pendekatan ini adalah fokus pada pengembangan Back-End Service Klod.id, tanpa perubahan besar pada Front-End, dengan tujuan untuk mempercepat proses pengembangan produk secara keseluruhan dan mencapai tujuan pengembangan yang lebih berkualitas. Dengan pendekatan ini, Klod.id siap menghadapi perubahan yang terus berkembang dalam dunia teknologi dan memberikan layanan yang unggul kepada pengguna SaaS-nya.

Kata kunci— Dev-Ops, CI/CD, Automatisasi, *Software as a Service* (SaaS), Back-End Service

I. PENDAHULUAN

Dev-Ops (*Development and Operation*) adalah filosofi dan metode yang digunakan untuk meningkatkan kolaborasi antara tim pengembangan dan operasi, dengan tujuan untuk meningkatkan kecepatan dalam pengiriman produk dan layanan serta meningkatkan kualitas produk yang dihasilkan.

CI/CD (*Continuous Integration and Continuous Deployment*) adalah proses otomatisasi yang digunakan untuk mengeksekusi, melakukan integrasi, dan melakukan deployment produk secara terus menerus.

Saat ini semakin besarnya kebutuhan akan adanya Dev-Ops dan CI/CD pada teknologi *cloud*. Banyak perusahaan *startup* dan instansi pemerintah sudah mulai menggunakan teknologi ini. Namun, dengan semakin banyaknya kebutuhan akan *resource* ini, perusahaan berlomba-lomba untuk mencari orang yang memiliki keahlian dalam bidang ini, sehingga *resource* tersebut menjadi semakin mahal. Selain itu, tingkat kompleksitas dalam proses *deployment* juga cukup tinggi.

Masalah ada pada kesulitan dalam mengelola dan mengoptimalkan proses pengembangan produk, serta biaya yang cukup tinggi dalam mengelola divisi atau unit Dev-Ops. Hal ini dapat menyebabkan keterlambatan dalam proses pengembangan produk dan menurunkan kualitas produk yang dihasilkan.

Untuk mengatasi masalah tersebut klod.id membuat produk yang mengimplementasikan sistem otomatisasi CI/CD dan Dev-Ops pada saat Deployment suatu aplikasi. Dengan menggunakan sistem ini, diharapkan klod.id dapat meningkatkan efisiensi dan efektivitas dalam proses pengembangan produk, serta mengurangi biaya investasi di bagian Dev-Ops. Selain itu, dengan mengadopsi metode Dev-Ops, proses pengembangan produk dapat dilakukan secara terus-menerus dan cepat, serta dapat mengurangi risiko *error* dan *downtime* sistem. Implementasi back-end service ini akan membantu klod.id dalam meningkatkan kualitas produk yang dihasilkan dan meningkatkan kepuasan pelanggan.

II. METODE

Penelitian ini menggunakan SDLC (*Software Development Life Cycle*), yaitu pendekatan sistematis untuk mengembangkan perangkat lunak. Metode ini digunakan untuk mengelola setiap tahap dalam pengembangan perangkat lunak, dari awal hingga akhir.

Tahap awal dalam pengembangan perangkat lunak adalah tahap perencanaan (*Planning*). Tahap ini bertujuan untuk mengidentifikasi kebutuhan, merumuskan tujuan, dan Menyusun rencana proyek. Rencana proyek mencakup anggaran, alokasi sumber daya, dan jadwal pelaksanaan proyek. Tahap perencanaan penting untuk memastikan bahwa pengembangan perangkat lunak berjalan sesuai rencana.

Tahap berikutnya adalah tahap analisis (*Analysis*). Tahap ini bertujuan untuk memahami kebutuhan pengguna atau pelanggan. Tahap ini mencakup identifikasi persyaratan fungsional dan non-fungsional yang harus dipenuhi oleh perangkat lunak. Analisis menyeluruh dilakukan untuk memastikan bahwa perangkat lunak yang akan dibangun memenuhi kebutuhan pengguna.

Tahap ketiga adalah tahap desain (*Design*). Tahap ini bertujuan untuk membuat rancangan teknis perangkat lunak.

Rancangan teknis ini mencakup arsitektur perangkat lunak, desain antarmuka pengguna, dan spesifikasi teknis. Rancangan teknis ini akan digunakan sebagai dasar untuk pengembangan perangkat lunak.

Tahap keempat adalah tahap pengembangan (*Development*). Tahap ini bertujuan untuk membuat perangkat lunak sesuai dengan rancangan teknis yang telah dibuat. Tahap ini melibatkan kolaborasi antara anggota tim pengembang untuk memastikan bahwa perangkat lunak berkualitas tinggi.

Setelah pengembangan selesai, perangkat lunak diuji secara menyeluruh dalam tahap pengujian (*Testing*). Tahap ini bertujuan untuk memastikan bahwa perangkat lunak memenuhi semua persyaratan dan berfungsi sesuai yang diharapkan. Pengujian dilakukan mulai dari unit tes hingga pengujian sistem.

Tahap selanjutnya adalah tahap implementasi (*Deployment*). Tahap ini bertujuan untuk meluncurkan perangkat lunak ke pengguna akhir atau pelanggan. Tahap ini membutuhkan perencanaan yang matang, koordinasi, dan pemantauan ketat untuk meminimalkan gangguan.

Tahap terakhir adalah tahap pemeliharaan (*Maintenance*). Tahap ini bertujuan untuk menjaga agar perangkat lunak tetap relevan dan berkinerja baik. Tahap ini melibatkan pemantauan terus-menerus dan respons yang cepat terhadap perubahan dan tantangan yang mungkin timbul.

Metode SDLC adalah metode penting untuk manajemen proyek perangkat lunak. Metode ini membantu memastikan bahwa pengembangan perangkat lunak berjalan terstruktur, efisien, dan sesuai dengan tujuan proyek. Metode ini juga membantu dalam manajemen risiko dan pengendalian kualitas selama siklus hidup perangkat lunak.

III. ANALISIS

A. Deskripsi Pekerjaan

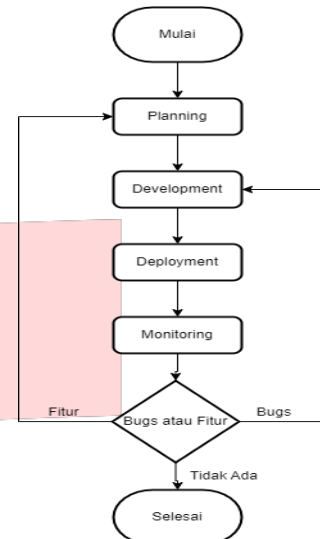
Backend developer memiliki peran krusial dalam proyek implementasi sistem otomatisasi CI/CD dan Dev-Ops pada *backend service* aplikasi klod.id. Tugas pertama dalam tanggung jawab adalah mengimplementasikan sistem otomatisasi CI/CD dan Dev-Ops dengan memanfaatkan REST API. Ini berarti harus merancang dan mengonfigurasi alur kerja otomatisasi yang akan mengintegrasikan, menguji, dan mendeploy perubahan ke dalam aplikasi secara efisien dan konsisten.

Selanjutnya, sebagai bagian dari tanggung jawab, *backend developer* akan terlibat dalam pengembangan dan *debugging* sistem yang telah diimplementasikan. Ini melibatkan pemantauan kinerja aplikasi, mengidentifikasi masalah potensial, dan meresponsnya dengan cepat untuk memastikan bahwa aplikasi berjalan dengan optimal. Dengan memahami secara mendalam arsitektur *backend* aplikasi, dapat memastikan bahwa setiap komponen berinteraksi dengan baik dan efisien.

Selain itu, dalam proyek ini, *backend developer* juga akan memainkan peran penting dalam upaya perbaikan jika ditemukan bug atau masalah dalam sistem. Harus memiliki keterampilan dalam analisis masalah, pemecahan masalah, dan peningkatan aplikasi untuk memastikan pengguna aplikasi klod.id dapat merasakan pengalaman yang cepat dan andal.

Secara keseluruhan, *backend developer* memiliki peran kunci dalam menggarap proyek ini, menghubungkan infrastruktur dan kode aplikasi, serta memastikan bahwa *backend service* aplikasi klod.id beroperasi dengan baik, terintegrasi, dan selalu siap menghadapi perubahan dan perbaikan yang diperlukan.

B. Alur Pekerjaan



GAMBAR III.
1 Alur Pekerjaan

Planning, tim akan merencanakan proyek yang akan dikerjakan, melakukan estimasi waktu dan biaya, melakukan analisis serta menentukan prioritas fitur yang akan dikembangkan.

Development, tahap ini adalah pengembangan kode yang dilakukan oleh tim. Kode yang dikembangkan akan di-*commit* dan di-*push* ke repositori GitLab.

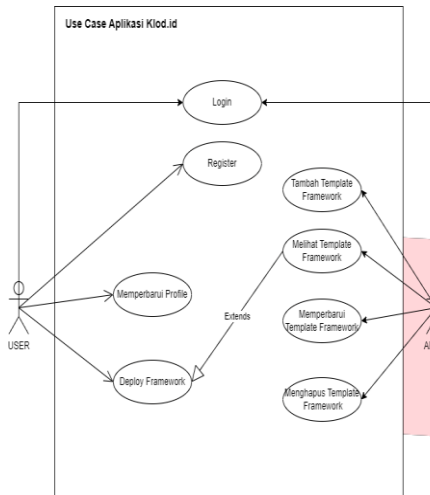
Deployment, tim akan melakukan *commit* dan *push* ke dalam repositori GitLab. Hal ini dilakukan agar kode yang sudah dibuat akan berjalan di *production*.

Monitoring, pada tahap ini tim akan memantau kinerja aplikasi dan memastikan aplikasi berjalan dengan baik. Jika ditemukan ada *bug* atau tambahan fitur, maka proses akan berulang Kembali. Jika ditemukannya ada *bug*, maka akan Kembali ke tahap *Development*. Jika adanya tambahan fitur, maka akan Kembali ke tahap *Planning*.

C. Analisis Sistem

Aplikasi Klod.id adalah aplikasi yang jenisnya adalah SaaS (*Software-as-a-Service*) yaitu aplikasi yang menyediakan layanan berbasis cloud. Aplikasi tersebut adalah meng-*host*-kan layanan aplikasi sesuai permintaan dari pelanggan ke dalam *cloud*. Mirip dengan aplikasi yang sudah ada, yaitu Heroku dan Vercel. Aplikasi tersebut akan menyimpan dan menjalankan kode di dalam *cloud* yang nantinya bisa diakses.

Dalam sistem ini, terdapat dua jenis pengguna yang memiliki peran dan use case masing-masing. Pertama, untuk *admin*, use case-nya melibatkan melakukan *login* akun serta mengelola CRUD (*Create, Read, Update, Delete*) untuk *framework* yang digunakan dalam aplikasi. *Admin* bertanggung jawab untuk mengontrol dan mengelola *framework* yang akan digunakan oleh pengguna *User*.



GAMBAR III. 2

Use Case Klod.id

Sementara itu, bagi *user* atau pengguna lainnya, mereka juga perlu melakukan *login* akun untuk mengakses sistem. Selain itu, mereka memiliki beberapa *use case* tambahan, seperti melakukan pendaftaran akun baru, memperbarui profil mereka, termasuk informasi seperti nama, *password*, dan alamat email. Selain itu, *user* juga memiliki *use case* untuk mendeploy *framework* sesuai dengan kebutuhan aplikasi yang mereka kelola.

Dengan demikian, sistem ini dirancang untuk memberikan pengalaman yang sesuai dengan peran masing-masing pengguna, dengan *admin* berfokus pada manajemen *framework* dan pengguna *user* memiliki akses untuk mengelola akun dan melakukan *deploy framework* yang relevan dengan aplikasi yang mereka kelola.

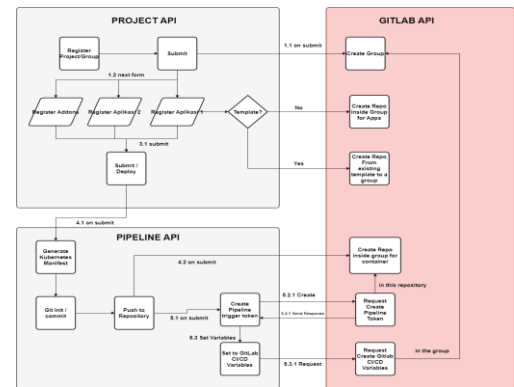
IV. HASIL DAN PEMBAHASAN

A. Analisis Hasil Pelaksanaan Kerja

Klod.id adalah aplikasi yang jenis bisnisnya adalah SaaS (*Software as a Service*), artinya Klod.id merupakan Hosting untuk Frontend atau Backend.

Dalam pengembangan sebuah aplikasi atau website, biasanya dibutuhkan sisi *Backend*. *Backend* berfungsi untuk memproses, memverifikasi, menjaga agar data client aman ketika mengirim ke *server*. Oleh sebab itu, Klod.id membutuhkan sisi *Backend* agar aplikasi yang dipilih oleh User bisa di-Hosting-kan dengan implementasi CI/CD pada sisi *Backend*nya.

Pada gambar IV. 1 ada dua warna, putih dan merah muda. Warna putih dimaksudkan proses yang dijalankan pada sisi *Backend* yang kami buat. Warna merah muda dimaksudkan adalah proses yang dijalankan oleh *server* GitLab.

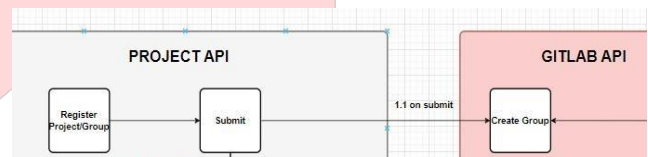


GAMBAR IV.

1 Alur Deploymen

1. Membuat Grup

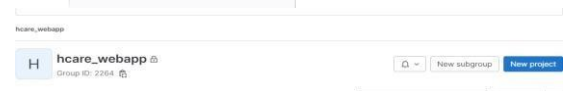
Pada tahap ini, user akan diminta membuat grup. Grup tersebut memiliki fungsi untuk menampung beberapa aplikasi yang telah dibuat nantinya oleh si user.



GAMBAR IV. 2

Alur Membuat Grup

Dibalik prosesnya membuat grup, *user* diwajibkan menamai grupnya. Ada dua proses ketika *user* selesai menamai grupnya, yaitu membuat grup di web klod.id dan membuat grup pula di dalam GitLabnya.



GAMBAR IV. 3

Hasil di GitLab

Pada gambar IV.3 adalah contoh berhasilnya ketika *user* membuat grup. Seperti yang sudah dijelaskan sebelumnya, grup dibuat pula di dalam GitLab.

LIST OF PROJECT			
No.	Project Name	Create Date	Action
1.	Project A.	28 - Aug - 2022	
2.	Project B.	28 - Aug - 2022	
3.	Project C.	28 - Aug - 2022	
4.	Project D.	28 - Aug - 2022	

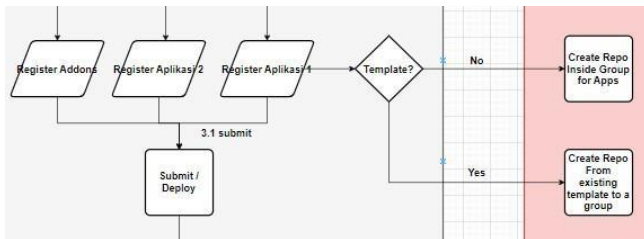
GAMBAR IV. 4

Hasil di Web Klod.id

Pada gambar IV.4 adalah contoh ketika pembuatan grup berhasil di dalam web Klod.id. Disini terdapat list nantinya ketika *user* membuat grup lebih dari satu.

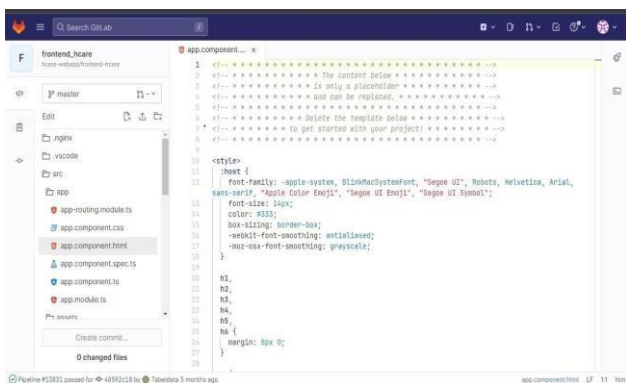
2. Membuat Aplikasi

Pada tahap kedua, setelah user membuat grup. User akan diberi pilihan untuk men-deploy aplikasinya, yaitu dengan memilih template *framework* yang telah tersedia atau *framework* yang telah user buat sendiri. Tetapi pada sistem sekarang, user hanya bisa memilih *framework* yang telah tersedia.



GAMBAR IV. 5
Alur Membuat Aplikasi

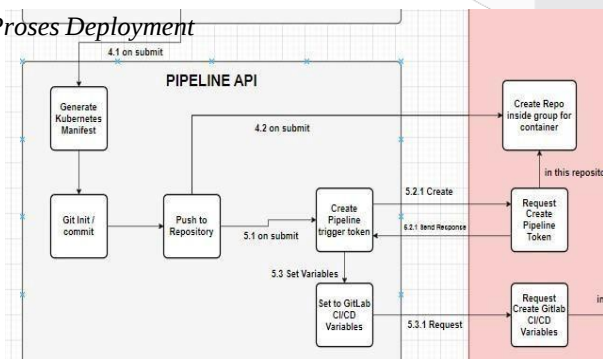
Pada tahap ini sama seperti tahap pertama, ada dua proses yaitu membuat aplikasi di web kلود.id dan membuat aplikasi pula di GitLab, namun penamaan aplikasi di dalam GitLab diubah menjadi repository. Lalu apa yang di dalam repository tersebut? Yaitu source code sesuai pilihan dari user dan di-push ke dalam repository. Repository ini disimpan didalam grup yang telah dibuat tadi sebelumnya.



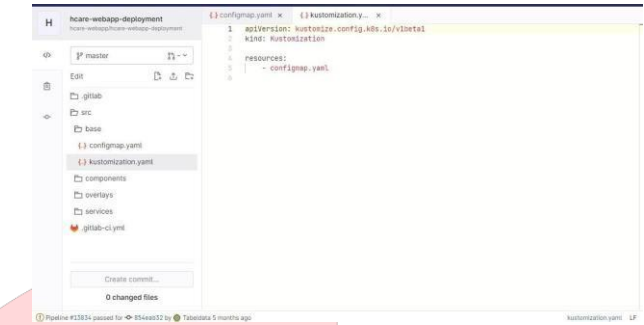
GAMBAR IV. 6
Hasil generate

Pada gambar IV.6 adalah hasil *source code* yang telah di-generate otomatis dan di-push ke repository di dalam grup yang telah dibuat sebelumnya.

3. Proses Deployment



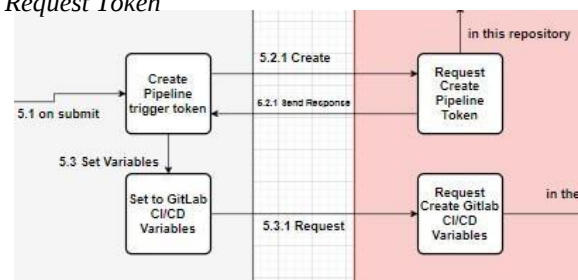
GAMBAR IV. 7
Proses deployment



GAMBAR IV. 8
Contoh hasil files kubernetes

Gambar diatas adalah hasil *files* kubernetes yang telah selesai di-generate secara otomatis. Setelah tahap ini selesai. Tahap selanjutnya adalah membuat token, token ini diambil dari repository deployment.

4. Request Token



GAMBAR IV. 9
Alur Request Token

Pada tahap ini token diambil dari repository *deployment* dan setelah token diambil, tahap selanjutnya mengambil response *token* tersebut untuk nantinya digunakan pada tahap me-set *CI/CD Variables*.

Token yang telah diambil akan digunakan untuk set *CI/CD Variables*. Dimana letak kita untuk men-set nya? Yaitu di dalam grup yang telah kita buat sebelumnya.

Variable	Value	Expanded	Default	Created
CL_DEFAULT_TRIGGER_BRANCH	master	Expanded	All (default)	
CL_ENV_REVIEW_ENABLED	true	Expanded	All (default)	
CL_TRIGGER_DEPLOY_PROJECT_ID	1290	Expanded	All (default)	
CL_TRIGGER_DEPLOY_TOKEN	glpitt-fabb...	Expanded	All (default)	

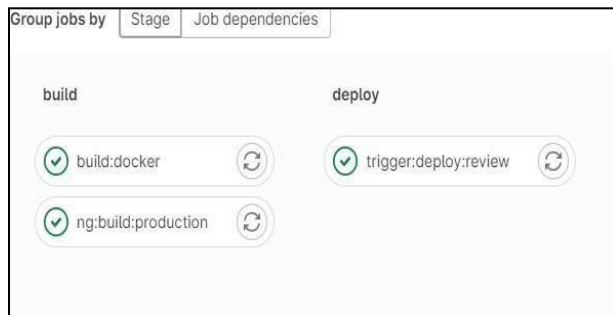
GAMBAR IV. 10
Hasil set token

Pada gambar IV.10 adalah hasil *set* token yang sudah dibuat sebelumnya ke dalam grup. Token yang dihasilkan disimpan di `CI_TRIGGER_DEPLOY_TOKEN`.

5. Yang Terjadi di GitLab

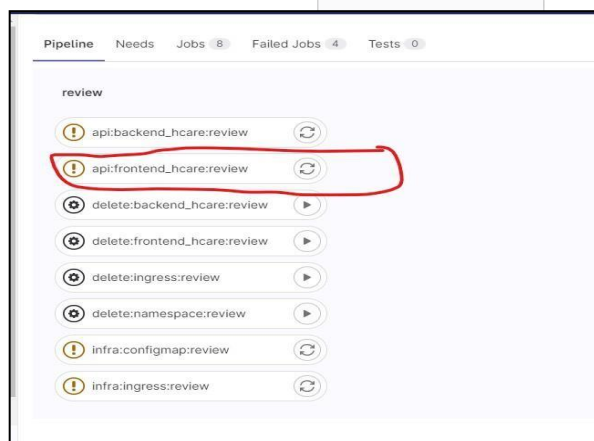
Sebenarnya apa yang terjadi di dalam GitLab Ketika proses- proses sudah dilalui, berikut adalah prosesnya:

Jalannya CI/CD di Repository, setelah sebelumnya otomatis membuat beberapa proses, di dalam GitLab yang pertama dilakukan adalah bagian **build** yaitu membuild docker dan membuild aplikasi, lalu setelah aplikasi di **build**, hasilnya akan di-deploy yang artinya dijalankan didalam server atau **cloud**.



GAMBAR IV. 11
CI/CD

Jalannya *Service*, Setelah tahap sebelumnya selesai, tahap selanjutnya adalah otomatis menjalankan servicenya. Lalu setelah hal tersebut terjadi maka aplikasi yang sudah dipilih oleh *user* akan bisa diakses di internet.



GAMBAR IV. 12
Service

V. KESIMPULAN

Dapat disimpulkan bahwa implementasi sistem otomatisasi CI/CD dan Dev-Ops pada Klod.id dapat meningkatkan efisiensi dan efektivitas dalam proses pengembangan produk, serta mengurangi biaya investasi di bagian Dev-Ops. Ini akan membantu Klod.id dalam meningkatkan kualitas produk yang dihasilkan dan meningkatkan kepuasan pelanggan. Namun, perlu diingat bahwa implementasi sistem otomatisasi CI/CD dan Dev-Ops juga memerlukan pengelolaan yang baik dan optimal

- [1] Keyur Talati, "The Rising Demand for DevSecOps Talent", 15 April 2022. [Online]. Available: <https://devops.com/the-rising-demand-for-devsecops-talent/>
- [2] Hossein Ashtari, "CI/CD vs. DevOps: Understanding 8 Key Differences", 7 April 2022.[Online].Available: <https://www.spiceworks.com/tech/devops/articles/cicd-vs-devops/>
- [3] Chnar, M., & Subhi, Z. (2021). Sufficient Comparison Among Cloud Computing Services: IaaS, PaaS, and SaaS: A Review. Science and Business, 5(2), 17–30. <https://doi.org/10.5281/zenodo.445012>
- [4] Ardiansyah, H., & Fatwanto, A. (2022). Comparison of Memory Usage between REST API in Javascript and Golang. 22(1), 229–240. <https://doi.org/10.30812/matrik.v22i1.1325>
- [5] Golightly, L., Chang, V., Xu, Q. A., Gao, X., & Liu, B. S. C. (2022). Adoption of cloud computing as innovation in the organization. International Journal of Engineering Business Management, 14, 1–17. <https://doi.org/10.1177/18479790221093992>
- [6] Putra, R. A. (2018). Analisa Implementasi Arsitektur Microservices Berbasis Kontainer Pada Komunitas Pengembang Perangkat Lunak Sumber Terbuka (OpenDayLight DevOps Community). Jurnal Sistem Informasi Teknologi Informasi Dan Komputer (Just It) Universitas Bina Nusantara Magister Manajemen Sistem Informasi Jakarta, 150–162.
- [7] Toba, H., Kandaga, T., #2, G., Narabel, J., Widjaja, A., Ferdian, S., & +5, S. (2022). JEPIN (Jurnal Edukasi dan Penelitian Informatika) Evaluasi Metodologi CI/CD untuk Pengembangan Perangkat Lunak dalam Perkuliahan. 8(2), 227–234.
- [8] Kroons, A. A., & Dewi, C. (2023). Jurnal Indonesia: Manajemen Informatika dan Komunikasi PENGEMBANGAN DASHBOARD TRIVY BERBASIS WEBSITE MENGGUNAKAN REACT JS DAN GOLANG Jurnal Indonesia: Manajemen Informatika dan Komunikasi. 4(3), 1037–1049.
- [10] Suwarno, & Yulandi, A. P. (2023). Analisis Performa Backend Framework: Studi Komparasi

- Framework Golang dan Node.js. Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK), 8(1), 155–168. <https://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- [11] About Us (2023) Tabel Data Informatika. Available at: <https://web.tabeldata.com/about-us/>
- [12] Lai, ST., Susanto, H., Leu, FY. (2022). Combining Pipeline Quality with Automation to CI/CD Process for Improving Quality and Efficiency of Software Maintenance. In: Barolli, L., Yim, K., Chen, HC. (eds) Innovative Mobile and Internet Services in Ubiquitous Computing. IMIS 2021. Lecture Notes in Networks and Systems, vol 279. Springer, Cham. https://doi.org/10.1007/978-3-030-79728-7_36
- [13] J. Fluri, F. Fornari and E. Pustulka, "Measuring the Benefits of CI/CD Practices for Database Application Development," 2023 IEEE/ACM International Conference on Software and System Processes (ICSSP), Melbourne, Australia, 2023, pp. 46-57, doi: 10.1109/ICSSP59042.2023.00015.
- [14] Natanael, Y. (2022) Gitlab: Pengertian, Fungsi Dan Kelebihan, DosenIT.com. Available at: <https://dosenit.com/software/gitlab>
- [15] Sekarputri, N. (2023) Terlengkap: Panduan menggunakan postman api, Software House & System Integrator di Malang, Indonesia. Available at: <https://www.sekawanmedia.co.id/blog/postman-api/>