

Implementasi CI/CD dalam Rancang Bangun Automated Final Project Deployment System untuk Mahasiswa D4 Manajemen Informatika

Muhammad Ainur Rofik¹, I Gde Agung Sri Sidhimantra²,

Program Studi Manajemen Informatika, Universitas Negeri Surabaya
Jl. Ketintang, Ketintang, Kec. Gayungan, Surabaya, Jawa Timur 60231

¹muhammadainur.22005@mhs.unesa.ac.id

²igdesidhimantra@unesa.ac.id

Abstrak - Pengelolaan produk tugas akhir mahasiswa pada Program Studi D4 Manajemen Informatika Universitas Negeri Surabaya saat ini masih terbatas pada penyimpanan arsip statis di Google Drive, sehingga aplikasi yang dihasilkan tidak dapat dieksekusi secara langsung dan kehilangan nilai fungsional sebagai sistem aktif. Permasalahan ini berdampak pada rendahnya efektivitas inventarisasi, evaluasi, serta publikasi portofolio digital institusi. Penelitian ini bertujuan untuk merancang dan mengimplementasikan Automated Final Project Deployment System berbasis web yang mampu mengotomatisasi proses validasi, konfigurasi lingkungan aplikasi, dan publikasi secara terintegrasi. Metode yang digunakan adalah Rapid Application Development (RAD) yang diintegrasikan dengan pendekatan Continuous Integration dan Continuous Deployment (CI/CD). Sistem dikembangkan menggunakan Laravel, MySQL, dan Nginx dengan dukungan konfigurasi virtual host dinamis. Hasil pengujian menggunakan black-box testing menunjukkan seluruh fungsionalitas berjalan sesuai spesifikasi. Implementasi sistem terbukti meningkatkan efisiensi, mempercepat akses aplikasi daring, serta mengoptimalkan pemanfaatan infrastruktur server program studi signifikan dalam mendukung pengelolaan portofolio digital yang berkelanjutan dan terintegrasi dengan baik.

Kata kunci : Automated, Deployment, Continuous Integration dan Continuous Deployment, Rapid Application Development, Nginx.

Abstract - The management of final project products of students in the D4 Informatics Management Study Program at Surabaya State University is currently still limited to static archive storage in Google Drive, so that the resulting application cannot be executed directly and loses its functional value as an active system. This problem has an impact on the low effectiveness of the inventory, evaluation, and publication of the institution's digital portfolio. This study aims to design and implement a web-based Automated Final Project Deployment System that is able to automate the validation process, application environment configuration, and publication in an integrated manner. The method used is Rapid Application Development (RAD) integrated with the Continuous Integration and Continuous Deployment (CI/CD) approach. The system was developed using Laravel, MySQL, and Nginx with the support of dynamic virtual host configuration. Test results using black-box testing show that all functionality runs according to specifications. The system implementation has been proven to increase efficiency, accelerate online application access, and optimize the utilization of the study program's server infrastructure

significantly in supporting sustainable and well-integrated digital portfolio management.

Keywords : Automated, Deployment, Continuous Integration dan Continuous Deployment, Rapid Application Development, Nginx.

I. PENDAHULUAN

Transformasi digital dalam dunia pendidikan tinggi tidak hanya menuntut institusi beradaptasi pada aspek administratif, tetapi juga mengelola dan menjaga keberlanjutan produk akademik sebagai aset digital. Perguruan tinggi perlu menyediakan ekosistem yang mendukung pengelolaan karya mahasiswa secara terstruktur dan mudah diakses. Pada Program Studi D4 Manajemen Informatika, mayoritas luaran tugas akhir mahasiswa berupa aplikasi berbasis web dari hasil pembelajaran berbasis proyek. Pemilihan teknologi web ini didasarkan pada tingkat kompatibilitasnya di berbagai perangkat serta kemudahan akses tanpa perlu instalasi tambahan pada komputer pengguna [15]. Aplikasi tersebut menjadi representasi kemampuan mahasiswa merancang solusi digital untuk kebutuhan industri. Namun, pengelolaan karya tersebut masih sangat terbatas pada proses pengumpulan dan penyimpanan arsip statis tanpa adanya manajemen pengelolaan lanjutan.

Praktik pengarsipan pasif ini menyebabkan aplikasi tidak dapat dijalankan secara langsung, sehingga kehilangan siklus hidup sebagai sistem aktif [3]. Produk akhir hanya berupa format arsip dan basis data, sehingga tidak bisa diakses secara praktis. Kondisi ini secara langsung berdampak pada sulitnya proses demonstrasi, evaluasi, dan publikasi portofolio digital institusi. Keterbatasan akses terhadap aplikasi sangat menghambat pengembangan lanjutan dan pemanfaatan ulang oleh mahasiswa lain sebagai referensi pembelajaran. Akibatnya, luaran tugas akhir yang memiliki nilai implementatif tinggi hanya menjadi arsip penyimpanan mati yang sangat jarang dimanfaatkan kembali oleh pihak kampus.

Di sisi lain, upaya mempublikasikan aplikasi melalui infrastruktur website universitas menghadapi kendala kebijakan keamanan dan keterbatasan teknis server terpusat. Kebijakan ketat mengakibatkan pihak program studi tidak memiliki hak akses penuh untuk melakukan konfigurasi server

maupun memodifikasi kode secara mandiri [19]. Proses publikasi aplikasi menjadi tidak fleksibel. Ironisnya, program studi sebenarnya memiliki fasilitas server fisik mandiri yang belum termanfaatkan optimal untuk kebutuhan hosting. Ketiadaan sistem otomatisasi terstandarisasi menyebabkan server tersebut tidak berfungsi maksimal. Padahal, proses *deployment* manual rentan terhadap kesalahan teknis, membutuhkan intervensi manusia pada setiap langkah, dan memakan waktu [21]. Jika publikasi dilakukan manual dalam skala besar, risiko kesalahan konfigurasi, bentrokan antar aplikasi, dan beban administratif laboratorium akan sangat meningkat setiap semester.

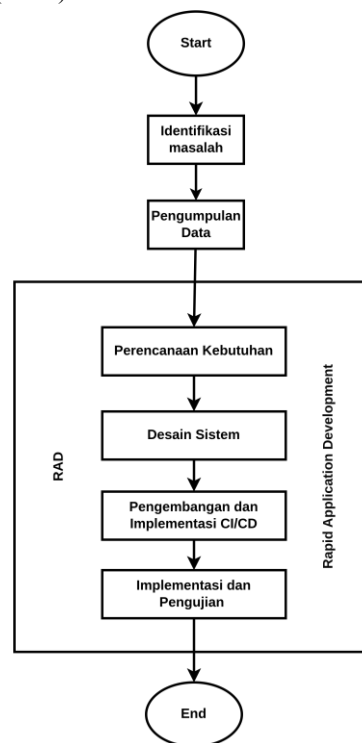
Untuk mengatasi masalah kompleks tersebut, diperlukan sebuah pendekatan strategis yang mampu mengintegrasikan kebutuhan akademik dengan pengelolaan server secara terstruktur. Pendekatan relevan adalah penerapan konsep *Continuous Integration* dan *Continuous Deployment (CI/CD)*. Sebagaimana ditunjukkan oleh Wijaya dkk. (2024), konsep ini sangat efektif diadaptasi untuk mengotomatisasi pengujian, integrasi kode, dan menerapkan hasil integrasi ke lingkungan server [31]. Melalui mekanisme ini, sistem dapat mengekstrak berkas, analisis struktur proyek, konfigurasi lingkungan aplikasi, pembuatan basis data, hingga aktivasi subdomain publik secara otomatis tanpa intervensi manual. Penerapan sistem otomatisasi ini terbukti mampu berjalan konsisten, meminimalkan risiko kesalahan manusia, serta menurunkan beban kerja administrator sistem akibat perbedaan lingkungan [11]. Hasil pengiriman sistem otomatisasi ini juga menghasilkan kualitas *deployment* yang terstandar dan aman digunakan dalam lingkup akademik [13].

Berdasarkan latar belakang tersebut, penelitian ini difokuskan pada perancangan dan pengembangan *Automated Final Project Deployment System* berbasis web menggunakan metode *Rapid Application Development (RAD)*. Metode RAD terbukti efektif untuk pengembangan sistem yang bersifat *customize* dan diselesaikan dalam siklus waktu singkat [2]. Penerapan metode ini dinilai sangat sesuai karena memungkinkan komponen sistem dikembangkan secara bertahap dan tersampaikan lebih cepat kepada pengguna akhir melalui siklus prototipe tanpa harus menunggu semua fitur selesai dibangun pada fase awal [18]. Sistem perangkat lunak manajemen otomatis yang dikembangkan akan mengintegrasikan antarmuka web dengan mekanisme otomatisasi tingkat lanjut di sisi server fisik mandiri melalui penerapan konsep CI/CD. Kehadiran inovasi sistem ini diharapkan dapat mempermudah operasional publikasi, memberdayakan infrastruktur server mandiri program studi secara optimal, serta mendukung terbentuknya ekosistem portofolio digital yang aktif, fungsional, dan berkelanjutan [25].

II. METODE PENELITIAN

Berdasarkan permasalahan pengelolaan produk tugas akhir pada Program Studi D4 Manajemen Informatika UNESA yang telah diuraikan pada bagian pendahuluan, penelitian ini dirancang untuk menghasilkan sistem publikasi karya mahasiswa yang aktif, otomatis, dan terpusat. Oleh karena itu, metode penelitian difokuskan pada implementasi konsep

Continuous Integration dan *Continuous Deployment (CI/CD)* di lingkungan server lokal serta pengembangan sistem interaktif berbasis web dengan pendekatan *Rapid Application Development (RAD)*.



Gambar 1 Alur Penelitian

A. Identifikasi Masalah

Identifikasi masalah dalam penelitian ini berfokus pada mekanisme pengarsipan produk tugas akhir di Program Studi D4 Manajemen Informatika Universitas Negeri Surabaya, yang saat ini masih terbatas pada penyimpanan berkas statis di Google Drive. Kondisi tersebut menyebabkan aplikasi yang dihasilkan tidak dapat dijalankan secara langsung dan kehilangan fungsi sebagai sistem yang aktif [3]. Permasalahan ini diperparah oleh pengelolaan repositori yang kurang informatif serta keterbatasan infrastruktur website universitas yang membatasi akses teknis untuk pengembangan fitur otomatisasi secara mandiri [19]. Di sisi lain, ketersediaan server fisik pada tingkat program studi belum dimanfaatkan secara optimal karena belum adanya sistem manajemen yang mampu mendukung hosting aplikasi mahasiswa secara terpusat dan terstandarisasi [13]. Oleh karena itu, penelitian ini dilakukan untuk mengatasi permasalahan tersebut melalui pembangunan *Automated Final Project Deployment System* sebagai solusi dalam mentransformasi infrastruktur server lokal menjadi ekosistem portofolio digital yang aktif, aman, dan terstandarisasi [32].

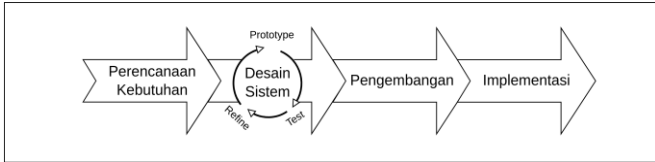
B. Pengumpulan Data

Tahap pengumpulan data dilakukan untuk membangun kerangka permasalahan yang valid melalui studi literatur, observasi, dan wawancara [33]. Studi literatur digunakan untuk memperoleh landasan teoretis terkait Nginx, Laravel,

Flask, konsep CI/CD, metode RAD, serta panel hosting seperti CyberPanel dan aaPanel. Observasi dilakukan terhadap mekanisme pengarsipan tugas akhir di Google Drive yang menunjukkan bahwa repositori hanya berfungsi sebagai penyimpanan statis dalam format .zip dan .sql tanpa kemampuan menjalankan aplikasi secara langsung (*live demo*).

Sebagai pelengkap, wawancara dilakukan bersama Dodik Arwin Dermawan, S.ST., S.T., M.T. selaku Koordinator Program Studi D4 Manajemen Informatika. Hasil wawancara menunjukkan bahwa publikasi karya mahasiswa melalui Gallery MI terkendala karena website program studi menggunakan sistem terpusat universitas yang membatasi modifikasi *backend* dan penambahan fitur otomatisasi. Selain itu, server fisik program studi belum dimanfaatkan optimal karena belum tersedia sistem manajemen *deployment* otomatis. Oleh karena itu, diperlukan sistem otomatis yang mampu mendukung *deployment* aplikasi mahasiswa secara mandiri dan terstandarisasi.

C. Perencanaan Kebutuhan



Gambar 2 Metode Rapid Application Development (RAD)

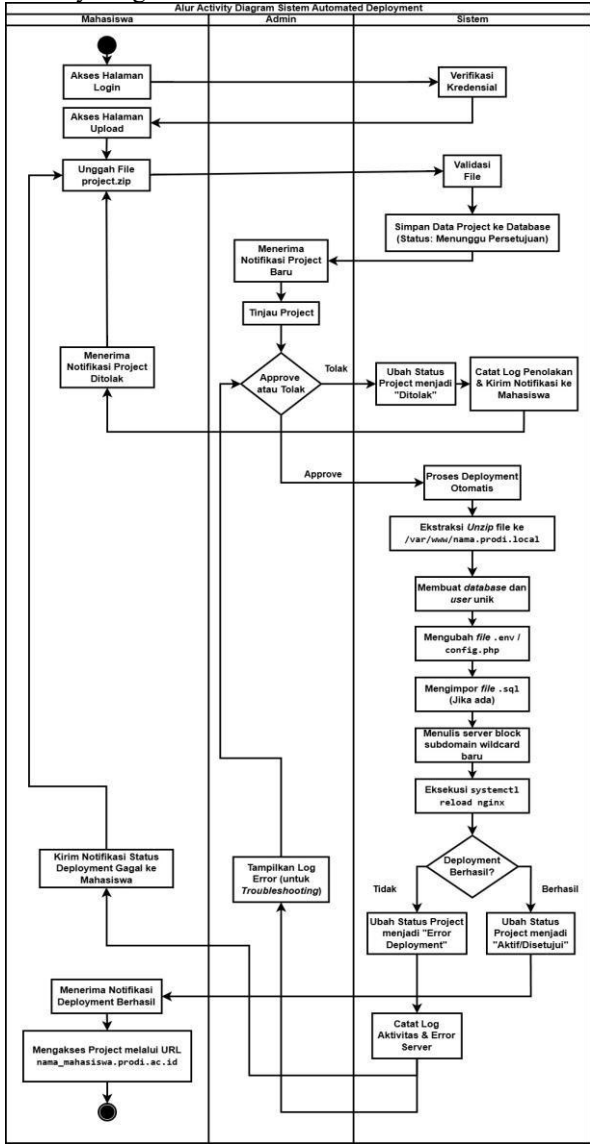
Tahap ini merupakan tahap awal dalam metode *Rapid Application Development* (RAD) yang berfokus pada identifikasi kebutuhan sistem sebagai dasar pengembangan aplikasi secara iteratif dan cepat [24]. Pada tahap ini dilakukan analisis kebutuhan pengguna, permasalahan sistem, serta spesifikasi teknis yang diperlukan agar proses pengembangan dapat berjalan sesuai tujuan penelitian. Tahap ini bertujuan mengidentifikasi spesifikasi sistem untuk mentransformasi penyimpanan arsip statis menjadi ekosistem aplikasi yang dapat diakses secara langsung (*live deployment*). Berdasarkan analisis permasalahan, kebutuhan sistem dirumuskan ke dalam aspek fungsional dan non-fungsional. Secara fungsional, sistem dituntut mampu mengelola pengguna *multi-level* [16], menerima unggahan format berkas standar (.zip dan .sql), serta menjalankan otomatisasi *deployment* pasca persetujuan admin mulai dari ekstraksi berkas, penyiapan lingkungan aplikasi (PHP/Python), konfigurasi database dinamis, hingga publikasi ke subdomain Nginx. Selain itu, sistem harus dilengkapi fitur pembersihan sumber daya otomatis dan perekaman log audit. Sementara itu, secara non-fungsional, sistem yang berjalan pada Ubuntu Server 22.04 ini ditargetkan mampu memproses *deployment* di bawah 60 detik dengan jaminan keamanan ketat, antarmuka web yang responsif, serta stabilitas uptime 100% saat pemutusan ulang konfigurasi Nginx agar tidak mengganggu akses pada proyek mahasiswa lain [22].

D. Desain Sistem

Desain sistem merupakan tahap krusial untuk memastikan *Automated Final Project Deployment System* terintegrasi secara aman dan efisien antara sisi pengguna, aplikasi, dan

server. Desain ini memodelkan alur kerja sistem mulai dari tahap unggah produk oleh mahasiswa hingga proses *deployment* otomatis pasca persetujuan admin. Tahapan perancangan ini mencakup arsitektur sistem, use case diagram, activity diagram, ERD, dan User Flow CI/CD.

1) Activity Diagram

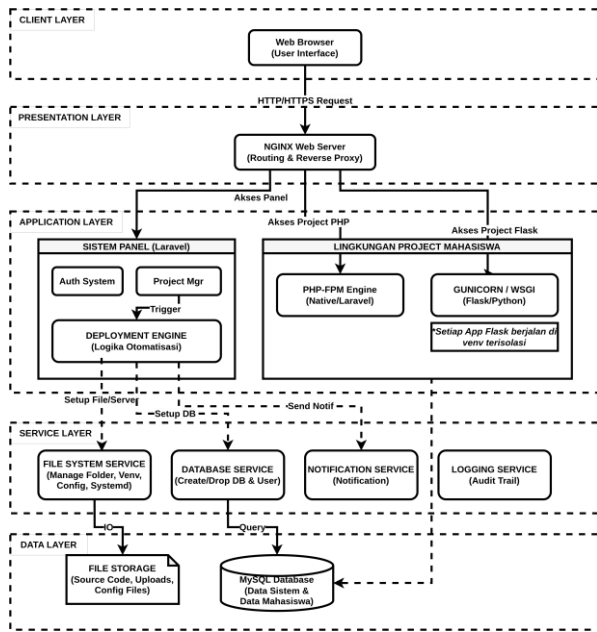


Gambar 3 Activity Diagram

Diagram aktivitas ini disusun untuk memodelkan secara komprehensif seluruh alur kerja operasional di dalam sistem, mulai dari interaksi awal pengguna hingga hasil akhir di sisi server. Pemodelan ini memvisualisasikan secara terstruktur bagaimana proses diawali oleh Mahasiswa saat melakukan unggah arsip proyek tugas akhir, yang kemudian dijumpai oleh intervensi Admin melalui tahap peninjauan dan persetujuan. Diagram ini juga merepresentasikan tahapan eksekusi teknis yang dijalankan sepenuhnya secara otomatis oleh sistem (*Deployment Engine*) setelah persetujuan, mencakup konfigurasi server hingga finalisasi akses publik, guna memastikan bahwa transisi dari

pengarsipan statis menuju penayangan aplikasi yang aktif dapat berjalan dengan efisien dan konsisten.

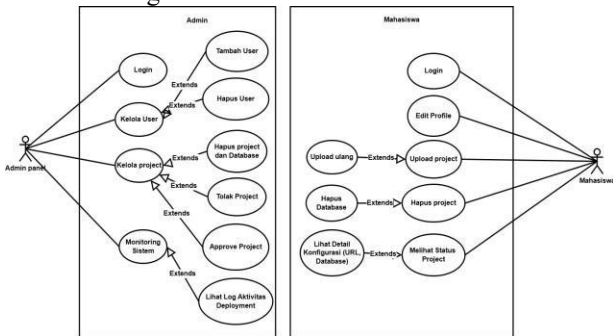
2) Arsitektur Sistem



Gambar 4 Arsitektur Sistem

Arsitektur sistem dirancang menggunakan model multi-tier untuk menjamin modularitas, keamanan, dan pemisahan tanggung jawab antar komponen. Arsitektur ini terdiri dari lima lapisan utama, yaitu *Client Layer* sebagai antarmuka pengguna melalui browser, *Presentation Layer* menggunakan Nginx untuk routing dan pengelolaan *virtual host*, *Application Layer* yang menangani autentikasi, manajemen proyek, dan proses *deployment* otomatis, *Service Layer* untuk pengelolaan file, database, notifikasi, dan *logging*, serta *Data Layer* sebagai penyimpanan database, file proyek, dan konfigurasi sistem. Penerapan arsitektur berlapis ini memungkinkan proses *deployment* berjalan lebih aman, terstruktur, dan mudah dipantau.

3) Use Case Diagram

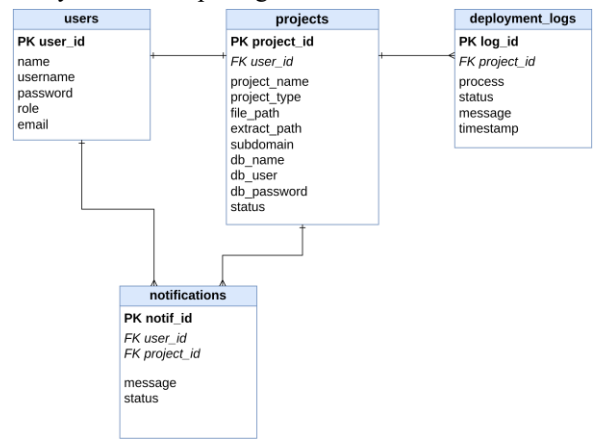


Gambar 5 Use Case Diagram

Diagram use case menampilkan dua aktor utama, yaitu Admin dan Mahasiswa. Admin memiliki hak akses untuk mengelola pengguna, proyek, serta melakukan monitoring sistem, sedangkan Mahasiswa dapat mengelola profil,

mengunggah proyek, dan melihat status *deployment* aplikasi yang diajukan.

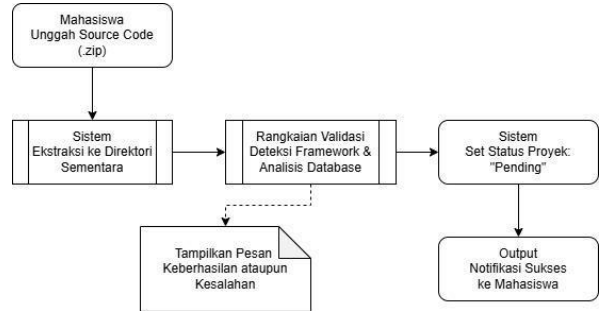
4) Entity Relationship Diagram



Gambar 6 Entity Relationship Diagram

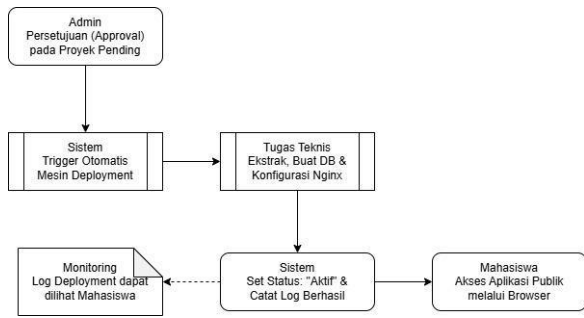
Entity Relationship Diagram (ERD) memodelkan struktur basis data sistem yang mencakup pengguna, proyek, kredensial database, log aktivitas, dan riwayat *deployment* untuk mendukung pengelolaan sistem secara terstruktur.

5) User Flow CI/CD



Gambar 7 User Flow CI

User flow ini menggambarkan pengalaman pengguna dan respon sistem saat proses validasi otomatis (CI) berjalan, tepat setelah file diunggah namun sebelum masuk ke tahap *deployment* (CD). Proses *Continuous Integration* (CI) dimulai secara otomatis sesaat setelah mahasiswa mengunggah file *source code* berformat *.zip* melalui antarmuka sistem. Sistem kemudian mengekstrak file tersebut ke direktori sementara untuk menjalankan rangkaian validasi bertingkat, yang mencakup deteksi jenis *framework* (Laravel, PHP Native, atau Flask), serta analisis kebutuhan database melalui kredensial yang telah dibuat. Jika seluruh kriteria teknis terpenuhi, sistem akan menandai status proyek menjadi "Pending" dan mengirimkan notifikasi sukses, dan jika ditemukan ketidaksesuaian struktur atau format, proses akan langsung dihentikan dan sistem menampilkan pesan kesalahan agar mahasiswa dapat segera melakukan perbaikan.



Gambar 8 User Flow CD

User flow ini memodelkan alur *Continuous Deployment* (CD) diawali oleh tindakan Admin yang memberikan persetujuan (*approval*) pada proyek mahasiswa yang berstatus pending melalui dasbor sistem. Aksi ini secara otomatis memicu mesin *deployment* di server untuk menjalankan serangkaian tugas teknis tanpa intervensi manual, mulai dari ekstraksi kode sumber, pembuatan database, hingga pembuatan file konfigurasi pada Nginx. Setelah seluruh proses di latar belakang selesai, sistem akan memperbarui status proyek menjadi "Aktif" dan mengirimkan catatan berhasil pada *log deployment* kepada mahasiswa, sehingga mereka dapat langsung mengakses aplikasi tugas akhir mereka secara publik melalui browser.

E. Pengembangan

Tahap pengembangan sistem merupakan fase konstruksi dalam metode RAD, di mana desain diimplementasikan menjadi aplikasi secara iteratif. Sistem dikembangkan menggunakan Ubuntu Server, Nginx, MySQL, PHP, dan *framework* Laravel dengan dukungan skrip otomatisasi server. Pada tahap ini, konsep CI/CD diterapkan untuk melakukan validasi awal, deteksi aplikasi, analisis database, hingga proses *deployment* otomatis seperti konfigurasi server, database, *virtual host*, dan aktivasi subdomain tanpa intervensi manual [26].

F. Implementasi

Tahap implementasi akhir (*cutover*) merupakan proses penyatuan seluruh modul aplikasi, skrip otomatisasi, dan konfigurasi server ke dalam lingkungan produksi internal program studi. Pada tahap ini, alur CI/CD diuji secara menyeluruh melalui pengujian fungsional menggunakan metode *black-box testing* serta pengujian integrasi untuk memastikan sinkronisasi antara antarmuka sistem, skrip *backend*, dan *routing* Nginx berjalan dengan baik tanpa bentrok antar proyek mahasiswa [28].

III. HASIL DAN PEMBAHASAN

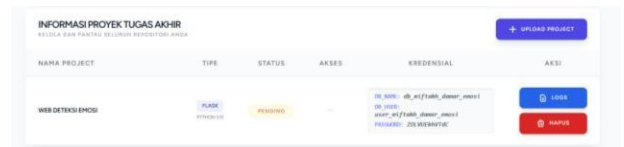
Bagian ini menguraikan hasil implementasi dari *Automated Final Project Deployment System* yang dikembangkan menggunakan metode *Rapid Application Development* (RAD) dengan penerapan konsep otomatisasi *Continuous Integration* dan *Continuous Deployment* (CI/CD). Implementasi sistem mencakup pengembangan antarmuka pengguna, integrasi modul *backend*, konfigurasi server, serta mekanisme *deployment* otomatis untuk aplikasi tugas akhir mahasiswa.

Selain itu, bagian ini juga memaparkan hasil pengujian sistem yang dilakukan untuk memverifikasi fungsionalitas, integrasi, dan stabilitas sistem dalam menjalankan proses validasi hingga *deployment* aplikasi secara otomatis.

A. Implementasi Mekanisme CI/CD

Tahap ini menguraikan hasil implementasi dari mekanisme *Continuous Integration* (CI) dan *Continuous Deployment* (CD) yang dijalankan secara lokal pada server program studi. Mekanisme ini dirancang untuk memastikan bahwa setiap proyek tugas akhir mahasiswa yang diunggah ke dalam sistem dapat diproses melalui proses validasi dan *deployment* secara otomatis tanpa memerlukan konfigurasi manual dari admin [1]. Dengan adanya mekanisme ini, proses pengelolaan dan publikasi proyek dapat dilakukan secara lebih cepat, konsisten, dan terstandarisasi.

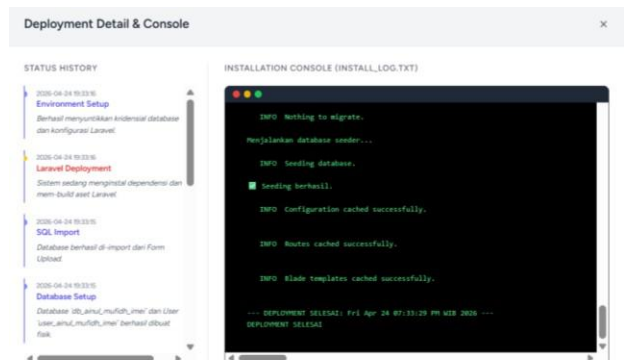
1) Hasil Eksekusi Continuous Integration (CI)



Gambar 9 Hasil Continuous Integration (CI)

Hasil implementasi menunjukkan bahwa sistem mampu melakukan analisis struktur proyek secara otomatis setelah file aplikasi diunggah oleh mahasiswa. Proses ini dilakukan untuk memastikan kelengkapan file, validitas struktur direktori, serta kompatibilitas proyek dengan mekanisme *deployment* pada server. Sistem juga mampu mengidentifikasi jenis *framework* atau aplikasi yang digunakan, seperti Laravel, PHP Native, maupun Flask, sehingga kebutuhan *deployment* dapat ditentukan secara otomatis. Selain itu, sistem dapat mendeteksi kebutuhan basis data berdasarkan file dan informasi yang diinput saat proses unggah proyek.

2) Hasil Eksekusi Continuous Deployment (CD)



Gambar 10 Hasil Continuous Deployment (CD)

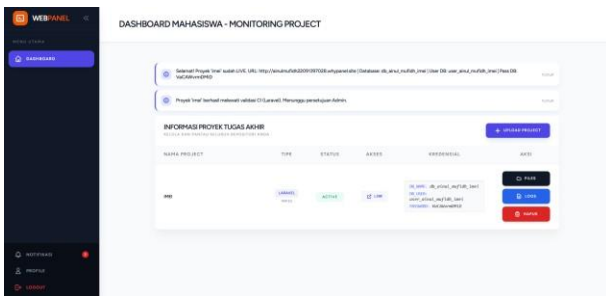
Berdasarkan hasil implementasi yang telah dilakukan, sistem *Continuous Deployment* (CD) mampu menjalankan proses *deployment* proyek mahasiswa secara otomatis setelah memperoleh persetujuan dari Admin. Mekanisme ini memungkinkan proyek yang telah lolos validasi pada tahap

CI dipublikasikan langsung ke server tanpa konfigurasi manual. Dengan demikian, proses *deployment* menjadi lebih cepat, konsisten, dan efisien dalam pengelolaan aplikasi tugas akhir mahasiswa.

B. Implementasi Sistem

Implementasi sistem pada *Automated Final Project Deployment System* dirancang untuk memudahkan mahasiswa dalam mengelola unggah dan *deployment* proyek secara mandiri melalui web. Sistem dikembangkan menggunakan Laravel sehingga dapat diakses melalui browser tanpa perlu akses langsung ke server.

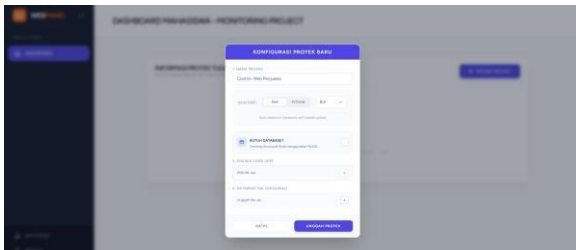
1) Tampilan Dashboard Mahasiswa



Gambar 11 Dashboard Mahasiswa

Halaman dashboard mahasiswa menampilkan informasi proyek beserta detail seperti nama proyek, *subdomain*, pemilik, *framework*, database, dan status *deployment*. Tampilan ini memudahkan mahasiswa untuk melihat informasi mengenai *deployment* proyeknya.

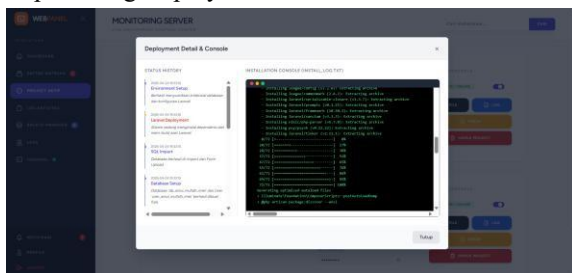
2) Tampilan Form Upload Proyek



Gambar 12 Form Upload Proyek

Tampilan ini adalah antarmuka unggah proyek mahasiswa yang berisi form nama proyek, pilihan lingkungan, serta unggahan file *.zip* dan *.sql* untuk *deployment* otomatis.

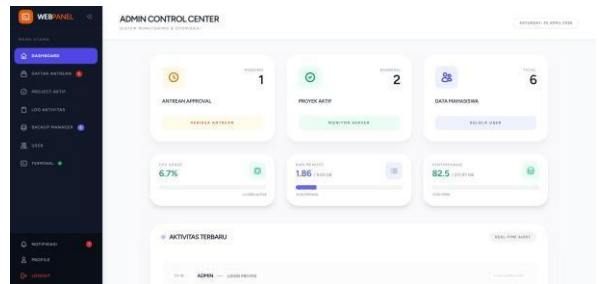
3) Tampilan Log Deployment



Gambar 13 Log Deployment

Tampilan ini adalah fitur *log deployment* yang menampilkan proses *deployment* secara transparan serta status keberhasilan atau kegagalan.

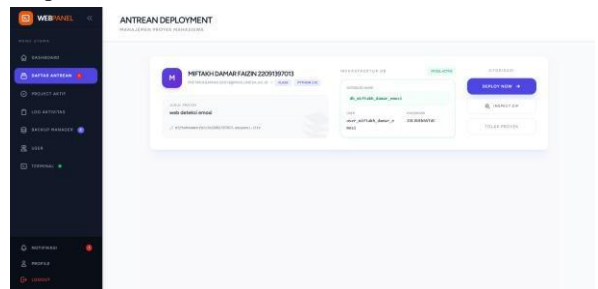
4) Tampilan Dashboard Admin



Gambar 14 Dashboard Admin

Tampilan ini adalah dashboard admin yang menampilkan jumlah antrian *deployment*, proyek aktif, dan pengguna untuk memantau sistem secara keseluruhan.

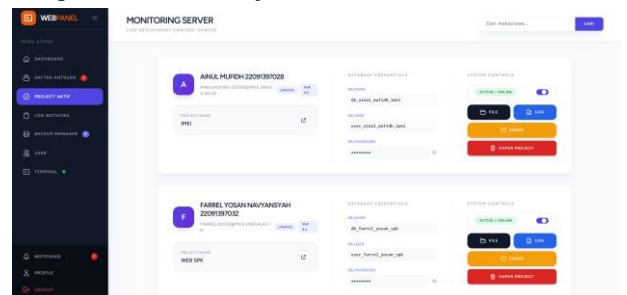
5) Tampilan Halaman Antrean



Gambar 15 Halaman Antrean

Tampilan ini adalah halaman antrian *deployment* yang menampilkan informasi proyek yang menunggu proses persetujuan admin. Halaman ini juga menyediakan beberapa aksi, seperti *deploy*, *tolak*, dan *inspect zip* untuk melihat isi file proyek secara langsung tanpa perlu mengunduhnya terlebih dahulu.

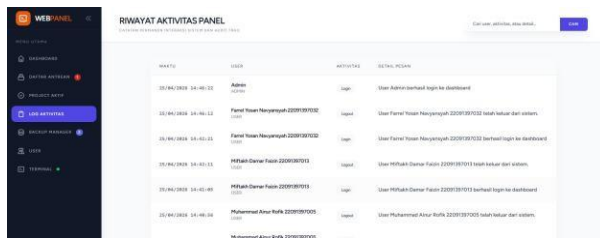
6) Tampilan Halaman Project Aktif



Gambar 16 Halaman Proyek Aktif

Tampilan ini adalah Halaman project aktif menampilkan daftar proyek yang telah berhasil dideploy beserta detail informasinya. Halaman ini juga menyediakan fitur pengelolaan seperti mengaktifkan atau menonaktifkan proyek, mengelola file, melihat *log deployment*, serta menghapus proyek jika diperlukan.

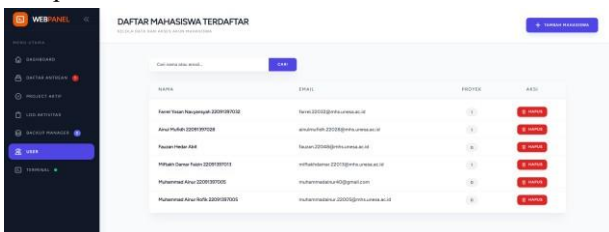
7) Tampilan Halaman Log Activity



Gambar 17 Halaman Log Activity

Tampilan ini adalah Halaman log aktivitas menampilkan riwayat aktivitas pengguna, seperti unggah proyek, persetujuan atau penolakan *deployment*, beserta waktu dan identitas pengguna, sehingga memudahkan pemantauan aktivitas sistem.

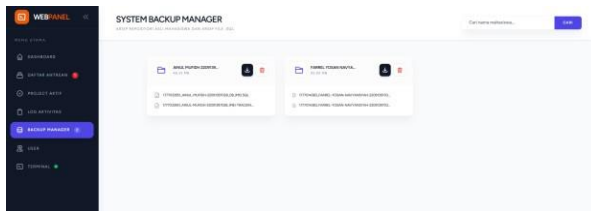
8) Tampilan Halaman Daftar User



Gambar 18 Halaman Daftar User

Tampilan ini adalah Halaman daftar user menampilkan data pengguna untuk memudahkan admin memantau dan mengelola akun. Selain itu, tersedia fitur untuk menambah dan menghapus user sesuai kebutuhan.

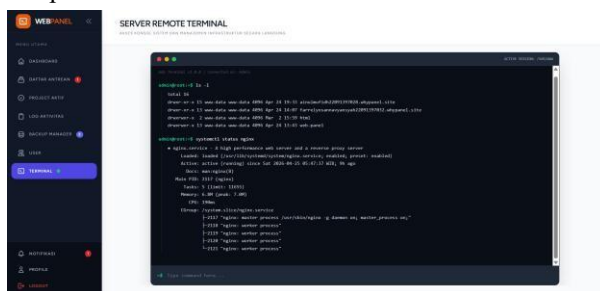
9) Tampilan Halaman Backup Manager



Gambar 19 Halaman Backup Manager

Halaman backup manager menampilkan file cadangan proyek (.zip) dan database (.sql) untuk menjaga keamanan data. Admin dapat mengunduh atau menghapus file backup sesuai kebutuhan.

10) Tampilan Halaman Terminal



Gambar 20 Halaman Terminal

Halaman terminal digunakan untuk memantau kondisi layanan server seperti Nginx melalui perintah dasar. Aksesnya dibatasi hanya untuk pemantauan, tanpa izin membuat atau mengedit file, guna menjaga keamanan sistem.

C. Pengujian Fungsional

Pengujian fungsional dilakukan menggunakan metode *black-box testing* untuk memastikan setiap fitur berjalan sesuai kebutuhan, meliputi unggah proyek, validasi CI, *deployment*, status *deployment*, penghapusan project, dan beberapa fitur lainnya.

Tabel 1 Pengujian Fungsional

ID	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TC. 01	Login dengan email & password valid.	Masuk ke <i>dashboard</i> sesuai <i>role</i> .	Berhasil masuk ke <i>dashboard</i> .	Lolos
TC. 02	Login dengan kredensial tidak valid.	Muncul pesan "Kredensial tidak valid".	Pesan <i>error</i> tampil di layar.	Lolos
TC. 03	Mengklik tombol "Logout" di panel.	Sesi berakhir & kembali ke halaman login.	Sesi terhapus & dialihkan ke login.	Lolos
TC. 04	Unggah proyek format .zip valid.	File diterima & disimpan ke direktori.	File tersimpan tanpa <i>error</i> .	Lolos
TC. 05	Unggah format tidak valid (.rar/.tar).	Sistem menolak & memunculkan <i>error</i> .	Pesan "Format tidak didukung" tampil.	Lolos
TC. 06	Unggah file melebihi batas (1GB).	Sistem menghentikan proses (<i>Timeout</i>).	Proses berhenti dengan <i>error timeout</i> .	Lolos
TC. 07	Unggah file database .sql valid.	File SQL tersimpan di repositori.	SQL tersimpan untuk proses <i>import</i> .	Lolos
TC. 08	Deteksi <i>framework</i> (Laravel/Flask/PHP).	Mesin CI mengenali jenis <i>framework</i> .	<i>Framework</i> terdeteksi otomatis.	Lolos
TC. 09	Unggah proyek tanpa <i>framework</i> (Native).	Sistem otomatis menggunakan <i>fallback</i> .	Terdeteksi sebagai PHP Native/HTML.	Lolos
TC. 10	Menekan tombol "Inspect" pada antrean.	Tampil daftar file di dalam ZIP.	Pop-up daftar file & ukuran tampil.	Lolos
TC. 11	Menyetujui proyek	Otomatisasi <i>deploy</i> &	Proyek <i>Active</i> &	Lolos

	dengan database.	pembuatan DB.	DB terbuat.	
TC. 12	Menyetujui proyek statis (tanpa DB).	Ekstraksi file & konfigurasi Nginx.	Web statis berhasil di- <i>deploy</i> .	Lolos
TC. 13	Menolak proyek (<i>Reject</i>) oleh Admin.	Status <i>Rejected</i> & notifikasi terkirim.	Status berubah & <i>log</i> tampil.	Lolos
TC. 14	Mengakses subdomain proyek <i>Active</i> .	Web mahasiswa tampil di <i>browser</i> .	Web dapat diakses via subdomain.	Lolos
TC. 15	Menonaktifkan proyek (<i>Suspend</i>).	Akses Nginx diputus & status berubah.	Web tidak dapat diakses (Error 403).	Lolos
TC. 16	Mengaktifkan kembali proyek (<i>Resume</i>).	Konfigurasi pulih & web kembali <i>Live</i> .	Web dapat diakses normal kembali.	Lolos
TC. 17	Menghapus proyek secara permanen.	File & database terhapus dari server.	Seluruh data fisik & DB terhapus.	Lolos
TC. 18	Perintah aman di Web Terminal (ls).	Tampil <i>output</i> direktori di layar.	Daftar file tampil di terminal web.	Lolos
TC. 19	Perintah bahaya di Terminal (rm -rf).	Sistem memblokir eksekusi perintah.	Pesan "Akses Ditolak" tampil.	Lolos
TC. 20	Mengunduh file cadangan (<i>backup</i>).	Browser mengunduh arsip .zip proyek.	File <i>backup</i> berhasil diunduh.	Lolos
TC. 21	Membuka log instalasi proyek.	Tampil tahapan instalasi <i>real-time</i> .	Log proses (Extract/DB) tampil.	Lolos
TC. 22	Membuka halaman <i>Activity Logs</i> .	Riwayat tindakan admin terekam.	Daftar aktivitas & waktu tampil.	Lolos
TC. 23	Menambah akun mahasiswa baru.	Akun tersimpan & bisa digunakan login.	User baru berhasil ditambahkan.	Lolos
TC. 24	Notifikasi kredensial database.	Pesan kredensial tampil di panel user.	User menerima info akses DB.	Lolos

Berdasarkan hasil pengujian fungsional yang dieksekusi terhadap 24 skenario uji menunjukkan tingkat keberhasilan 100%. Sistem tidak hanya mampu menjalankan fungsionalitas utamanya seperti otomatisasi *deployment* dan konfigurasi server, tetapi juga terbukti tangguh dalam melakukan *error handling* terhadap input yang tidak valid (seperti pencegahan format file yang salah dan pembatasan ukuran file). Hal ini

menunjukkan bahwa sistem telah memenuhi spesifikasi kebutuhan fungsional secara keseluruhan.

D. Pengujian Integrasi

Pengujian integrasi dilakukan untuk memastikan seluruh komponen sistem terhubung dan bekerja secara terpadu menggunakan metode *black-box testing*, meliputi integrasi *frontend backend*, CI/CD, dan web server.

Tabel 2 Pengujian Integrasi

ID	Skenario Pengujian	Hasil yang Diharapkan	Hasil Aktual	Status
TI.01	Transmisi <i>payload</i> Frontend ke Backend.	Backend menerima & validasi file zip.	File diterima utuh oleh controller.	Lolos
TI.02	Pemicuan <i>Bash Script</i> oleh Backend.	Script CI/CD tereksekusi otomatis.	Script jalan tanpa kendala izin.	Lolos
TI.03	Integrasi Script dengan Nginx Config.	Virtual Host terbuat & Nginx reload.	Subdomain aktif secara otomatis.	Lolos
TI.04	Integrasi Script dengan MySQL Server.	Pembuatan DB & user database baru.	Database proyek berhasil di-generate.	Lolos
TI.05	Isolasi antar proyek di Nginx.	Routing subdomain tidak bertabrakan.	Setiap web tampil di URL berbeda.	Lolos
TI.06	Umpan balik error Deployment ke UI.	Log error server tampil di panel.	Pesan error muncul di layar user.	Lolos

Berdasarkan pengujian pada 6 skenario integrasi mencapai tingkat keberhasilan 100%. Komunikasi data antar-komponen mulai dari antarmuka pengguna (*frontend*), pemrosesan logika di *backend*, eksekusi *script bash* CI/CD, hingga pembuatan layanan basis data (MySQL) dan konfigurasi *routing web server* (Nginx) dapat beroperasi secara sinkron tanpa kendala. Hal ini membuktikan bahwa arsitektur sistem telah terintegrasi dengan kokoh.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian, sistem *Automated Final Project Deployment System* berhasil mengotomatisasi proses *deployment* proyek tugas akhir, mulai dari unggah, validasi melalui *Continuous Integration* (CI), hingga *deployment* dengan *Continuous Deployment* (CD) pada server. Sistem mampu mendeteksi jenis *framework* seperti Laravel, PHP Native, dan Flask, serta menyesuaikan proses *deployment* secara otomatis. Hasil

pengujian menggunakan *black-box testing* menunjukkan bahwa seluruh fitur utama, seperti upload project, CI/CD, manajemen project, dan monitoring, berjalan dengan baik. Selain itu, sistem menyediakan antarmuka yang memudahkan mahasiswa dan admin dalam mengelola *deployment* secara terpusat tanpa konfigurasi manual, sehingga proses publikasi aplikasi menjadi lebih efisien dan terstruktur.

B. Saran

Berdasarkan hasil penelitian, pengembangan selanjutnya disarankan untuk menambah dukungan berbagai *framework* dan teknologi agar sistem lebih fleksibel. Selain itu, sistem dapat dikembangkan untuk menangani aplikasi non-web seperti mobile dan game, serta diarahkan menjadi platform distribusi aplikasi yang memungkinkan karya mahasiswa diakses dan dimanfaatkan secara luas, sehingga tidak hanya berfungsi sebagai media *deployment* tetapi juga sebagai sarana publikasi.

REFERENSI

- [1] D. S. Wayan, P. I Nyoman, and P. W. Buana, "Integrasi Infrastructure as Code dengan Continuous Integration/Continuous Deployment di Google Cloud Platform," *JURNAL FASILKOM*, vol. 14, no. 2, pp. 346–354, Aug. 2024, doi: 10.37859/jf.v14i2.7236.
- [2] T. Pricillia and Zulfachmi, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *Jurnal Bangkit Indonesia*, vol. 10, no. 1, pp. 6–12, Mar. 2021, doi: 10.52771/bangkitindonesia.v10i1.153.
- [3] N. Ruest, S. Fritz, and I. Milligan, "Creating order from the mess: web archive derivative datasets and notebooks," *Archives and Records*, vol. 43, no. 3, pp. 316–331, 2022, doi: 10.1080/23257962.2022.2100336.
- [4] I. C. Donca, O. P. Stan, M. Misaros, D. Gota, and L. Miclea, "Method for Continuous Integration and Deployment Using a Pipeline Generator for Agile Software Projects," *Sensors*, vol. 22, no. 12, Jun. 2022, doi: 10.3390/s22124637.
- [5] D. Abdelfattah, H. A. Hassan, and F. A. Omara, "A novel role-mapping algorithm for enhancing highly collaborative access control system," *Distrib. Parallel Databases*, vol. 40, no. 2–3, pp. 521–558, Sep. 2022, doi: 10.1007/s10619-022-07407-9.
- [6] J. Luo, H. Wang, Y. Li, and Y. Lin, "Intrusion Detection System Based on Genetic Attribute Reduction Algorithm Based on Rough Set and Neural Network," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/5031236.
- [7] D. Kartika and Y. Mardiana, "Dns Server And Web Server Simulation With Debian Operating System On Local Area Network," 2023.
- [8] D. Deprizon, R. Fithri, W. Wismanto, B. Baidarus, and R. Refika, "Sistem Perencanaan Manajemen Pendidikan di Madrasah Ibtidaiyah Negeri 2 (MIN 2) Pekanbaru," *Mitra PGMI: Jurnal Kependidikan MI*, vol. 9, no. 1, pp. 1–15, Jan. 2023, doi: 10.46963/mpgmi.v9i1.800.
- [9] A. K. Nasich, S. A. Wicaksono, and M. C. Saputra, "Implementasi Role Based Access Control (RBAC) dalam Sistem Informasi Manajemen Pelanggan dan Pembayaran Air Berbasis Web (studi pada PT Tirta Wangi Sejahtera)," 2025. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [10] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, and W. Ramadhan, "Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 11, no. 1, p. 48, Jun. 2021, doi: 10.36448/expert.v11i1.2012.
- [11] N. S. Aji and A. L. Dwi, "IMPLEMENTASI CONTINUOUS INTEGRATION/CONTINUOUS DELIVERY (CI/CD) PADA PERFORMANCE TESTING DEVOPS."
- [12] U. Nugraha, "Design of information systems for population data collection based on client-server at Bagolo village," in *AIP Conference Proceedings*, American Institute of Physics Inc., Jun. 2017, doi: 10.1063/1.4985526.
- [13] M. Syauqi Amiq Amrullah and G. Indah Marthasari, "Otomatisasi Proses Deployment dengan Metode CI/CD Menggunakan Jenkins dan Docker Pada Web Service i-Lab," *REPOSITOR*, vol. 6, no. 4, pp. 313–322, 2024.
- [14] Omoniyi Babatunde Johnson, Jeremiah Olamijuwon, Zein Samira, Olajide Soji Osundare, and Harrison Oke Ekpobimi, "Developing advanced CI/CD pipeline models for Java and Python applications: A blueprint for accelerated release cycles," *Computer Science & IT Research Journal*, vol. 5, no. 12, pp. 2645–2663, Dec. 2024, doi: 10.51594/csitrj.v5i12.1758.
- [15] M. Nang Al Kodri, J. Ratu Penghulu No, K. Sari, and D. Redaksi, "INFORMATIKA DAN TEKNOLOGI (INTECH) The Design Of Web-Based Thesis Management Data Information System INFORMASI ARTIKEL ABSTRACT," *JURNAL INTECH*, vol. 5, no. 2, pp. 109–113, 2024.
- [16] M Sahyudi and Erliyan Redy Susanto, "Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning," *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 5, no. 1, pp. 105–116, Apr. 2025, doi: 10.54259/satesi.v5i1.3997.
- [17] K. Eliyen, R. Zulmy Alhamri, T. A. Cinderatama, F. Sofian Effendi, I. S. Asti, and F. Sukya, "Penerapan Pendaftaran Online untuk Program Edukasi Budidaya Ikan Lele pada Wisata Kampung Lele Kediri." [Online]. Available: <https://registration.kampunglelekediri.com>.
- [18] L. Santoso and J. Amanullah, "PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (RAD)," vol. 15, no. 2, pp. 250–259, 2022, [Online]. Available: <http://journal.stekom.ac.id/index.php/elkom> [page250]
- [19] J. Li, W. Xiao, and C. Zhang, "Data security crisis in universities: Identification of key factors affecting data breach incidents," *Humanit. Soc. Sci. Commun.*, vol. 10, no. 1, Dec. 2023, doi: 10.1057/s41599-023-01757-0.
- [20] K. H. Kumar, K. Subrahmanya, and S. Baliga, "An Approach to basic GUI-enabled CI/CD pipeline with Static Analysis tool."
- [21] G. Hyun, J. Oak, D. Kim, and K. Kim, "The Impact of an Automation System Built with Jenkins on the Efficiency of Container-Based System Deployment," *Sensors*, vol. 24, no. 18, Sep. 2024, doi: 10.3390/s24186002.
- [22] G. Putra Perdana and A. Prihanto, "Analisis Perbandingan Performa Web Server Apache, Nginx, Dan Litespeed Studi Kasus Implementasi Di VPS," *Journal of Informatics and Computer Science*, vol. 06, 2025.
- [23] Divya Kodi, "Efficient CI/CD Strategies: Integrating Git with automated testing and deployment," *World Journal of Advanced Research and Reviews*, vol. 20, no. 2, pp. 1517–1530, Nov. 2023, doi: 10.30574/wjarr.2023.20.2.2363.
- [24] C. C. Siburian, A. Andriani, and C. B. Dewa, "APLIKASI MYPRESENT UNTUK PENGELOLAAN DATA PRESENSI KARYAWAN DENGAN METODE RAD," 2024.
- [25] B. Quinteros et al., "Optimization of Document Management in Public Universities: Strategies and Technologies to Improve Administrative Processes," 2024. [Online]. Available: www.nano-ntp.com
- [26] I. Maulana, R. Umar, and A. Yudhana, "Implementasi Deployment Layanan Website Menggunakan Kubernetes Dengan Ci/Cd Jenkins," *Agustus*, vol. 23, pp. 290–296, 2024, [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jis/index>
- [27] J. Nurhayati and A. Info, "Implementation of Continuous Integration and Continuous Deployment (CI/CD) to Speed up the Automation Process of Software Delivery In the Production Process Using Node.Js, Docker, and React," *Informatika dan Sains*, vol. 14, 2024, doi: 10.54209/infosains.v14i02.
- [28] A. Farid and I. Gita Anugrah, "Implementasi CI/CD Pipeline Pada Framework Androbase Menggunakan Jenkins (Studi Kasus: PT. Andromedia)," *Jurnal Nasional Komputasi dan Teknologi Informasi*, vol. 4, no. 6, 2021.
- [29] A. Alexander and W. Wella, "CONTINUOUS INTEGRATION PIPELINE WITH JENKINS," *Jurnal Pilar Nusa Mandiri*, vol. 21, no. 1, pp. 1–8, Mar. 2025, doi: 10.33480/pilar.v21i1.6062.
- [30] Y. Fransiska Aladina, A. Bhawiyuga, R. Andria Siregar, P. Hari Trisnawan, U. Brawijaya, and P. Korespondensi, "PENERAPAN MEKANISME CONTINUOUS DEPLOYMENT DALAM PENGEMBANGAN DAN PEMBARUAN PERANGKAT LUNAK SISTEM BENAM BERBASIS INTERNET OF THINGS," vol. 9, no. 3, pp. 647–654, 2022, doi: 10.25126/jtiik.202295750.

- [31] E. P. Wijaya, S. Kosasi, and David, "Implementing Continuous Integration and Deployment Strategy: Cloversy.id RESTful API Development," *Jurnal RESTI*, vol. 8, no. 3, pp. 368–376, Jun. 2024, doi: 10.29207/resti.v8i3.5527.
- [32] A. D. Setyoko and A. Zahra, "Perbandingan Efisiensi Proses CI/CD Multi-Lingkungan melalui Implementasi Paralel dan Berurutan," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 911–925, May 2024, doi: 10.57152/malcom.v4i3.1334.
- [33] N. Fadilah, R. Hartono, D. Syahrul Anwar Teknik Informatika, U. J. Perjuangan Tasikmalaya Peta No, and J. Barat, "PENGEMBANGAN PROTOTYPE ARSITEKTUR CI/CD PADA SISTEM INFORMASI MANAJEMEN TERINTEGRASI UNIVERSITAS PERJUANGAN MENGGUNAKAN JENKINS," 2025.