

**ANALISIS KEANDALAN CI/CD *PIPELINE* PADA ARSITEKTUR  
MONOLITIK DAN *MICROSERVICES* DALAM MENANGANI  
KEGAGALAN *DEPLOYMENT***



**SKRIPSI**

Diajukan untuk memenuhi Sebagian dari persyaratan dalam memperoleh gelar  
Sarjana Teknik pada Program Studi Sistem Telekomunikasi

Oleh:

Tita Rismawati

2104911

**PROGRAM STUDI SISTEM TELEKOMUNIKASI**

**KAMPUS UPI DI PURWAKARTA**

**UNIVERSITAS PENDIDIKAN INDONESIA**

**2025**

**LEMBAR HAK CIPTA**

**ANALISIS KEANDALAN CI/CD *PIPELINE* PADA ARSITEKTUR  
MONOLITIK DAN *MICROSERVICES* DALAM MENANGANI  
KEGAGALAN *DEPLOYMENT***

Oleh

**Tita Rismawati**

Sebuah skripsi yang diajukan untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Teknik pada Program Studi Sistem Telekomunikasi

©Tita Rismawati

Universitas Pendidikan Indonesia

Juni 2025

Hak Cipta dilindungi oleh undang-undang

Skripsi ini tidak boleh diperbanyak seluruhnya atau Sebagian,  
Dengan dicetak ulang, difoto kopi, atau cara lainnya tanpa ijin dari penulis.

**LEMBAR PENGESAHAN**

Skripsi dengan Judul

**ANALISIS KEANDALAN CI/CD *PIPELINE* PADA ARSITEKTUR  
MONOLITIK DAN *MICROSERVICES* DALAM MENANGANI  
KEGAGALAN *DEPLOYMENT***

Oleh

Tita Rismawati

2104911

**Disetujui dan Disahkan Oleh Pembimbing:**

Pembimbing I,  


**Galura Muhammad Suranegara, S.Pd., M.T.**  
NIP. 920190219920111101

Pembimbing II,



**Endah Setyowati, S.T., M.T.**  
NIP. 920190219920908201

Mengetahui,

**Ketua Program Studi**

**Sistem Telekomunikasi**



**Galura Muhammad Suranegara, S.Pd., M.T.**  
NIP. 92019021992011110

## PERNYATAAN BEBAS PLAGIARISME

Yang bertandatangan dibawah ini:

Nama : Tita Rismawati  
NIM : 2104911  
Program Studi : Sistem Telekomunikasi  
Judul Karya : Analisis Keandalan *CI/CD Pipeline* pada Arsitektur Monolitik dan *Microservices* dalam Menangani Kegagalan *Deployment*

Dengan ini menyatakan bahwa karya tulis ini merupakan hasil kerja saya sendiri. Saya menjamin bahwa seluruh isi karya ini, baik sebagian maupun keseluruhan, bukan merupakan plagiarisme dari karya orang lain, kecuali pada bagian yang telah dinyatakan dan disebutkan sumbernya dengan jelas.

Jika di kemudian hari ditemukan pelanggaran terhadap etika akademik atau unsur plagiarisme, saya bersedia menerima sanksi sesuai peraturan yang berlaku di Universitas Pendidikan Indonesia.

Bandung, Juni 2025

Yang membuat pernyataan

Tita Rismawati

## ABSTRAK

Penelitian ini bertujuan untuk menganalisis keandalan *Continuous Integration and Continuous Deployment (CI/CD) pipeline* pada arsitektur monolitik dan *microservices* dengan studi kasus platform Moodle. Analisis difokuskan pada *Mean Time to Recovery (MTTR)* yang mencakup *downtime* dan waktu *rollback*, serta tingkat keberhasilan *deployment* dalam menangani kegagalan *deployment* akibat kesalahan dependensi. Pengujian dilakukan sebanyak 20 kali pada masing-masing arsitektur dengan dua skenario, yaitu tanpa kesalahan dependensi dan dengan kesalahan dependensi. Hasil pengujian menunjukkan bahwa pada skenario tanpa kesalahan dependensi, rata-rata MTTR pada Moodle dengan arsitektur monolitik mencapai 19 detik, sedangkan pada *microservices* hanya berkisar antara 4,1 hingga 4,3 detik, dengan tingkat keberhasilan *deployment* sebesar 100% pada kedua arsitektur. Namun, pada skenario dengan kesalahan dependensi, MTTR pada arsitektur monolitik meningkat drastis menjadi 117,3 detik, termasuk waktu *rollback* sebesar 26,85 detik. Sementara itu, pada arsitektur *microservices*, MTTR meningkat menjadi 26,35 hingga 28,8 detik, termasuk waktu *rollback* antara 5,4 hingga 6,2 detik. Pada kondisi ini, tingkat keberhasilan *deployment* menurun menjadi 0% pada kedua arsitektur. Temuan ini mengindikasikan bahwa Moodle dengan arsitektur *microservices* lebih tangguh terhadap kesalahan dependensi dan lebih efektif dalam meminimalkan risiko kegagalan *deployment*, serta mampu mengurangi MTTR secara signifikan dibandingkan arsitektur monolitik. Oleh karena itu, penerapan arsitektur *microservices* sangat direkomendasikan untuk meningkatkan keandalan proses *deployment* dalam pengembangan perangkat lunak khususnya dalam pengembangan *Learning Management System (LMS)* seperti Moodle.

**Kata Kunci:** Moodle, *CI/CD Pipeline*, Monolitik, *Microservices*, *Downtime*, *Rollback*, Kesalahan dependensi

## **ABSTRACT**

*This research aims to analyze the reliability of the Continuous Integration and Continuous Deployment (CI/CD) pipeline on monolithic and microservices architectures with a case study of the Moodle platform. The analysis focused on Mean Time to Recovery (MTTR) which includes downtime and rollback time, as well as the deployment success rate in handling deployment failures due to dependency errors. Tests were conducted 20 times on each architecture with two scenarios, namely without dependency errors and with dependency errors. The test results showed that in the scenario without dependency errors, the average MTTR on Moodle with monolithic architecture reached 19 seconds, while that on microservices only ranged from 4.1 to 4.3 seconds, with a deployment success rate of 100% on both architectures. However, in the scenario with dependency errors, the MTTR on the monolithic architecture increased dramatically to 117.3 seconds, including a rollback time of 26.85 seconds. Meanwhile, in the microservices architecture, the MTTR increased to 26.35 to 28.8 seconds, including a rollback time between 5.4 to 6.2 seconds. In this condition, the deployment success rate decreases to 0% in both architectures. These findings indicate that Moodle with microservices architecture is more resilient to dependency errors and more effective in minimizing the risk of deployment failures, as well as being able to significantly reduce MTTR compared to monolithic architecture. Therefore, the application of microservices architecture is highly recommended to improve the reliability of the deployment process in software development, especially in the development of Learning Management System (LMS) such as Moodle.*

**Keywords:** Moodle, CI/CD Pipeline, Monolithic, Microservices, Downtime, Rollback, Dependency errors

## DAFTAR ISI

LEMBAR HAK CIPTA .....	ii
LEMBAR PENGESAHAN .....	iii
PERNYATAAN BEBAS PLAGIARISME .....	iv
HALAMAN UCAPAN TERIMAKASIH.....	v
ABSTRAK.....	vii
<i>ABSTRACT</i> .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian.....	3
1.4 Manfaat Penelitian.....	3
1.4.1 Secara Teoritis .....	3
1.4.2 Secara Praktis .....	4
1.5 Ruang Lingkup Penelitian.....	5
BAB II TINJAUAN PUSTAKA .....	6
2.1 <i>Learning Management System (LMS) Moodle</i> .....	6
2.2 <i>Continuous Integration and Continuous Deployment (CI/CD) Pipeline</i> ....	7
2.2.1 <i>Continuous Integration (CI)</i> .....	7
2.2.2 <i>Continuous Deployment (CD)</i> .....	8
2.3 <i>Arsitektur Perangkat Lunak</i> .....	8
2.3.1 <i>Arsitektur Monolitik</i> .....	9
2.3.2 <i>Arsitektur Microservices</i> .....	9
2.4 <i>Teknologi dalam CI/CD Pipeline</i> .....	10
2.4.1 <i>Jenkins</i> .....	10
2.4.2 <i>Docker</i> .....	11

2.4.3	Gitlab.....	11
2.4.4	Node.js.....	12
2.4.5	Swagger.....	12
2.5	Kegagalan <i>Deployment</i> .....	12
2.6	Studi Literatur dan Penelitian Terdahulu.....	14
BAB III METODE PENELITIAN.....		17
3.1	Alur Penelitian.....	17
3.1.1	Studi Literatur .....	17
3.1.2	Pengembangan Moodle Monolitik.....	18
3.1.3	Pengembangan <i>Web Service Microservices</i> Moodle.....	19
3.1.4	Pembuatan <i>Pipeline</i> .....	20
3.1.5	Pengujian <i>Pipeline</i> .....	23
3.1.6	Analisis Data .....	27
BAB IV HASIL DAN PEMBAHASAN.....		29
4.1	Hasil Penelitian .....	29
4.1.1	Implementasi <i>CI/CD Pipeline</i> pada Arsitektur Monolitik dan <i>Microservices</i> .....	29
4.1.2	Perbandingan <i>CI/CD Pipeline</i> pada Arsitektur Monolitik dan <i>Microservices</i> Berdasarkan Parameter <i>Downtime</i> , Waktu <i>Rollback</i> , dan Keberhasilan <i>Deployment</i> .....	43
4.1.3	Mengidentifikasi Korelasi antara Kesalahan Dependensi dan Kegagalan <i>Deployment</i> pada Arsitektur Monolitik dan <i>Microservices</i> .....	46
BAB V SIMPULAN DAN SARAN .....		48
5.1	Simpulan .....	48
5.2	Saran.....	48
DAFTAR PUSTAKA .....		50
LAMPIRAN.....		53
DAFTAR RIWAYAT PENULIS .....		61



## DAFTAR GAMBAR

Gambar 2.1 Arsitektur teknis Moodle.....	6
Gambar 2.2 Arsitektur perangkat lunak .....	8
Gambar 2.3 Alur kerja Jenkins.....	10
Gambar 2.4 Arsitektur Docker .....	11
Gambar 3.1 Alur Penelitian.....	17
Gambar 3.2 Pengembangan Moodle Monolitik.....	18
Gambar 3.3 Pengembangan <i>Web Service Microservices</i> Moodle.....	19
Gambar 3.4 Tahapan <i>pipeline</i> Moodle monolitik.....	21
Gambar 3.5 Tahapan <i>pipeline</i> Moodle <i>microservices</i> .....	22
Gambar 3.6 Alur Pengujian.....	24
Gambar 4.1 Moodle berjalan di <i>production</i> dengan <i>container</i> docker .....	29
Gambar 4.2 Halaman dashboard <i>default</i> Moodle .....	30
Gambar 4.3 Menghentikan <i>container</i> Moodle yang berjalan di <i>production</i> .....	30
Gambar 4.4 Melakukan <i>deploy</i> kode dengan perubahan terbaru.....	31
Gambar 4.5 Cek curl Moodle yang berjalan di <i>production</i> .....	31
Gambar 4.6 <i>Pipeline</i> melakukan cek <i>logs container</i> .....	31
Gambar 4.7 Terdeteksi <i>error</i> di Moodle .....	32
Gambar 4.8 <i>Pipeline</i> menjalankan <i>rollback</i> di monolitik .....	32
Gambar 4.9 <i>Rollback</i> berhasil dan memberikan hasil <i>downtime</i> dan waktu <i>rollback</i> .....	32
Gambar 4.10 <i>User service</i> berjalan pada port 3001 .....	33
Gambar 4.11 Dokumentasi API <i>user service</i> untuk menambahkan <i>user</i> .....	33
Gambar 4.12 Menambahkan <i>user</i> berhasil.....	34
Gambar 4.13 <i>Course service</i> berjalan pada port 3002 .....	34
Gambar 4.14 Dokumentasi API <i>course service</i> untuk menambahkan kategori <i>course</i> .....	35
Gambar 4.15 Berhasil menambahkan kategori baru .....	35
Gambar 4.16 Dokumentasi API <i>course service</i> menambahkan <i>course</i> baru .....	36
Gambar 4. 17 Berhasil menambahkan <i>course baru</i> .....	36
Gambar 4.18 <i>Enrol service</i> berjalan pada port 3003.....	36

Gambar 4.19 Dokumentasi API <i>enrol service</i> melakukan <i>user enrol</i> ke dalam <i>course</i> .....	37
Gambar 4.20 Berhasil melakukan <i>user enrol</i> .....	37
Gambar 4.21 <i>Pipeline</i> melakukan cek port .....	38
Gambar 4.22 <i>Pipeline</i> melakukan <i>deploy</i> dan paralel <i>stage</i> dengan curl di <i>production</i> .....	38
Gambar 4.23 <i>Pipeline</i> melakukan <i>rollback</i> ke aplikasi stabil sebelumnya.....	39
Gambar 4.24 <i>Rollback</i> berhasil dan aplikasi bisa di akses di <i>production</i> .....	39
Gambar 4.25 <i>Pipeline</i> berhasil mendeteksi versi <i>image</i> Moodle dan MariaDB yang valid pada arsitektur monolitik.....	40
Gambar 4.26 Proses <i>deployment</i> dilanjutkan setelah versi <i>image</i> divalidasi sebagai sesuai.....	40
Gambar 4.27 <i>Pipeline</i> menghentikan proses karena versi <i>image</i> tidak sesuai.....	41
Gambar 4.28 Eksekusi <i>server.js</i> berhasil dijalankan tanpa <i>error</i> pada salah satu <i>service</i> dalam arsitektur <i>microservices</i> . .....	41
Gambar 4.29 Proses <i>deployment</i> dilanjutkan setelah <i>service</i> berhasil dijalankan dengan kode yang valid.....	42
Gambar 4.30 <i>Pipeline</i> mendeteksi kesalahan pada kode <i>service</i> yang diuji.....	42
Gambar 4.31 <i>Pipeline</i> menghentikan proses eksekusi <i>stage</i> . .....	42
Gambar 4.32 Rata-rata <i>downtime deployment</i> Moodle.....	46

## DAFTAR TABEL

Tabel 4.1 Perhitungan rata-rata <i>downtime</i> dan waktu <i>rollback</i> .....	43
Tabel 4.2 Data hasil pengukuran <i>deployment</i> Moodle .....	43

## DAFTAR PUSTAKA

- Badampudi, D., Usman, M., & Chen, X. (2025). Large scale reuse of microservices using CI/CD and InnerSource practices—A case study. *Empirical Software Engineering*, 30(2), 41. <https://doi.org/10.1007/s10664-024-10595-w>
- Chippagiri, S., & Kassetty, N. (2025). BEYOND THE MONOLITH: COMPREHENSIVE STRATEGIES FOR ARCHITECTING, SCALING, AND SUSTAINING RESILIENT DISTRIBUTED SYSTEMS. *INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND INFORMATION TECHNOLOGY*, 8(1), 152–168. [https://doi.org/10.34218/IJRCAIT\\_08\\_01\\_016](https://doi.org/10.34218/IJRCAIT_08_01_016)
- Choudhury, P., Crowston, K., Dahlander, L., Minervini, M. S., & Raghuram, S. (2020). GitLab: Work where you want, when you want. *Journal of Organization Design*, 9(1), 23. <https://doi.org/10.1186/s41469-020-00087-8>
- Dahri, F., Hanafi, A. M. E., Handoko, D., & Wulan, N. (2022). Implementation of Microservices Architecture in Learning Management System E-Course Using Web Service Method. *Sinkron*, 7(1), 76–82. <https://doi.org/10.33395/sinkron.v7i1.11229>
- Dakić, P. (2024). Software compliance in various industries using CI/CD, dynamic microservices, and containers. *Open Computer Science*, 14(1), 20240013. <https://doi.org/10.1515/comp-2024-0013>
- David, A., Mihai, D., Mihailescu, M.-E., Carabas, M., & Tapus, N. (2022). Scalability through Distributed Deployment for Moodle Learning Management System. *Procedia Computer Science*, 214, 34–41. <https://doi.org/10.1016/j.procs.2022.11.145>
- Guggenheimer, S., & Koduri, A. (2023). AWS Prescriptive Guidance—The CI/CD litmus test: Is your pipeline fully CI/CD? *Amazon Web Services*. <https://docs.aws.amazon.com/prescriptive-guidance/latest/strategy-cicd-litmus/>

- Gurusamy, A., & Mohamed, I. A. (2020). The Evolution of Full Stack Development: Trends and Technologies Shaping the Future. *Journal of Knowledge Learning and Science Technology*, 100-108.
- Hassan, F. (2024). Software Architecture Between Monolithic and Microservices Approach. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.4753649>
- Islavath, N. (2021). Jenkins for Continuous Integration and Deployment in DevOps Engineer: A Deep Dive into Features and Best Practices. *International Journal of Science and Research (IJSR)*, 10(12), 1546–1550. <https://doi.org/10.21275/SR20216084821>
- Ismail, A. (2023). High Availability Moodle dengan Load Balancer pada Three-tier Architecture. *Jurnal Minfo Polgan*, 12(1), 297–302. <https://doi.org/10.33395/jmp.v12i1.12376>
- Jaswanth Alahari, Abhishek Tangudu, Chandrasekhara Mokkaapati, Om Goel, & Prof. (Dr.) Arpit Jain. (2024). Implementing Continuous Integration/Continuous Deployment (CI/CD) Pipelines for Large-Scale iOS Applications. *Darpan International Research Analysis*, 12(3), 522–534. <https://doi.org/10.36676/dira.v12.i3.104>
- Kolawole, I., & Fakokunde, A. (2025). Improving Software Development with Continuous Integration and Deployment for Agile DevOps in Engineering Practices. *International Journal of Computer Applications Technology and Research*, 25–39.
- Korrapati, R. (2020). Modernizing Software Architecture: Transitioning to Monolithic Designs for Enhanced. *SSRN*.
- Listiani, T. (2021). IMPROVING STUDENT LEARNING OUTCOMES IN STATISTICS FOR RESEARCH EDUCATION COURSES USING MOODLE PLATFORM DURING PANDEMIC. *AKSIOMA: Jurnal Program Studi Pendidikan Matematika*, 10(3), 1916. <https://doi.org/10.24127/ajpm.v10i3.4032>
- Nagaraju Thallapally. (2021). Implementing continuous integration and continuous deployment (CI/CD) pipelines. *International Journal of Science and*

*Research Archive*, 3(2), 248–253.  
<https://doi.org/10.30574/ijjsra.2021.3.2.0073>

- Rani, J. U., Kumar, J. P. S., & Raj, D. B. C. (2023). Adoption of Learning Management System Among Students in Higher Educational Institutions—A Case on Moodle LMS. In S. Jayasingh, K. Boobalan, & T. Thiagarajan (Eds.), *Proceedings of the International Conference on Emerging Trends in Business & Management (ICETBM 2023)* (Vol. 242, pp. 22–34). Atlantis Press International BV. [https://doi.org/10.2991/978-94-6463-162-3\\_4](https://doi.org/10.2991/978-94-6463-162-3_4)
- Santos, J., Azevedo, L., Soares, E., Thiago, R., & Silva, V. (2020). Analysis of Tools for REST Contract Specification in Swagger/OpenAPI: *Proceedings of the 22nd International Conference on Enterprise Information Systems*, 201–208. <https://doi.org/10.5220/0009381202010208>
- Taras, S., & Sergiy, T. (2023). Modular Monolith as a Microservices Precursor. *Modern Engineering and Innovative Technologies*, 25–31.
- Wangdong Wu. (2024). Automated Model Management for Microservices: A CI/CD Approach. *Computer Fraud and Security*, 34–43. <https://doi.org/10.52710/cfs.87>