

# Implementasi CICD Dalam Pengembangan Sistem Absensi Online

Komariyuli Anwariyah<sup>1</sup>

<sup>1</sup> Politeknik Negeri Bali ; [yuliaanwariyah@gmail.com](mailto:yuliaanwariyah@gmail.com)

\* Korespondensi: yuliaanwariyah@gmail.com;

**Abstract:** System development using the Systems Development Life Cycle (SDLC) framework ensures that the resulting software is effective, efficient, and aligned with its objectives. This research optimizes the SDLC implementation process by integrating Continuous Integration and Continuous Deployment (CI/CD) pipelines to achieve faster, safer, and more consistent development cycles. The implementation covers the entire workflow from code development to automated server deployment. Experimental results demonstrate that the CI/CD integration significantly shortens operational time, achieving an average deployment duration of 42 seconds. Furthermore, the utilization of CI/CD effectively reduces manual intervention, thereby enhancing overall development performance and team productivity.

**Keywords:** Continuous Integration, Continuous Deployment, SDLC, GitLab Pipeline, Deployment Automation.

**Abstrak:** Pengembangan sistem menggunakan metode Systems Development Life Cycle (SDLC) bertujuan untuk memastikan bahwa sistem yang dihasilkan berkualitas, sesuai dengan tujuan, serta mampu berjalan secara efektif dan efisien. Penelitian ini mengusulkan optimalisasi proses implementasi dalam SDLC melalui penerapan Continuous Integration dan Continuous Deployment (CI/CD) untuk menciptakan alur kerja yang lebih cepat, aman, dan konsisten. Fokus implementasi dilakukan pada siklus pengembangan kode hingga tahapan deployment ke server secara otomatis. Hasil pengujian menunjukkan bahwa penerapan CI/CD berhasil memangkas waktu operasional secara signifikan, dengan rata-rata waktu yang dibutuhkan untuk proses deployment hanya sebesar 42 detik. Selain itu, pemanfaatan CI/CD terbukti mampu mengurangi intervensi manual dalam proses pengembangan, yang pada akhirnya meningkatkan produktivitas dan kinerja tim pengembang secara keseluruhan.



Copyright: © 2025 oleh para penulis.

Karya ini dilisensikan di bawah Creative Commons Attribution-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-sa/4.0/>).

**Kata kunci:** Continuous Integration, Continuous Deployment, SDLC, GitLab Pipeline, Otomatisasi Deployment.

## 1. Pendahuluan

Saat ini penggunaan sistem informasi sudah merambah diberbagai sektor kehidupan baik dalam kehidupan sehari-hari maupun dalam pengelolaan manajemen dan administrasi. Salah satu implementasi penggunaan sistem informasi ialah pada proses absensi online pegawai yang saat ini sudah sangat umum digunakan. Tingginya tingkat penggunaan sistem ini tentunya harus dibarengi dengan metode pengembangan yang terintegrasi dan berkesinambungan, salah satu metode pengembangan sistem yang cukup efektif untuk diimplementasikan adalah *System Development Life Cycle (SDLC)*. SDLC merupakan proses pengembangan sistem yang meliputi tahapan perencanaan, analisis, desain, implementasi dan testing [1].

Pengembangan sistem menggunakan metode SDLC dapat membantu memastikan bahwa sistem yang dihasilkan sesuai dengan tujuan yang diharapkan, berkualitas dan

dapat mampu berjalan secara efektif dan efisien [2]. Proses implementasi pengembangan sistem dengan metode ini dapat dioptimalkan dengan pengimplementasian code Integration Code Deployment (CI CD) dalam beberapa fase SDLC secara lebih cepat, aman dan konsisten. CI/CD merupakan metode pengembangan perangkat lunak yang menekankan pada frekuensi perubahan code secara berkala, pengujian otomatis dan proses *deploy* sistem yang berkelanjutan guna memastikan seluruh pembaharuan sistem dapat dikirimkan secara lebih cepat, efisien dan aman [3]. Penerapan CI/CD memberikan beberapa keuntungan diantaranya memudahkan kolaborasi dan interaksi antar tim, memperpendek siklus pengembangan, dan proses deployment yang lebih sederhana, peningkatan kualitas dan mengurangi resiko gangguan atau msalah yang significant [4][5]. Meski memiliki berbagai keuntungan penerepan CI CD bukan tanpa tantangan, terdapat beberapa tantangan dalam penerapan ini yaitu perbedaan budaya, kompleksitas, integrasi tools, dan keamanan [5].

Merujuk pada kajian diatas pada penelitian ini akan dilakukan pengujian terkait bagaimana implementasi CI/CD dalam pengembangan sistem sederhana. Hal ini dilakukan untuk mengetahui kendala yang dihadapi dan efektifitas penerapannya. Pengujian dilakukan pada pengembangan sistem absensi online yang nantinya akan digunakan sebagai sistem absensi online pada Politeknik Negeri Bali Kampus Lombok Barat.

## 2. Bahan dan Metode

### Bahan

Penelitian ini membutuhkan beberapa perangkat pendukung baik perangkat keras maupun perangkat lunak. Secara detail kebutuhan perangkat keras dan lunak pada proses pengembangan sistem ini adalah sebagai berikut :

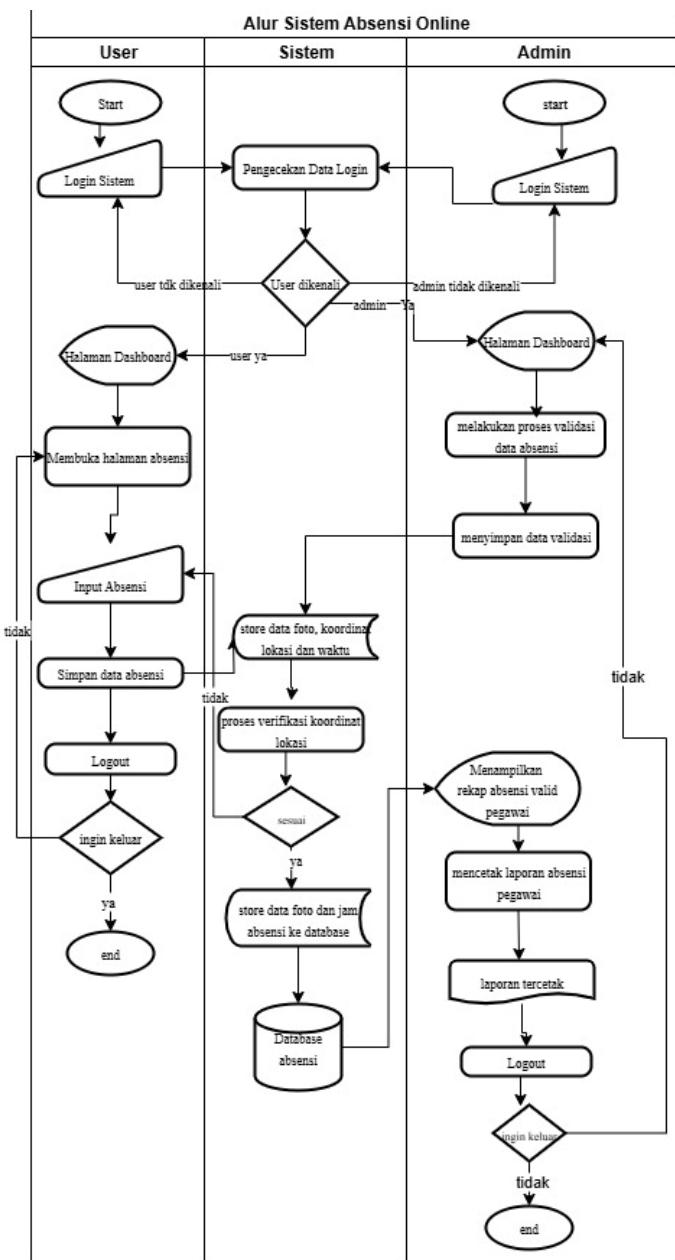
- Kebutuhan Perangkat keras
  - Perangkat lokal developer processor intel core i7 gen 8, Ram 16 GB, SSD 512 GB
  - Perangkat Server
- Kebutuhan perangkat lunak
  - Gitlab repository
  - Gitlab Runner
  - Framework Laravel versi 11.9
  - PHP 8.2
  - Ubuntu Server 22.04
  - MySql 8
  - Git

### Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah system Development Life Cycle (SDLC) dengan mengimplementasikan aliran CI/CD pada beberapa tahapannya. Hal ini dilakukan untuk mengoptimalkan proses pengembangan sistem yang dilakukan. Pada metode SDLC terdapat beberapa tahapan diantaranya

- Analisa
- Tahapan analisa dilakukan untuk mengetahui kebutuhan sistem yang akan dibangun. Pada tahapan ini dilakukan pengumpulan data dari berbagai sumber termasuk dari Politeknik Negeri Bali Kampus Lombok Barat selaku pengguna akhir dari sistem yang akan dibangun. Proses pengumpulan data diakukan dengan observasi dan wawancara langsung kepada pengguna. Berdasarkan data yang didapatkan dilakukan analisis untuk menentukan kebutuhan fungsional dari sistem yang akan dibangun.
- Desain Sistem
- Berdaasarkan analisis yang telah dilakukan dibuat analisis alur sistem seperti yang tampak pada gambar 1 berikut ini

93



Gambar 1. Alur sistem absensi online

94

95

96

97

98

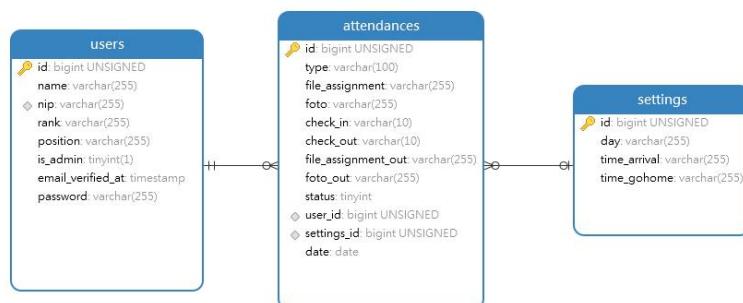
99

100

101

102

Pada gambar 1 dapat dilihat alur sistem absensi yang akan di bangun pada penelitian ini. Alur ini selanjutnya akan diimplementasikan kedalam code program untuk selanjutnya dilakukan pengujian dan deployment. Selain alur sistem pada tahapan desain dilakukan pembuatan desain database yang nantinya akan digunakan untuk menyimpan data yang ada pada sistem. Adapun rancangan ERD dari sistem ini seperti tampak pada gambar 2 dibawah ini



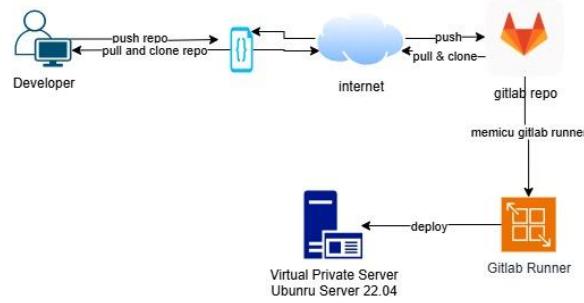
103

Gambar 2 ERD database sistem

Pada diagram ERD (gambar 2) ditunjukkan bahwa database siste ini cukup sederhana hanya terdiri dari 3 buah entity yaitu *user*, *attendances* dan *settings*

- Pengembangan (Coding)

Pada proses pengembangan code dilakukan dengan menggunakan framework laravel dengan integrasi gitlab repository dan server stagging. Pada proses pengembangan ini membutuhkan instalasi beberapa perangkat lunak pendukung untuk membantu interkoneksi antara perangkat lokal developer dengan gitlab repository. Pada proses pengembangan ini mulai diimplementasikan fase CI CD diadalah dalam prosesnya. Adapun code pada aliran CI CD seperti tampak pada gambar 3.



Gambar 3 diagram Alur aliran CI/CD

Pada alur pengembangan dapat dilihat bahwa developper dapat melakukan clone ataupun pull project dari repository unutk memastikan repository selalu dalam keadaan terbarukan. Selanjutnya developper dapat melakukan push perubahan code ke repository yang memicu eksekusi perintah yang ada pada `gitlab-ci.yml` untuk menjalankan setiap stage proses yang telah diberikan.

- Implementasi/Deploy

Tahapan ini dilakukan proses implementasi code yang telah dibuat pada server yang telah disiapkan. Pada server telah terinstal gitlab-runner dan dilakukan registrasi gitlab-runner dengan project pada repository gitlab yang akan di deploy. Gitlab runner nantinya akan menerima dan mengeksekusi job CI/CD dari pipeline gitlab. Job yang dijalankan gitlab-runner bergantung pada jumlah stage dan statement yang telah dikonfigurasi pada `gitlab-ci.yml`. pada penelitian ini hanya akan diimplementasikan stage sederhana mulai dari pengujian, pengecekan kesiapan server, serta proses deployment.

- Pengujian

Pada penelitian ini proses pengujian tidak diintegrasikan langsung kedalam job CI/CD, proses pengujian dilakukan diluar aliran CI/CD setelah proses deployment selesai dilakukan. Proses pengujian dilakukan untuk memastikan sistem yang telah dideploy melalui aliran CI/CD dapat diakses dan digunakan dengan baik.

### 3. Hasil

#### 3.1. Hasil konfigurasi

##### 3.1.1. Persiapan perangkat lokal

Pada tahapan implementasi aliran CI/CD dilakukan konfigurasi pada perangkat lokal developer untuk menghubungkan project yang ada pada perangkat lokal developer dengan gitlab repository.

```

Windows PowerShell x + -
PS E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi> git clone https://gitlab.com/yuliaanwariyah/polda-attendance-kampus-poltik.git
Cloning into 'polda-attendance-kampus-poltik'...
remote: Enumerating objects: 100% (4568/4568), done.
remote: Compressing objects: 100% (3215/3215), done.
remote: Total 4568 (delta 1232), reused 4561 (delta 1231), pack-reused 0 (from 0)
Receiving objects: 100% (4568/4568), 45.99 MiB | 3.49 MiB/s, done.
Resolving deltas: 100% (1232/1232), done.
PS E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi> ls

Directory: E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi

Mode                LastWriteTime         Length Name
----                -----        15.05          polda-attendance-kampus-poltik

PS E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi> cd .\polda-attendance-kampus-poltik\
PS E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi\polda-attendance-kampus-poltik> ls

Directory: E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi\polda-attendance-kampus-poltik

Mode                LastWriteTime         Length Name
----                -----        15.05          6289 README.md

PS E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi\polda-attendance-kampus-poltik> git checkout
Your branch is up to date with 'origin/main'.
PS E:\DataKampus\PNB\PSDKU LOBAR\Kinerja\Program\New_gitlab_absensi\polda-attendance-kampus-poltik> git checkout master
Updating files: 100% (5956/5956), done.

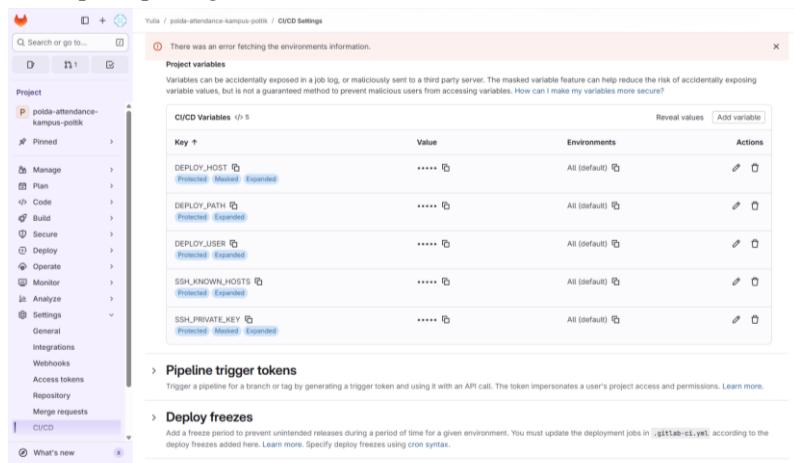
```

Gambar 4 konfigurasi lokal

Pada gambar 4 dapat dilihat pada lokal workplace dilakukan git clone untuk menduplikasi project yang ada pada gitlab repository, selanjutnya dilakukan checkout pada branch master untuk dapat masuk ke project brach master yang ada pada repository.

### 3.1.2. Konfigurasi gitlab repository

Pada gitlab repository dilakukan konfigurasi beberapa variable CI/CD diantaranya SSH\_Private\_Key, SSH\_Known\_Host, Deploy\_Host, Deploy\_User, dan Deploy\_Path. Proses konfigurasi ini dilakukan didalam project yang akan di dideploy ke server. Penambahan variable dilakukan melalui menu setting lalu masuk ke sub menu CI/CD dan selanjutnya masuk ke menu Variable dan tambahkan setiap variable yang telah ditetapkan seperti pada gambar 5



Gambar 5 konfigurasi variable gitlab CI/CD

Selanjutnya pada Runners dilakukan pembuatan runner project yang nantinya akan diregistrasi pada server yang akan digunakan.

### 3.1.3. Konfigurasi Private Virtual Server (VPS)

Terdapat beberapa konfigurasi yang harus dibuat pada virtual private server diantaranya konfigurasi dan instalasi php, mysql, composer dan web server. Konfigurasi awal ini dilakukan untuk memastikan bahwa server benar – benar telah siap. Selain itu pada server dilakukan instalasi gitlab runner seperti tampak pada gambar 6. Proses instalasi gitlab runner dilanjutkan dengan registrasi runner, hal ini dilakukan

172  
173  
174

untuk mendaftarkan runner yang ada pada project kedalam server dengan menggunakan token yang telah disediakan, hal ini dilakukan untuk memastikan keamanan dimana hanya akses code hanya dapat dilakukan oleh runner yang terdaftar.

175

176

177

178

179

```

root@prsdkulobar: ~
# Unable to locate package gitlab-runner
# curl -L https://packages.gitlab.com/gitlab/gitlab-runner/gpgkey
# curl -L https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh | sudo bash
% Total % Received % Xferd Average Speed Time Time Current
          0 0 0 0 0 0
Total 100 6885 100 6885 0 0 15871 0 --:--:--:--:--:-- 15900
Detected operating system as Ubuntu/jammy.
Checking for curl...
Detected curl...
Checking for GPG...
Detected gpg...
Running apt-get update... done.
Installing /etc/apt/sources.list.d/runner_gitlab-runner.list...done.
Importing packagecloud gpg key... done.
Running apt-get update... done.
Running apt-get upgrade... done.
Running gitlab-runner --version
version: 18.3.0
Git revision: 9ba718cd
Git branch: 18-3-stable
GO version: go1.24.4 Xmacospro
Build date: 2024-02-23T11:12Z
OS/Arch: linux/amd64
root@prsdkulobar:~# sudo systemctl enable gitlab-runner
root@prsdkulobar:~# systemctl start gitlab-runner
root@prsdkulobar:~# gitlab-runner register
Runtime platform
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com):
https://gitlab.com

```

Gambar 6 konfigurasi gitlab runner

180  
181  
182  
183  
184  
185  
186

### 3.1.4. Konfigurasi .gitlab-ci.yml

Pada gitlab proses CI/CD hanya bisa berjalan apabila terdapat file .gitlab-ci.yml didalam project yang akan di deploy. File ini berisi perintah yang akan dijalankan oleh pipeline. Pada penelitian ini dilakukan konfigurasi deploy sederhana untuk menguji apakah proses deployment dapat berjalan dengan baik serta untuk mengetahui lama waktu yang dibutuhkan pada satu kali push melalui lokal developper. Adapun code yang diimplementasikan pada konfigurasi ini dapat dilihat pada listing code dibawah ini.

```

deploy :
  stage: deploy
  image: alpine:latest
  before_script:
    - apk add --no-cache openssh-client rsync bash sshpass
      # Menambahkan host key server ke known_hosts
    - mkdir -p ~/.ssh
    - ssh-keyscan -H $DEPLOY_IP >> ~/.ssh/known_hosts
    # - COMPOSER_ALLOW_SUPERUSER=1 composer install
  script:
    - sshpass -p "$DEPLOY_PASS" rsync -avz --delete --exclude='*.git/' --exclude='storage/' --exclude='*.env' ./ $DEPLOY_USER@$DEPLOY_IP:$DEPLOY_PATH
      - sshpass -p "$DEPLOY_PASS" ssh -o StrictHostKeyChecking=no $DEPLOY_USER@$DEPLOY_IP "cd $DEPLOY_PATH && COMPOSER_ALLOW_SUPERUSER=1 composer install --no-dev --optimize-autoloader"
    - echo "deployment Finished"
  only:
    - master
  when: on_success

```

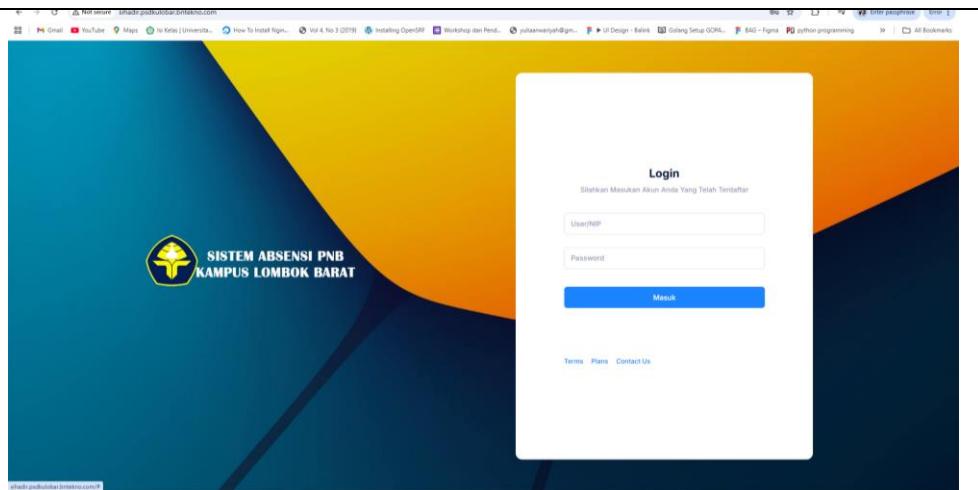
187  
188  
189

190

191  
192  
193  
194

### 3.2. Hasil Uji Deployment

Berdasarkan hasil pengujian yang dilakukan implementasi fase deployment menggunakan aliran CI/CD dapat berjalan dengan baik. Hal ini ditunjukkan dengan sistem yang mampu diakses dengan baik melalui domain yang telah ditetapkan seperti terlihat pada gambar 7.



Gambar 7 hasil deployment sistem absensi online

#### 4. Pembahasan

Berdasarkan hasil pengujian yang telah dilakukan pada penelitian ini didapatkan hasil bahwa proses implementasi CI/CD pada pembangunan sistem absensi online dapat berjalan dengan baik. Dinilai dari lama eksekusi pada proses deployment dapat dikatakan cukup baik. Berdasarkan 5 kali pengujian yang dilakukan (tabel 1) dapat dilihat bahwa rata – rata waktu eksekusi yang dibutuhkan untuk sekali proses push ialah sebanyak 42,6 detik. Lama waktu eksekusi dipengaruhi oleh besar file project yang dideploy dan kompleksitas sistem yang dibangun. Berdasarkan waktu eksekusi yang dibutuhkan dapat dikatakan bahwa proses deployment dengan menggunakan metode CI/CD cukup efisien. Hal ini sejalan dengan penelitian [6] yang menyampaikan bahwa proses deployment dengan menggunakan CI/CD mampu mengurangi proses dan dapat meningkatkan kinerja.

Tabel 1. Durasi deployment

No	Pengujian	Lama Durasi Eksekusi
1	Uji 1 penambahan gitlab-ci (test runner)	35 detik
2	Uji 2 menambahkan stage deploy	41 detik
3	Uji 3 Perbaikan code deploy	47 detik
4	Uji 4 perubahan layout	45 detik
5	Uji 5 perubahan	45 detik
Rata rata waktu		42,6 detik

Pada penelitian lain yang dilakukan oleh [7] juga menunjukkan bahwa implementasi CI/Cd pada proses deployment mampu mengurangi waktu pengembangan, waktu pengembangan dan mempercepat proses perbaikan atas code yang telah dibuat. Terlepas dari efektifitas dan keuntungan yang diperoleh implementasi bberapa fase SDLC mulai dari soding, deployment hingga testing bukan tanpa kendala, salah satu kendala yang paling penting ialah terkait pemahaman dan kemampuan enginer dalam melakukan proses integrasi dan pengembangan code CI/CD. Pada uji awal integrasi yang dilakukan terdapat beberapa kesalahan pada code gitlab-ci.yml sehingga pipeline gagal tereksekusi dan proses deployment tidak dapat berjalan. Seperti terlihat pada gambar 8 terjadi error akibat kesalahan pada proses identifikasi hostname sehingga proses pembukaan akses menuju server melalui SSH gagal.

```

40 executing busybox-1.37.0-r18.trigger
41 OK: 15 MiB in 38 packages
42 $ mkdir -p ~/.ssh
43 $ echo "$SSH_PRIVATE_KEY" | tr -d '\r' > ~/.ssh/id_ed25519
44 $ chmod 600 ~/.ssh/id_ed25519
45 $ echo "$SSH_KNOWN_HOSTS" > ~/.ssh/known_hosts
46 $ chmod 644 ~/.ssh/known_hosts
47 $ rsync -az --delete \# collapsed multi-line command
48 ssh: Could not resolve hostname : Name does not resolve
49 rsync: connection unexpectedly closed (0 bytes received so far) [sender]
50 rsync error: unexplained error (code 255) at io.c(232) [sender=3.4.1]
51 Cleaning up project directory and file based variables
52 ERROR: Job failed: exit code 255

```

Gambar 8 tampilan hasil eksekusi pipeline job

Kendala ini juga disampaikan oleh [8] yang menyatakan bahwa tantangan dalam pengimplementasian CI/CD ialah terkait resistansi perubahan, kurangnya keahlian dalam penggunaan tools CI/CD serta kesulitan dalam proses kolaborasi *cross functional*. Melalui analisis ini diketahui bahwa proses implementasi CI/CD memerlukan keahlian yang memadai mulai dari konsep, penguasaan tools dan semua hal yang menunjang proses implementasi ini. Pada penelitian selanjutnya perlu dilakukan ujicoba implemntasi CI/CD pada model pipeline job yang lebih kompleks dan menggunakan kompleksitas project yang lebih tinggi juga, sehingga pengukuran terhadap kinerja impelementasinya pun menjadi lebih baik. Ada penelitian selanjutnya juga perlu dilakukan integrasi proses pengujian /unit test didalam stage pipeline serta integrasi uji keamanan perlindungan dipertimbangkan.

## 5. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan, dapat disimpulkan bahwa implementasi CI/CD pada fase pengembangan dan deployment sistem absensi online telah berhasil dilaksanakan dengan baik. Penerapan otomatisasi ini mampu memangkas waktu operasional secara signifikan, di mana rata-rata waktu yang dibutuhkan untuk proses deployment hanya sebesar 42 detik. Selain meningkatkan efisiensi waktu, penggunaan CI/CD terbukti dapat mengurangi intervensi manual serta meningkatkan kinerja pengembang secara keseluruhan. Meskipun memberikan keuntungan teknis yang besar, implementasi ini menghadapi tantangan pada aspek sumber daya manusia, terutama terkait penguasaan tools, kemampuan pengkodean, serta adaptasi terhadap perubahan budaya kerja dari metode tradisional ke budaya DevOps.

## Referensi

- [1] M. I. H. -, "Software Development Life Cycle (SDLC) Methodologies for Information Systems Project Management," *Int. J. Multidiscip. Res.*, vol. 5, no. 5, pp. 1–36, 2023, doi: 10.36948/ijfmr.2023.v05i05.6223.
- [2] A. Alpery and M. A. F. Ridha, "Implementasi CI/CD Dalam Pengembangan Aplikasi Web Menggunakan Docker dan Jenkins," *Appl. Bus. Eng. Conf.*, pp. 287–296, 2021.
- [3] V. V. Reddy Boda and J. Immaneni, "Optimizing CI/CD in Healthcare: Tried and True Techniques," *Int. J. Emerg. Res. Eng. Technol.*, vol. 3, no. 2, pp. 28–38, 2022, doi: 10.63282/3050-922x.ijeret-v3i2p104.
- [4] M. Ccallo and A. Quispe-Quispe, "Adoption and Adaptation of CI/CD Practices in Very Small Software Development Entities: A Systematic Literature Review," pp. 1–7, 2024.
- [5] Y. Jani, "Implementing Continuous Integration and Continuous Deployment (CI/CD) in Modern Software Development," *Int. J. Sci. Res.*, vol. 12, no. 6, pp. 2984–2987, 2023, doi: 10.21275/sr24716120535.
- [6] J. Jaeni, N. A. S., and A. D. Laksito, "Implementasi Continuous Integration/Continuous Delivery (Ci/Cd) Pada Performance Testing Devops," *J. Inf. Syst. Manag.*, vol. 4, no. 1, pp. 62–66, 2022, doi: 10.24076/joism.2022v4i1.887.
- [7] R. Setiabudi, "Analisis Efektifitas Ci / Cd Dan Manual ( Tradisional ) Dalam Pengembangan Website Rakyatweb.Com," *J. Ismetek*, vol. 17, no. 2, pp. 29–36, 2024.
- [8] I. Dias, R. Wickramarachchi, and S. Jayasinghe, "Key Success Factors for Adoption of CI / CD with Agile Project Management - Systematic Literature Review," *Int. Res. Conf. Smart Comput. Syst. Eng.*, 2025.