



Implementasi *Continuous Integration* dan *Continuous Deployment* pada Pengembangan Sistem Peminjaman Laboratorium Komputer Berbasis Website

Rahmadina ¹, Depandi Enda ²

^{1,2} Jurusan Teknik Informatika, Politeknik Negeri Bengkalis, Indonesia

e-mail: rahmadinaa2002@gmail.com, depandienda@polbeng.co.id

Abstract: *The computer laboratory at Bengkalis State Polytechnic has an important role in supporting teaching and learning activities, especially in the Informatics Engineering Department. However, the existing web-based laboratory lending system is still limited to basic features, such as data recap and schedules, so it is less than optimal. This study aims to develop a laboratory lending system by adding features for checking availability, inventory lending, and notifications, as well as implementing Continuous Integration (CI) and Continuous Deployment (CD) to automate the application development process. The implementation of CI/CD helps developers detect and fix problems faster, increasing productivity and application quality. The results of the study show that the developed system is more reliable, efficient, and feasible to use to support optimal laboratory management.*

Keywords: *Continuous Integration, Continuous Deployment, Computer Laboratory*

Abstrak: Laboratorium komputer di Politeknik Negeri Bengkalis memiliki peran penting dalam mendukung kegiatan belajar mengajar, khususnya di Jurusan Teknik Informatika. Namun, sistem peminjaman laboratorium berbasis website yang ada masih terbatas pada fitur dasar, seperti rekap data dan jadwal, sehingga kurang optimal. Penelitian ini bertujuan untuk mengembangkan sistem peminjaman laboratorium dengan menambahkan fitur pengecekan ketersediaan, peminjaman inventaris, dan notifikasi, serta menerapkan *Continuous Integration* (CI) dan *Continuous Deployment* (CD) untuk mengotomatisasi proses pengembangan aplikasi. Penerapan CI/CD membantu developer mendeteksi dan memperbaiki masalah dengan lebih cepat, meningkatkan produktivitas dan kualitas aplikasi. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan lebih andal, efisien, dan layak digunakan untuk mendukung pengelolaan laboratorium secara optimal.

Kata Kunci: *Continuous Integration, Continuous Deployment, Laboratorium Komputer*

PENDAHULUAN

Dalam era globalisasi sekarang ini, perkembangan teknologi semakin pesat dan arus informasi bergerak semakin cepat. Oleh sebab itu, berbagai bidang berlomba-lomba guna memanfaatkan teknologi informasi dengan baik. Salah satunya adalah bidang Pendidikan yang mulai memanfaatkan teknologi komputer sebagai salah satu konsentrasi dalam keilmuannya. Pemanfaatan komputer sebagai bentuk teknologi dituangkan berupa sarana dan prasarana yaitu laboratorium komputer. Laboratorium merupakan ruangan yang didalamnya terdapat berbagai macam alat dan bahan yang digunakan untuk kepentingan kegiatan [1].

Salah satu pemanfaatan Laboratorium komputer di Jurusan Teknik Informatika Politeknik Negeri Bengkalis berperan sebagai fasilitas pendukung utama dalam kegiatan belajar mengajar. Selain digunakan untuk pembelajaran bersama dosen, mahasiswa juga dapat memanfaatkan laboratorium untuk berbagai keperluan, termasuk menyelesaikan tugas akhir. Saat ini, proses peminjaman laboratorium telah didukung oleh sistem berbasis website. Namun, hasil wawancara dan observasi ulang menunjukkan bahwa sistem tersebut masih memiliki

banyak kekurangan, terutama pada kelengkapan fiturnya. Sistem yang ada saat ini hanya mencakup satu laboratorium dengan fitur-fitur seperti rekap data, jadwal, laporan akhir, validasi serta akses untuk melihat laporan.

Berdasarkan dari permasalahan tersebut, perlu adanya pengembangan pada website peminjaman laboratorium komputer, Pengembangan yang dilakukan, yaitu penambahan beberapa fitur baru seperti fitur melihat ketersediaan laboratorium, peminjaman inventaris laboratorium, serta notifikasi jika ada peminjaman yang masuk. Dengan menerapkan penerapan Continuous Integration (CI) dan Continuous Deployment (CD) sebagai solusi modern untuk meningkatkan efisiensi pengembangan dan pengelolaan sistem. Continuous Integration (CI) memungkinkan pengintegrasian kode secara otomatis ke dalam repositori dengan pengujian yang cepat dan sering, sehingga mempermudah deteksi dan perbaikan kesalahan. Sementara itu, Continuous Deployment (CD) melanjutkan proses setelah Continuous Integration (CI) dengan memastikan aplikasi dapat dibangun dan dirilis secara otomatis [3].

Untuk mengatasi permasalahan yang ada, penelitian sebelumnya oleh Malyda Aulia mengusulkan pengembangan aplikasi sistem peminjaman laboratorium komputer menggunakan metode Rapid Application Development (RAD). Metode ini berfokus pada pengembangan perangkat lunak secara inkremental dengan waktu pengerjaan yang lebih singkat [2]. Namun, metode ini masih memiliki keterbatasan dalam hal pengelolaan proses pengembangan yang berkelanjutan dan pengujian otomatis.

Dalam pengembangan sistem peminjaman laboratorium komputer di Jurusan Teknik Informatika Politeknik Negeri Bengkalis, penerapan Continuous Integration (CI) dan Continuous Deployment (CD) diharapkan dapat meningkatkan efisiensi dan kualitas sistem. Dengan Continuous Integration (CI) dan Continuous Deployment (CD), proses pengembangan menjadi lebih cepat dan terorganisir, karena setiap perubahan pada sistem dapat diuji dan diperbarui secara otomatis. Prinsip utama Continuous Integration (CI) dan Continuous Deployment (CD), seperti integrasi berkelanjutan dan otomatisasi, membantu tim pengembang dalam mengelola proyek dengan lebih baik, mempercepat proses perbaikan jika terjadi kesalahan, serta memastikan sistem tetap stabil. Selain itu, penggunaan Continuous Integration (CI) dan Continuous Deployment (CD) juga memungkinkan pemantauan dan pengelolaan sistem secara lebih efektif, sehingga mendukung kelancaran peminjaman laboratorium bagi mahasiswa dan dosen.

KAJIAN PUSTAKA

1. Sistem Peminjaman Laboratorium Berbasis Website

Sistem peminjaman laboratorium berbasis website merupakan salah satu solusi digital untuk mengelola penggunaan fasilitas laboratorium secara efisien. Nishom dan Wibowo (2020) mengemukakan bahwa sistem ini sangat membantu dalam manajemen jadwal, inventaris, serta laporan peminjaman, yang sebelumnya dilakukan secara manual dan rentan terhadap kesalahan. Dalam konteks pendidikan tinggi, sistem ini menjadi penting karena frekuensi penggunaan laboratorium oleh mahasiswa dan dosen sangat tinggi. Aplikasi serupa juga dikembangkan oleh Wijaya dan Ramadhani (2020), dengan pendekatan prototyping untuk menghasilkan sistem yang dapat diakses lintas platform. Mereka menyoroti bahwa integrasi antarmuka pengguna yang sederhana dengan fitur yang lengkap dapat mempercepat proses pelayanan peminjaman dan pengelolaan aset laboratorium.

2. Metode Rapid Application Development (RAD)

Metode Rapid Application Development (RAD) merupakan pendekatan pengembangan perangkat lunak yang menekankan kecepatan dan fleksibilitas. Menurut Cahyaningtyas et al. (2024), RAD cocok digunakan untuk sistem informasi skala kecil hingga menengah, karena fokus pada iterasi cepat dan pengujian langsung terhadap pengguna. Aulia (2023) menerapkan RAD dalam pengembangan sistem peminjaman laboratorium di lingkungan politeknik, namun menghadapi kendala dalam otomatisasi dan pemeliharaan jangka panjang karena keterbatasan metode tersebut dalam integrasi proses berkelanjutan.

3. Continuous Integration (CI)

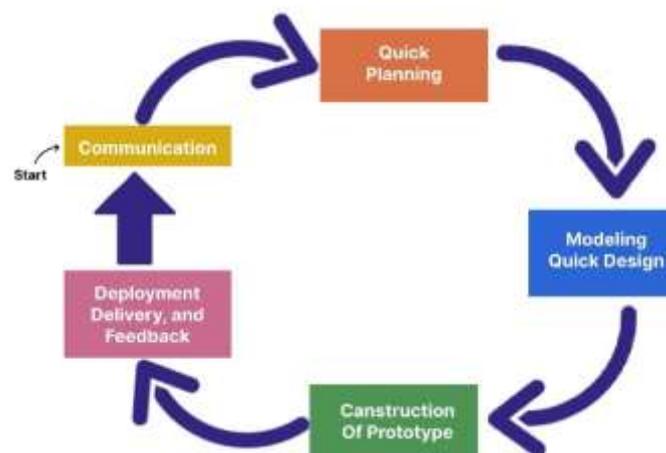
Continuous Integration (CI) adalah praktik pengembangan perangkat lunak yang mendorong penggabungan kode secara berkala ke dalam repositori pusat. Menurut Firmansyah dalam penelitian ini, setiap integrasi diuji secara otomatis untuk mendeteksi kesalahan sejak dini. Hal ini meningkatkan kecepatan siklus pengembangan dan mengurangi risiko kerusakan sistem akibat konflik kode atau kesalahan sintaks. CI juga memungkinkan transparansi dalam pengelolaan proyek karena seluruh anggota tim dapat melihat dan memantau perubahan secara real-time. Putra Apriliyanto et al. (2024) menambahkan bahwa penerapan GitHub Actions dalam CI dapat menghemat waktu pengujian dan mempercepat rilis perangkat lunak tanpa mengorbankan kualitas.

4. Continuous Deployment (CD)

Continuous Deployment (CD) melanjutkan prinsip CI dengan mengotomatiskan proses pengiriman aplikasi ke lingkungan produksi setelah kode berhasil diuji. CD memastikan bahwa perangkat lunak selalu dalam kondisi siap rilis kapan pun dibutuhkan. Noviantama dan Wahyu (2021) menyatakan bahwa CD sangat bermanfaat dalam sistem berbasis web yang memerlukan pembaruan rutin dan cepat. Implementasi CD mengurangi ketergantungan pada proses manual, yang rentan terhadap kelalaian dan keterlambatan. Dalam konteks sistem peminjaman laboratorium, CD memungkinkan fitur baru seperti notifikasi otomatis dan pengecekan ketersediaan laboratorium langsung diterapkan begitu selesai dikembangkan dan diuji.

5. Prototyping dalam Pengembangan Sistem

Metode prototyping adalah pendekatan pengembangan sistem yang menekankan pada pembuatan model awal (prototype) untuk diuji dan dievaluasi pengguna sebelum dikembangkan sepenuhnya. Aditya et al. (2021) menunjukkan bahwa metode ini efektif dalam sistem peminjaman karena memungkinkan pengguna memberikan umpan balik langsung terkait fitur-fitur yang dibutuhkan. Metode ini juga mendorong kolaborasi antara pengembang dan pengguna akhir sehingga aplikasi yang dihasilkan lebih tepat guna.



Gambar 1 Metode Prototyping

METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan metode prototyping, yang memungkinkan pengembangan sistem dilakukan secara iteratif berdasarkan masukan pengguna.

1. Pengumpulan Data

Data diperoleh melalui wawancara langsung dengan laboran di Jurusan Teknik Informatika Politeknik Negeri Bengkalis, untuk mengetahui kebutuhan sistem dan kekurangan dari sistem peminjaman laboratorium yang telah ada.

2. Perancangan Sistem

Tahapan ini meliputi:

- Pembuatan desain antarmuka menggunakan *mockup tools*.
- Perancangan database.
- Implementasi fitur utama seperti: pengecekan ketersediaan laboratorium, peminjaman inventaris, dan notifikasi otomatis.

3. Implementasi CI/CD

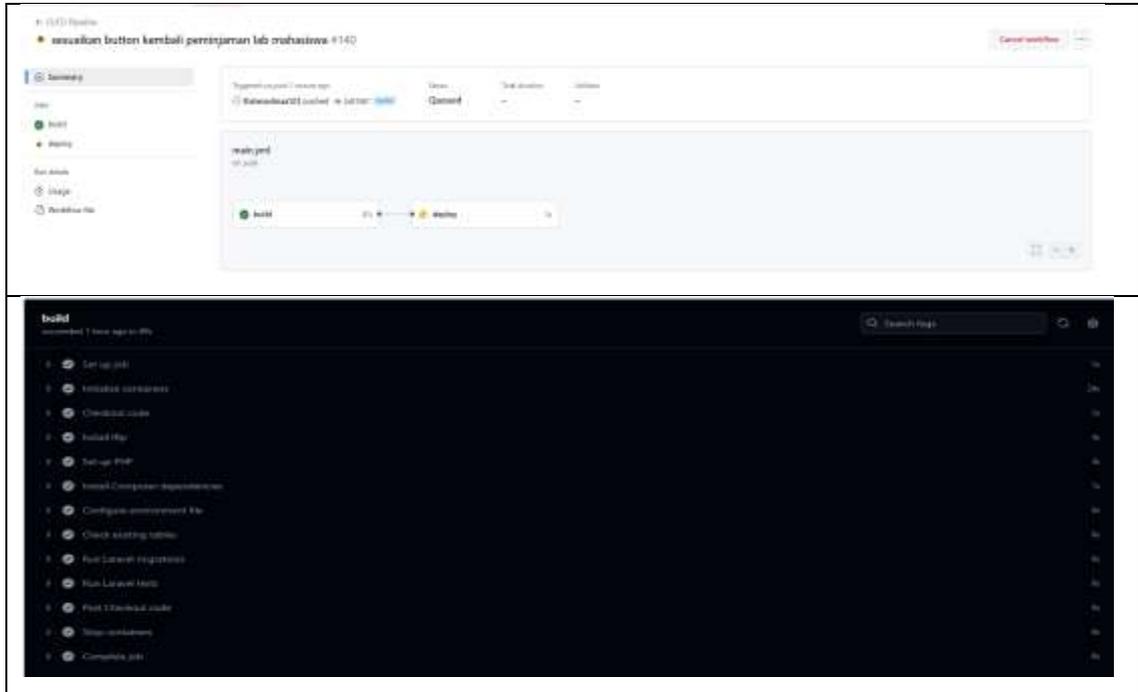
Proses pengembangan sistem didukung oleh penerapan Continuous Integration dan Continuous Deployment (CI/CD) menggunakan GitHub Actions. CI digunakan untuk menguji kode secara otomatis setiap ada perubahan, sedangkan CD digunakan untuk mendistribusikan aplikasi secara otomatis ke server.

4. Pengujian dan Evaluasi

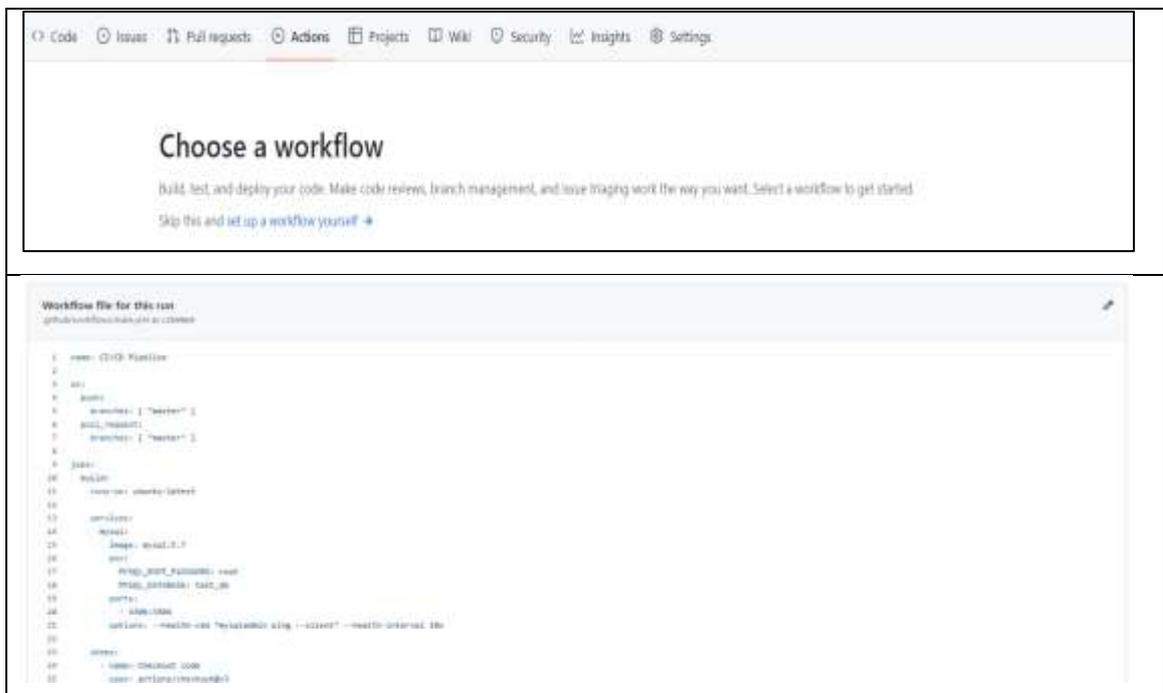
Sistem diuji menggunakan metode pengujian fungsional dan performa, termasuk metrik seperti *build success rate*, *deployment time*, dan stabilitas aplikasi.

HASIL DAN PEMBAHASAN

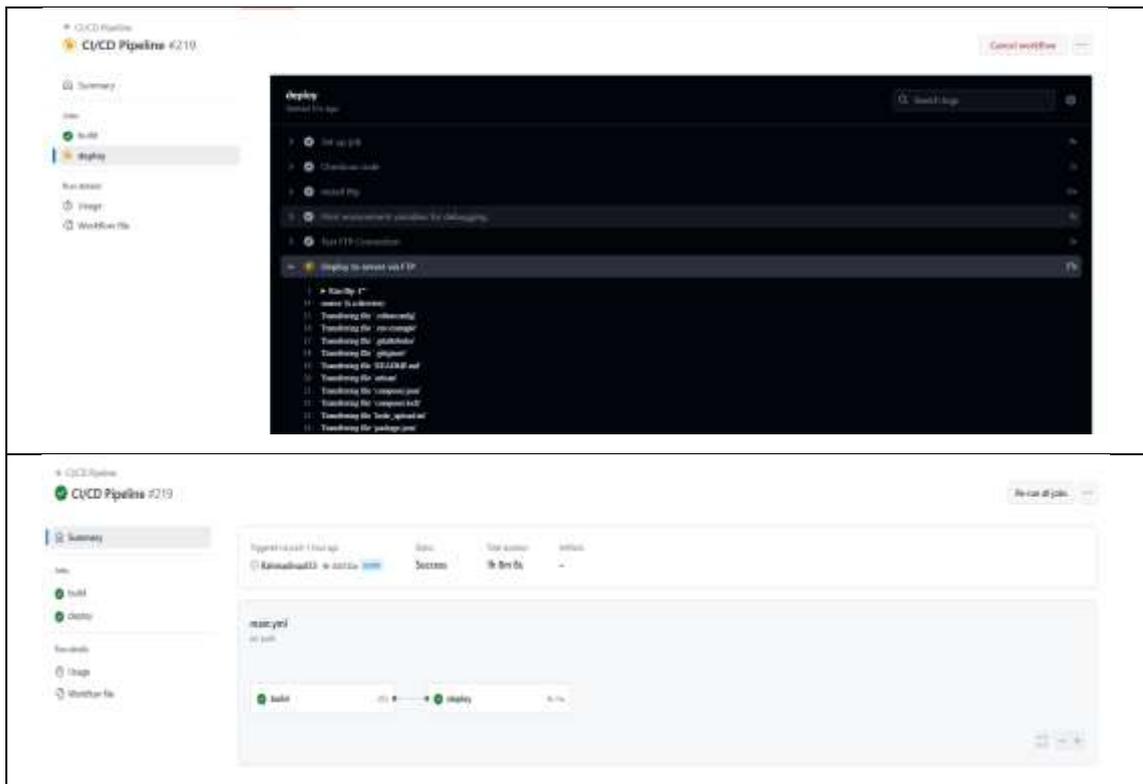
Hasil



Gambar 2 Hasil Proses Build



Gambar 3 Proses WorkFlow



Gambar 4 Proses Hasil Deploy

SIPELA

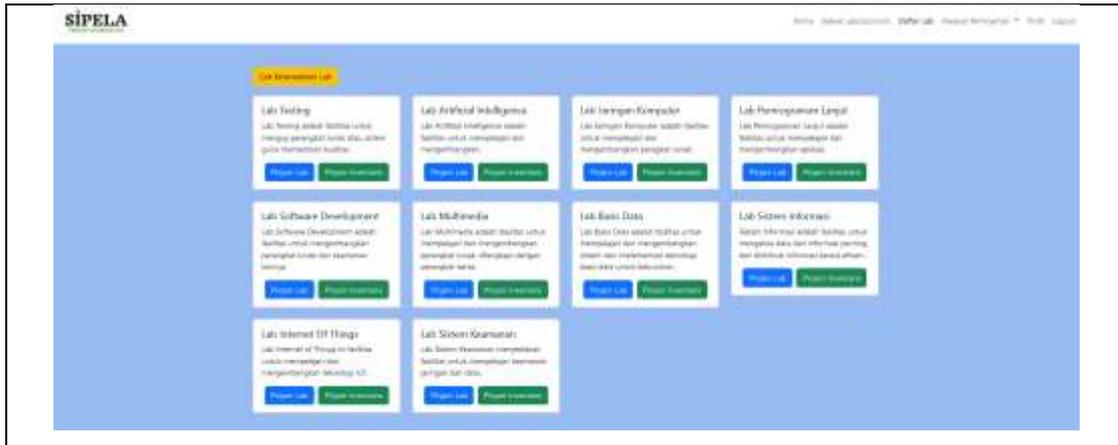
Home Jadwal Laboratorium Daftar Lab Bantu Berlangganan Profil Logout

Jadwal Laboratorium

Choose a Lab:
Lab Software Development

Hari	Tanggal	Jam	Kelas	Mata Kuliah	Dosen	Lab
Senin	2025-02-03	08:00 - 10:00	3A	Dasar pemrog	Lidyia wati, M.Kom	Lab Software Development
	2025-02-03	08:00 - 10:00	7a	web 2	Deponil Enda, M.Kom	Lab Software Development
	2025-02-03	13:00 - 15:00	3a	Komunikasi Data Iq	Agus Teddyana, M.Kom	Lab Software Development
	2025-02-03	13:00 - 15:00	3a	Komunikasi Data	Agus Teddyana, M.Kom	Lab Software Development
Selasa	2025-02-04	08:00 - 12:00	3a	perograman	osep subandi, M.Kom	Lab Software Development

Gambar 5 Hasil Tampilan Jadwal lab



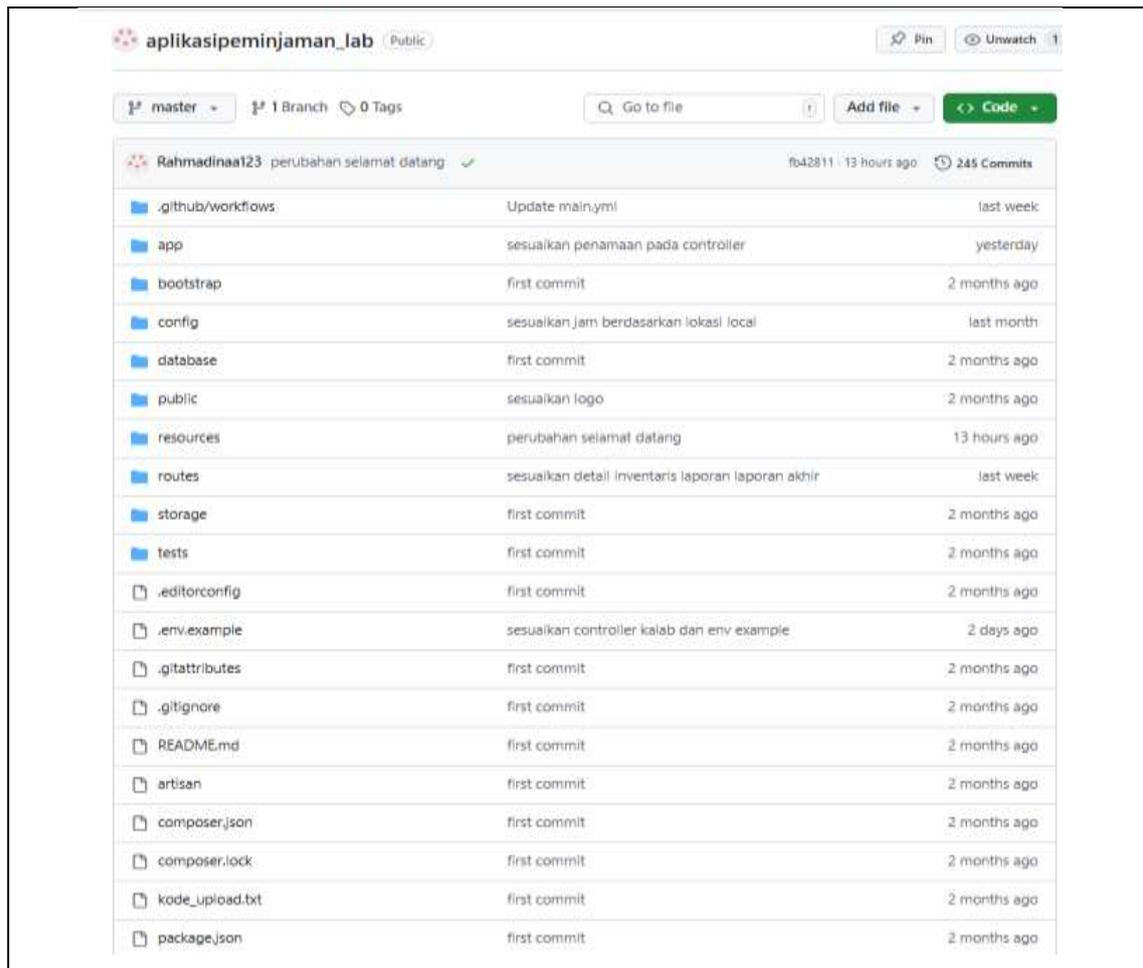
Gambar 6 Hasil Tampilan Daftar Lab

Pembahasan

Penelitian ini mengimplementasikan Continuous Integration (CI) dan Continuous Deployment (CD) pada pengembangan aplikasi peminjaman laboratorium komputer. Dengan CI/CD, setiap perubahan kode yang dilakukan akan diuji dan diterapkan secara otomatis ke dalam aplikasi, mempercepat proses pengembangan dan menjaga kualitas aplikasi. Evaluasi dilakukan untuk mengukur seberapa efektif dan efisien penerapan CI/CD ini dalam mempercepat alur pengembangan, memastikan aplikasi tetap berjalan lancar, dan memudahkan pengelolaan laboratorium komputer.

1. Proses Integrasi (CI)

Pada bagian ini akan diterapkan proses perubahan kode maupun penambahan fitur pada aplikasi peminjaman laboratorium komputer. Sejak awal proyek ini di-push ke GitHub, sudah ada beberapa perubahan yang dilakukan pada kode dan juga penambahan fitur, di antaranya pembaruan halaman peminjaman laboratorium, pembaruan fitur peminjaman inventaris, penambahan notifikasi yang mengarah ke halaman notifikasi, upgrade dependencies, https://github.com/Rahmadinaa123/aplikasipeminjaman_lab. Semua perubahan tersebut telah diuji dan dipust ke repository di Github dibawah ini adalah gambar dari tampilan Github proyek yang telah di-pust dan di-update setelah dilakukan pengujian.



Gambar 7 Proses Integrasi (CI)

Dari awal proyek ini dimulai dengan push pertama pada tanggal 20 November. Hingga saat ini, telah dilakukan sebanyak 234 commit yang mencakup berbagai perubahan, mulai dari penambahan fitur baru, perbaikan bug, hingga pengoptimalan aplikasi. Setiap commit mencerminkan proses pengembangan yang terus berlangsung untuk menyempurnakan aplikasi ini. Berikut adalah daftar lengkap commit beserta penjelasan dan gambar yang menggambarkan perubahan yang telah dilakukan di setiap tahap:

1. Commit pada 20 November 2024

Pada tanggal 20 November 2024, penulis melakukan pengiriman (push) kode ke repositori GitHub. Tindakan ini dilakukan untuk mengunggah proyek peminjaman laboratorium komputer ke GitHub sekaligus memperbarui dan menyimpan perubahan terbaru pada proyek perangkat lunak yang sedang dikembangkan. Kode yang diunggah mencakup perbaikan pada halaman Update main.yml dan fitur jadwal.



Gambar 8 Commits pada 20 Nov 2024

2. Commits pada 21 November 2024

Pada tanggal 21 November 2024, penulis melakukan push kode ke repositori GitHub. Tindakan ini dilakukan untuk memperbarui sekaligus menyimpan perubahan terbaru pada proyek perangkat lunak yang sedang dikembangkan. Kode yang di-push berisi perbaikan bug pada halaman detail peminjaman inventaris dan detail peminjaman laboratorium.



Gambar 9 Commits pada 21 Nov 2024

3. Commits pada 22 November 2024

Pada tanggal 22 November 2024, penulis melakukan pengiriman (push) kode repositori Github. Tindakan ini dilakukan untuk memperbarui dan menyimpan perubahan terbaru dalam proyek perangkat lunak yang sedang dikembangkan. Kode yang dipush adalah menambahkan controller jadwal lab untuk actor kalab, Kemudian menambahkan route jadwal laboratorium.



Gambar 10 Commits pada 22 Nov 2024

4. Commits pada 03 Desember 2024

Pada tanggal 03 Desember 2024, penulis melakukan pengiriman (push) kode repositori Github. Tindakan ini dilakukan untuk memperbarui dan menyimpan perubahan terbaru dalam proyek proses mencoba mengimplemmentasikan CI/CD di github actions. Kode yang di push yaitu penambahan halaman, controller serta route edit inventaris laboratorium.



Gambar 11 Commits pada 03 Des 2024

2. Commits pada 09 Desember 2024

Pada tanggal 09 Desember 2024, penulis melakukan pengiriman (push) kode ke repositori Github. Tindakan ini dilakukan untuk memperbarui dan menyimpan perubahan terbaru dalam proyek perangkat lunak yang sedang dikembangkan. Kode yang di-push untuk halaman edit peminjaman inventaris laboran.



Gambar 12 Commits pada 09 Des 2024

3. Commits pada 22 Desember 2024

Pada tanggal 22 Desember 2024, penulis melakukan pengiriman (push) kode ke repositori Github. Tindakan ini dilakukan untuk memperbarui dan menyimpan perubahan terbaru dalam proyek perangkat lunak yang sedang dikembangkan. Kode yang di-push yaitu perubahan pada edit peminjaman laboratorium pada actor kalab kemudian menyesuaikan form edit peminjaman inventaris laboratorium. Lalu menyesuaikan tambah laboratorium agar tidak bisa pinjam di halaman yang sama.



Gambar 13 Commits pada 22 Des 2024

4. Commits pada tanggal 02 Januari 2025

Pada tanggal 02 Januari 2025, penulis melakukan pengiriman (push) kode ke repositori Github. Tindakan ini dilakukan untuk memperbarui dan menyimpan perubahan terbaru dalam proyek perangkat lunak yang sedang dikembangkan. Kode yang di-push merupakan penyesuaian fitur notifikasi peminjaman lab dan notifikasi peminjaman inventaris laboratorium.



Gambar 14 Commits pada 02 Jan 2025

Setelah semua perubahan kode di-*push*, langkah selanjutnya adalah membuat folder baru dengan nama `.github/workflows/` dan menambahkan file bernama `main.yml`. File ini digunakan untuk menyusun konfigurasi agar proses CI/CD dapat berjalan secara otomatis menggunakan GitHub Actions. GitHub Actions akan menjalankan proses *build* secara otomatis setiap kali ada perubahan atau penambahan fitur yang di-*push* ke repositori. Proses ini menjalankan perintah yang telah ditentukan dalam kode dibawah ini.

Potongan script

```
name:
  CI/CD
  Pipeline

on:
  push:
    branches: [ "master" ]
  pull_request:
    branches: [ "master" ]

jobs:
  build:
    runs-on: ubuntu-latest

  services:
    mysql:
      image: mysql:5.7
      env:
        MYSQL_ROOT_PASSWORD: root
```

```
MYSQL_DATABASE: test_db

ports:
  - 3306:3306

options: --health-cmd "mysqladmin ping --silent" --health-interval 10s

steps:
  - name: Checkout code
    uses: actions/checkout@v3
```

B. Studi Kasus Penerapan CI/CD

Pada studi kasus ini terdapat beberapa tahapan dalam penerapan proses Continuous Integration dan Continuous Deployment (CI/CD). Dalam melakukan perubahan kode di dalam repository github. Proses ini dilakukan dengan cara sebagai berikut:

1. Proses perubahan kode

Pada proses perubahan kode, tahap ini digunakan untuk memperbarui atau mengubah kode yang sebelumnya menampilkan pesan “Welcome To”. Perubahan ini dilakukan untuk memperbarui teks, sesuai kebutuhan proyek atau permintaan pengguna. Misalnya, pesan dirubah menjadi “Selamat Datang”.





```

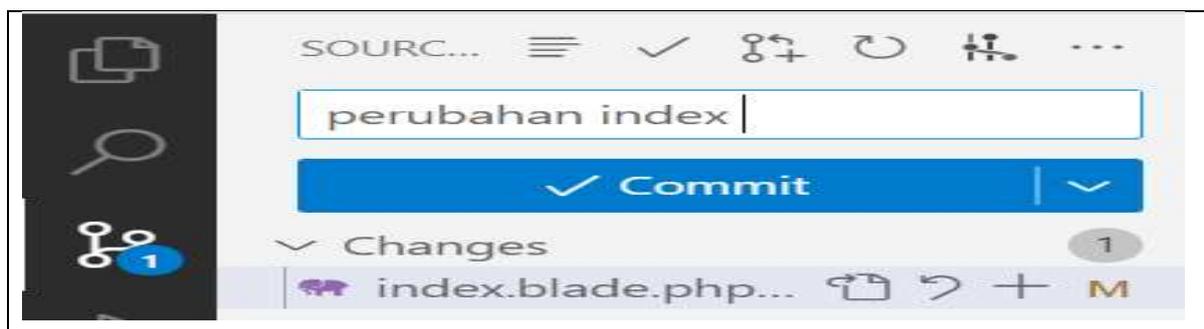
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 <div class="main">
59 <!-- Hero Section -->
60 <section id="hero" class="hero section" style="background-color: #f8d9f3;">
61 <div class="container">
62 <div class="row justify-content-center justify-content-lg-between">
63 <div class="col-12 col-lg-5 text-center">
64 <h1 data-aos="fade-up" data-aos-delay="200">Selamat Datang</h1>
65 <h2 data-aos="fade-up" data-aos-delay="200">Sistem Peninjauan Laporan
66 <h3 data-aos="fade-up" data-aos-delay="200">Politeknik Negeri
67 <h4 data-aos="fade-up" data-aos-delay="200">Bengkalis</h4>
68 <p data-aos="fade-up" data-aos-delay="200">Silahkan untuk menggunakan Laboratori
69 <p data-aos="fade-up" data-aos-delay="200">mudah! Segera Login atau Daftar untuk memulai proses peninjauan sekarang!
70 </div>
71 <div class="col-12 col-lg-5 text-center">
72 <div data-aos="fade-up" data-aos-delay="200">
73 <button class="btn btn-primary" href="#register">Register</button>
74 <button class="btn btn-primary" href="#login">Login</button>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>

```

Gambar 15 Perubahan Kode

2. Proses Melakukan Push/Commit

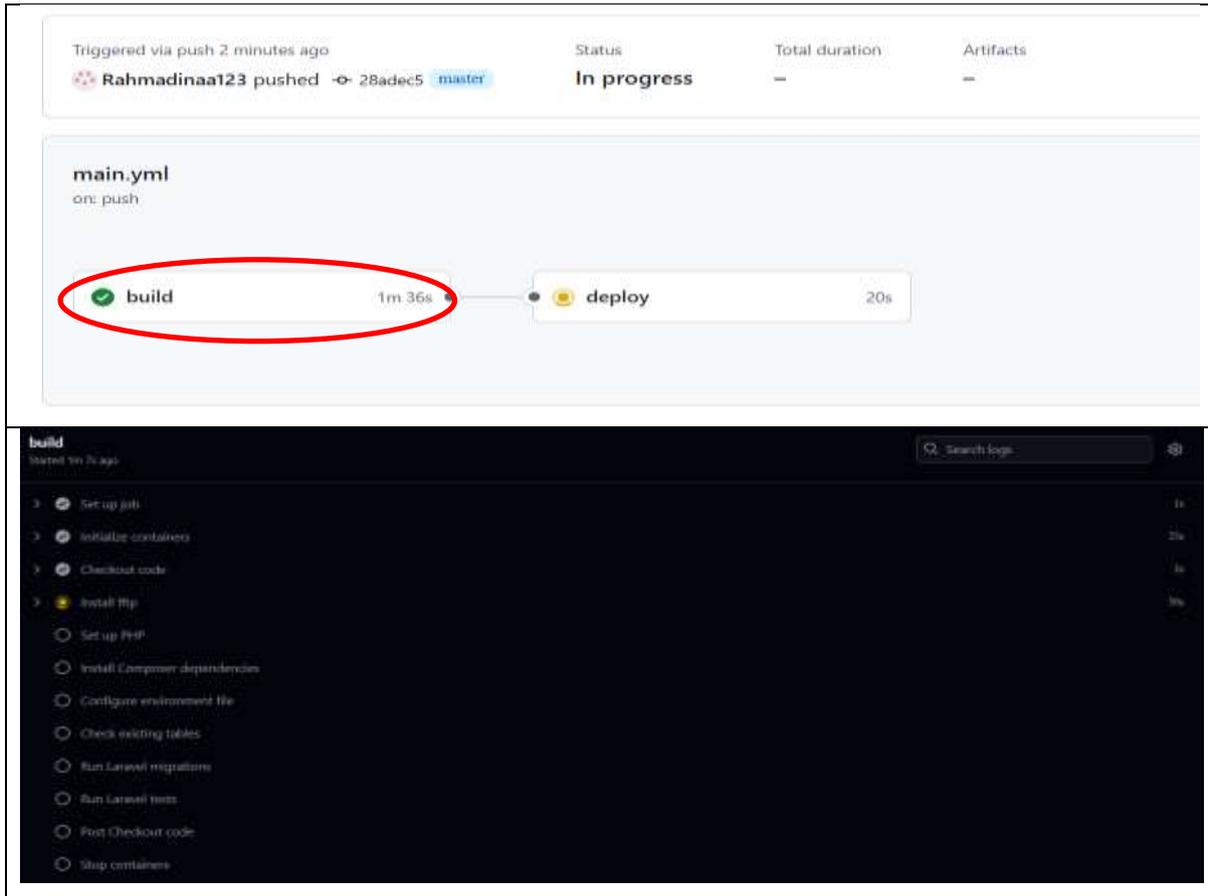
Pada proses push, tahap ini dilakukan setelah perubahan kode berhasil diperbarui atau diubah sesuai kebutuhan. Setelah memastikan bahwa kode telah memenuhi persyaratan dan berjalan dengan baik, pengembang melakukan commit untuk mencatat perubahan tersebut di sistem version control. Selanjutnya, perubahan tersebut diunggah (push) ke repository Github.



Gambar 16 Proses Melakukan Push/Commit

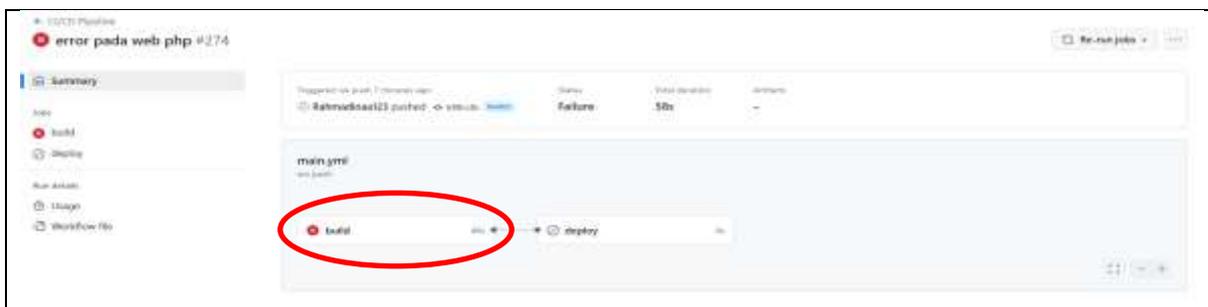
3. Proses Build

Proses build dimulai setelah perubahan kode diunggah (push) ke repository. Sistem Continuous Integration dan Continuous Deployment (CI/CD) secara otomatis memulai proses build untuk menguji kode yang telah diubah. Tahapan ini mencakup kompilasi kode (jika diperlukan), pengujian unit, dan pengujian integrasi untuk memastikan bahwa perubahan tidak menyebabkan kegagalan atau masalah pada aplikasi. Proses build biasanya memerlukan waktu sekitar 1 menit untuk menyelesaikannya. Jika terjadi kesalahan selama proses build, kesalahan tersebut akan ditampilkan, dan proses build akan dihentikan sebelum dilanjutkan ke tahap deployment.



Gambar 17 Proses Build Berhasil

Jika terjadi kesalahan saat proses build, error akan muncul ketika kode yang telah diubah dipush. Build secara otomatis akan memeriksa skrip atau kode yang diunggah. Jika terdapat kesalahan dalam kode, proses build akan gagal dan tidak dapat dilanjutkan ke tahap deployment. Oleh karena itu, sebelum melakukan push, pastikan kode telah diuji dan diperiksa dengan baik untuk menghindari error yang dapat menghambat proses deployment.





Gambar 18 Proses Build Error

4. Proses Deploy

Proses deploy dilakukan setelah proses build selesai dan berhasil tanpa adanya kesalahan. Pada tahap ini, aplikasi yang dihasilkan dari proses build akan diterapkan ke server atau lingkungan yang telah ditentukan. Proses deploy mencakup pengiriman artefak hasil build, konfigurasi sistem, dan pengaturan layanan aplikasi agar berjalan sesuai dengan kebutuhan produksi atau pengujian. Jika terjadi kesalahan selama proses deploy, proses tersebut akan dihentikan, dan koneksi ke server tidak akan berhasil. Namun, jika tidak ada kesalahan, proses deploy akan dilanjutkan, meskipun memerlukan waktu 7 menit untuk menghubungkan aplikasi ke server hosting yang telah terintegrasi dengan repository GitHub.

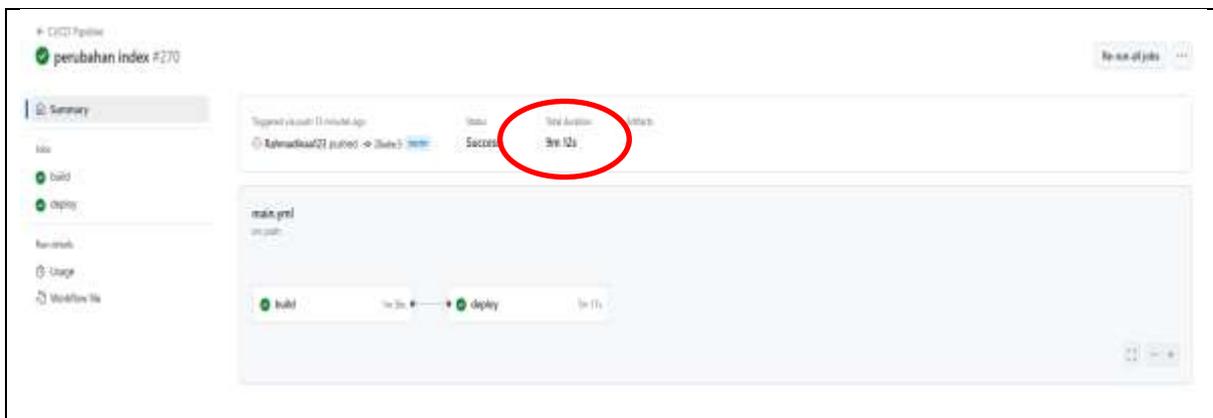




Gambar 19 Proses Deploy

5. Hasil Proses Build dan Deploy

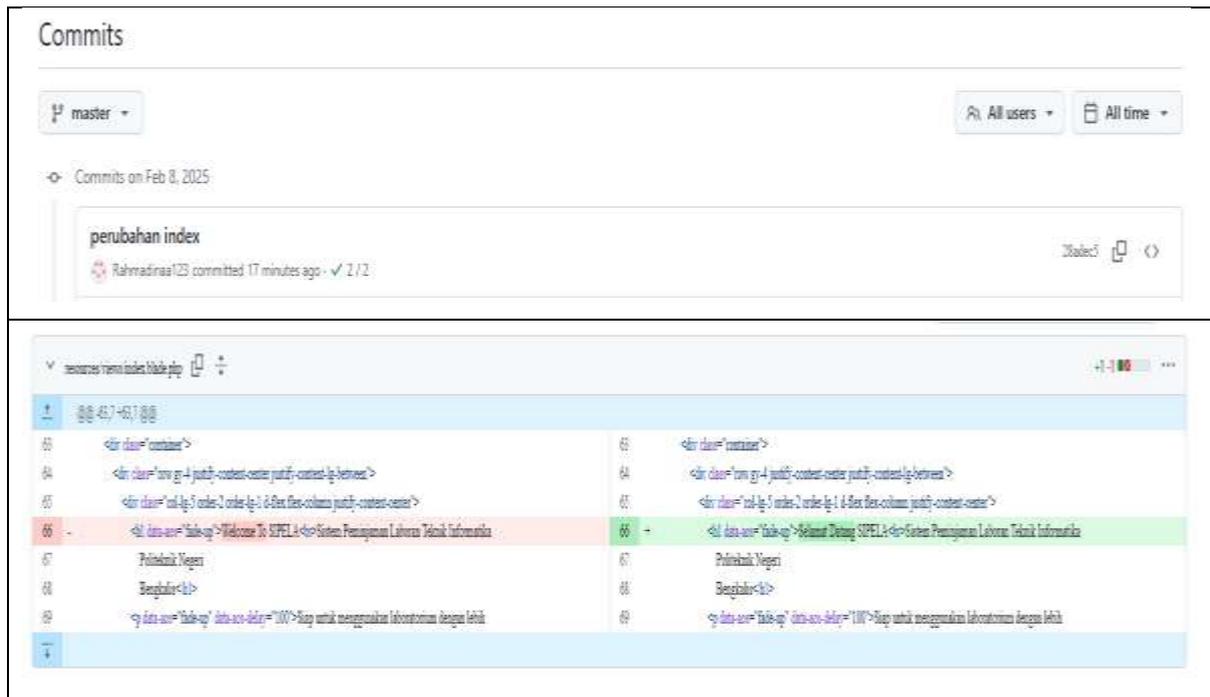
Ketika proses build dan deploy selesai tanpa adanya kesalahan, keduanya akan terhubung, memungkinkan artefak hasil build diintegrasikan ke lingkungan target melalui proses deploy. Jika kedua proses ini berjalan lancar, aplikasi akan berhasil diimplementasikan dan siap digunakan di lingkungan yang ditentukan, seperti staging atau production. Dengan demikian, seluruh rangkaian proses build dan deploy dapat dianggap selesai, memastikan bahwa aplikasi telah melewati tahap pengujian dan implementasi dengan baik tanpa kendala. Proses penyelesaian build dan deploy ini biasanya membutuhkan waktu sekitar 9 menit tergantung pada kompleksitas aplikasi dan lingkungan yang digunakan.



Gambar 20 Hasil Proses Build dan Deploy

6. Hasil Perubahan Kode Selesai

Setelah proses build dan deploy selesai, perubahan kode yang dilakukan oleh pengembang akan terlihat di GitHub repository. Perubahan tersebut mencakup riwayat commit, pull request, dan status pipeline CI/CD, yang memberikan gambaran lengkap tentang perjalanan kode dari tahap pengembangan hingga penerapan ke lingkungan server.



Gambar 21 Hasil Perubahan Kode Selesai

C. Evaluasi proses deployment

Setelah aplikasi berhasil di-deploy ke server, pengguna dapat melihat laporan kinerja aplikasi yang digunakan oleh mahasiswa, laboran, dan ketua laboratorium. Laporan ini mencakup analisis data terkait performa aplikasi, yang menjadi dasar untuk mengevaluasi area yang memerlukan perbaikan, menilai keberhasilan fitur-fitur baru, dan mengidentifikasi potensi peningkatan. Hasil analisis ini akan membantu dalam pengambilan keputusan strategis terkait pengembangan dan penyempurnaan perangkat lunak di masa mendatang. Selain itu, laporan juga akan menampilkan informasi jika terdapat masalah pada stabilitas performa, aksesibilitas, keamanan, maupun tingkat kepercayaan terhadap aplikasi, berikut adalah penjelasan gambar laporan metrik kinerja web yang dihasilkan melalui alat pengembang (DevTools) di browser.

1. LCP (Largest Contentful Paint): 1.25 detik

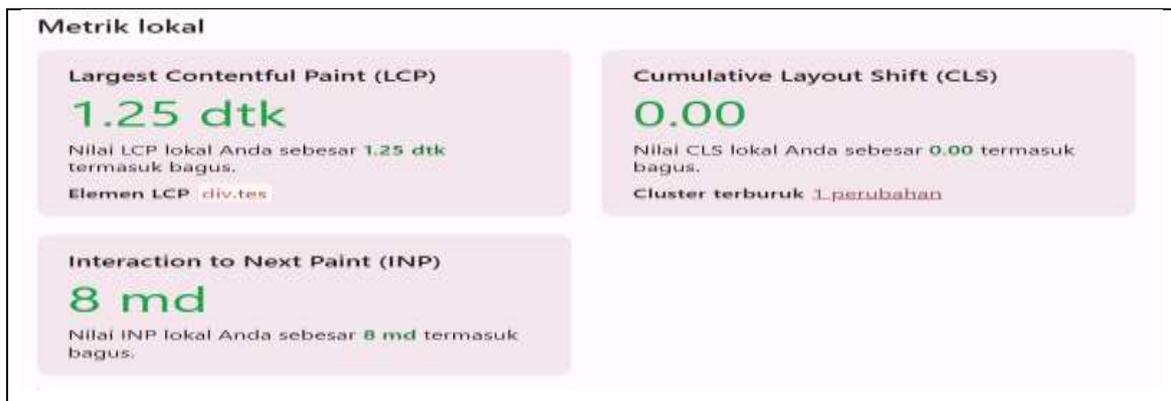
Metrik ini mengukur waktu yang dibutuhkan untuk membuat elemen terbesar yang terlihat di layar, seperti gambar atau teks utama. Nilai 1.25 detik tergolong sangat baik, karena berada di bawah batas 2,5 detik, yang menunjukkan bahwa halaman memuat konten utama dengan cepat.

2. CLS (Cumulative Layout Shift): 0.00

Metrik ini menilai kestabilan tata letak halaman selama proses pemuatan. Nilai 0.00 berarti tata letak halaman stabil tanpa pergeseran elemen yang tidak terduga, sehingga memberikan pengalaman pengguna yang ideal. Nilai ini menunjukkan tidak ada gangguan visual akibat perubahan tata letak.

3. INP (Interaction to Next Paint): 8 ms

Metrik ini mengukur waktu respons halaman terhadap interaksi pengguna, seperti klik atau input. Nilai 8 ms menunjukkan bahwa halaman sangat responsif (ideal di bawah 200 ms), sehingga memberikan pengalaman yang halus dan cepat bagi pengguna.



Gambar 22 Hasil Laporan Performa

KESIMPULAN DAN SARAN

Kesimpulan

Penerapan Continuous Integration dan Continuous Deployment (CI/CD) membuat proses pengembangan aplikasi peminjaman laboratorium komputer menjadi lebih cepat, terstruktur, dan minim kesalahan. Continuous Integration (CI) membantu memastikan aplikasi dapat dibangun dan dijalankan dengan baik serta meminimalkan kemungkinan kesalahan kode sebelum digunakan oleh pengguna. Sementara itu, Continuous Deployment (CD) memungkinkan aplikasi yang sudah teruji langsung diterapkan di server atau hosting dengan cepat dan mudah. Dengan pendekatan ini, tim pengembang dapat bekerja lebih efisien, mempercepat peluncuran pembaruan atau fitur baru, dan menjaga kualitas aplikasi tetap tinggi. Selain itu, Continuous Integration dan Continuous Deployment (CI/CD) membantu aplikasi lebih mudah beradaptasi dengan perubahan dan kebutuhan baru, sehingga pengguna mendapatkan pengalaman yang lebih baik dan sistem yang selalu andal.

Saran

Aplikasi peminjaman laboratorium ini masih memiliki beberapa kekurangan dan perlu dilakukan pengembangan serta penyempurnaan lebih lanjut, khususnya dalam mengimplementasikan Continuous Integration (CI) dan Continuous Deployment (CD). Hal ini bertujuan agar aplikasi mendapatkan hasil yang maksimal untuk kedepannya. Adapun saran untuk penelitian ini selanjutnya sebagai berikut:

- a. Aplikasi dapat ditingkatkan dengan menambahkan fitur notifikasi otomatis untuk peminjaman yang mendekati tenggat waktu. Fitur ini akan membantu mengingatkan pengguna agar dapat mengembalikan inventaris laboratorium tepat waktu dan menghindari keterlambatan. Saat ini, fitur tersebut belum tersedia dalam sistem, namun dapat menjadi tambahan yang bermanfaat bagi pengguna.
- b. Untuk pengembangan aplikasi selanjutnya, diharapkan sistem ini dapat dibuat dalam versi Android. Dengan adanya aplikasi berbasis Android, pengguna dapat mengakses dan menggunakan sistem secara lebih mudah dan fleksibel melalui perangkat seluler, sehingga meningkatkan efisiensi dan kenyamanan dalam pengelolaan laboratorium.

DAFTAR PUSTAKA

- J. R. Naldo, P. A. Hernando, I. Artambo, and B. Pangaribuan, Pembuatan Website Admininstrator Peminjaman Ruangan Laboratorium Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jakarta. 2021.
- C. Cahyaningtyas, C. Gudiato, M. Sari, T. Informasi, I. Shanti Bhuana, and K. Barat, "Implementasi Metode Rapid Application Development (RAD) Pada Sistem Informasi Peminjaman Kunci Ruangan Berbasis Website (Studi Kasus : Institut Shanti Bhuana)," 2024.[Online]. Available: <https://jurnal.publikasitecno.id/index.php/jim>
- I. Guna Noviantama and A. W. Purno Wahyu, "Implementasi Contionous Integration Dan Continous Deployment Pada Aplikasi Learning Management System Di Pt. Millennia Solusi Informatika," 2021.
- R. Aditya, V. Handrianus Pranatawijaya, P. Bagus Adidyana Anugrah Putra, J. Hendrik Timang, K. Palangkaraya, and K. Tengah, "Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype," 2021.
- M. Nishom and D. S. Wibowo, "Sistem Informasi Laboratorium Berbasis Website Di Program Studi D Iv Teknik Informatika Politeknik Harapan Bersama Tegal," Gema Teknologi, vol. 21, no. 1, pp. 1–10, Oct. 2020, doi: 10.14710/gt.v21i1.28746.

- F. Putra Apriliyanto, Y. Prihati, and C. Asa Bakti, “Implementasi Ci/Cd Menggunakan Github Actions Untuk Pengembangan Web Landing Page Umkm Xyz Implementation Of Ci/Cd Using Github Actions For The Development Of A Web Landing Page For Umkm Xyz,” *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 7, no. 4, 2024.
- L. Veronica Wijaya and S. Resty Ramadhani, “Sistem Informasi Peminjaman Laboratorium pada Cross-Platform dengan Metode Prototyping (Studi Kasus: Politeknik Caltex Riau),” 2020. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- A. Fiotama Josyaf, E. Fatkhiyah, and J. Triyono, “Rancangan Prototype Sistem Informasi Peminjaman Laptop Berbasis Web Pada Laboratorium Komputer,” *Jurnal SCRIPT*, vol. 9, no. 2, 2021.
- W. Widiantara, G. P. Megayasa, L. Putu, and R. Noviana, “Rancang Bangun Sistem Continuous Integration (Ci) Dan Continuous Delivery (Cd) Menggunakan Jenkins Pada Aplikasi Berbasis Web,” *Jurnal Manajemen dan Teknologi Informasi (JMTI)*, vol. 14, pp. 36–39, doi: 10.59819.
- N. H. Marthiawati, K. Kurniawansyah, and H. A. Rohayani, “Sistem Jadwal Pemakaian Laboratorium Komputer Pada Universitas Muhammadiyah Jambi,” *Jurnal Informatika, Sistem Informasi dan Kehutanan*, vol. 1, no. 2, 2022.
- K. Infrastruktur Jaringan Komputer et al., “Implementasi Continuous Integration/Continuous Deployment Untuk Mengotomatisasi Manajemen.”
- I. Carolina and A. Rusman, “Universitas Bina Sarana Informatika,” *Jl. Kamal Raya*, vol. 4, no. 2, 2019.
- Y. Anggraini, D. Pasha, and A. Setiawan, “Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus : Orbit Station),” *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol.1, no. 2, pp.64 70,2020,[Online].Available:<http://jim.teknokrat.ac.id/index.php/JTSI>
- M. Saed Novendri et al., “Aplikasi Inventaris Barang Pada Mts Nurul Islam Dumai Menggunakan Php Dan Mysql.”
- M. Ronaldo and D. Pasha, “Sistem Informasi Pengelolaan Data Santri Pondok Pesantren An-Ahl Berbasis Website,” 2021.
- T. Bin Tahir, M. Rais, and M. A. Hs, “Aplikasi Point OF Sales Menggunakan Framework Laravel Point OF Sales Appilaction using Laravel Framework,” *Jurnal Informatika dan Komputer* p-ISSN, vol. 2, no. 2, pp. 2355–7699, 2019, doi: 10.33387/jiko.
- Aulia, Malyda “Implementasi Sistem Peminjaman Laboratorium Komputer Berbasis Website Menggunakan Metode Rapid Application Development (Rad)”. Undergraduate thesis, Politeknik Negeri Bengkalis 2023.