

# BAB I

## PENDAHULUAN

### I.1 Latar Belakang

Di era Revolusi Industri 4.0, kemampuan dalam membangun aplikasi web menjadi salah satu keterampilan utama yang perlu dimiliki oleh mahasiswa di bidang teknologi informasi. Menurut Rosa (2023), aplikasi web adalah aplikasi perangkat lunak yang dijalankan melalui *web browser* dengan memanfaatkan teknologi internet, terdiri dari *frontend* yang dijalankan di sisi klien dan *backend* yang dijalankan di *server*, serta biasanya terhubung dengan sistem basis data. Proses kerjanya melibatkan interaksi antara pengguna dengan web browser yang mengirimkan *request* ke *server*, kemudian *server* memproses permintaan tersebut, mengakses *database* jika diperlukan, dan mengembalikan hasil dalam bentuk tampilan web kepada pengguna. Contoh aplikasi web mencakup sistem informasi akademik, layanan administrasi digital, *e-commerce*, hingga *platform* media sosial (Hamzah & Seman, 2022).

Dengan peran aplikasi web yang semakin penting dalam berbagai sektor industri dan pemerintahan, pemahaman tentang pengembangannya menjadi sangat relevan untuk dikuasai (Hamzah & Seman, 2022). Oleh karena itu, Program Studi Sistem Informasi Universitas Telkom menghadirkan mata kuliah *Web Application Development* sebagai wadah bagi mahasiswa untuk memperoleh pengalaman langsung dalam merancang dan mengembangkan aplikasi web. Mata kuliah ini tidak hanya fokus pada penguasaan teknis, tetapi juga mendorong mahasiswa untuk berpikir sistematis, kolaboratif, dan mengikuti standar industri dalam pengembangan perangkat lunak.

Dalam pelaksanaannya, praktikum *Web Application Development* dirancang untuk memberikan pengalaman langsung kepada mahasiswa dalam membangun aplikasi web menggunakan berbagai teknologi, seperti PHP, MySQL, dan *framework* Laravel. Praktikum ini bertujuan tidak hanya untuk melatih kemampuan teknis mahasiswa, tetapi juga membiasakan mereka dengan alur kerja pengembangan perangkat lunak secara profesional, seperti penggunaan *version control* dan penerapan arsitektur aplikasi. Namun demikian, di balik tujuan yang

mulia tersebut, terdapat tantangan-tantangan teknis yang kerap menghambat proses pembelajaran mahasiswa, terutama yang masih berada di tahap awal pemahaman terhadap teknologi yang digunakan (Malan dkk., 2023).

Salah satu masalah yang seringnya ditemui adalah proses instalasi dan konfigurasi perangkat lunak yang diperlukan. Menurut Malan dkk. (2023), Banyak mahasiswa pemula yang sudah kesulitan dalam menulis kode "*Hello World*", kendala ini menjadi lebih besar ketika terjadi galat "*Command Not Found*". Saat ini, mahasiswa diharuskan untuk melakukan instalasi *Visual Studio Code* (VSCode), XAMPP, dan Composer pada laptop masing-masing sebelum memulai praktikum. Meskipun langkah ini terdengar sederhana, kenyataannya terdapat banyak masalah yang muncul selama proses tersebut.

Data yang didapatkan dari wawancara dengan 2 (dua) asisten praktikum menunjukkan bahwa terdapat banyak *error* yang terjadi sebelum dan saat berjalannya praktikum. Kedua asisten menyebutkan angka sekitar 60% mahasiswa pada keseluruhan kelas yang diampu memerlukan *troubleshooting* dari *error* yang terjadi ketika menjalankan *programming environment* dengan XAMPP dan 40% sisanya dapat melakukan *troubleshooting* sendiri atau tidak mengalami *error*. Lalu dari 60% mahasiswa yang mengalami masalah tersebut, sekitar 60% *error* terjadi pada MySQL dan 40% sisanya terjadi pada Apache *web server*. Asisten menduga penyebab utama dari masalah ini adalah bentrok *port* Apache dan MySQL dari XAMPP dengan perangkat lunak dari praktikum mata kuliah lain, khususnya praktikum yang terkait dengan sistem operasi. Tak hanya itu, asisten mendapati solusi dengan mengganti *port* yang diblok ke *port* yang lain tidak dapat diterapkan pada beberapa perangkat mahasiswa. Asisten memperkirakan bahwa hal ini terjadi karena banyaknya perbedaan perangkat keras, versi sistem operasi, dan perbedaan versi perangkat lunak yang diunduh dan dimiliki oleh mahasiswa. Lalu, solusi dengan mengganti XAMPP dengan Laragon tidak dapat mengatasi masalah ini secara keseluruhan karena masalah *port* yang bentrok masih terjadi pada perangkat lunak tersebut.

Modul 0 (nol) terkait instalasi dan konfigurasi yang seharusnya mudah justru menjadi beban tambahan bagi mahasiswa, karena mahasiswa harus melakukan konfigurasi teknis bahkan sebelum memulai menulis kode. Bantuan dalam

*troubleshooting* instalasi dan konfigurasi tersebut seringkali menyita waktu yang cukup lama, apalagi ketika mahasiswa menggunakan perangkat lunak (*software*), perangkat keras (*hardware*), dan sistem operasi (OS) yang berbeda-beda (Snowberger & You, 2024). Isu teknis dan proses pemecahan masalah yang perlu dilakukan oleh mahasiswa dan pengajar dapat menyita waktu yang berharga, meningkatkan tingkat kecemasan mahasiswa dalam pembelajaran selanjutnya, dan mengaburkan tujuan pembelajaran yang hendak dicapai (Fernald dkk., 2023). Selain tantangan teknis dalam instalasi dan konfigurasi, proses pengumpulan hasil praktikum juga menimbulkan masalah tersendiri. Saat ini, mahasiswa diharuskan untuk mengumpulkan *source code* dalam *repository* GitHub masing-masing, disertai dengan *screenshot* dari halaman web hasil pengerjaan. Namun, metode ini memiliki potensi masalah, seperti manipulasi *screenshot* yang tidak merefleksikan hasil sebenarnya dari pekerjaan mahasiswa. Selain itu, pengecekan hasil pengerjaan dengan cara menjalankan *source code* secara manual akan terlalu menghabiskan waktu.

Wawancara dengan kedua asisten praktikum menjelaskan bahwa proses penilaian praktikum memerlukan tahapan yang cukup kompleks. Asisten menjabarkan bahwa proses penilaian dimulai dari melihat performa pengerjaan mahasiswa saat berjalannya praktikum dan *screenshot* hasil *website* yang sudah di-*submit* setelah praktikum selesai. Apabila kedua aspek tersebut dirasa tidak bermasalah oleh asisten, maka penilaian hanya akan dilihat dari kedua aspek tersebut. Namun jika dirasa bermasalah, asisten akan mengecek algoritma pada *source code* yang ditulis oleh mahasiswa. Jika algoritma dirasa sudah benar, asisten cukup menilai sampai tahap tersebut dan jika algoritma dirasa tidak benar, asisten akan menyiapkan *programming environment* jika belum sesuai, melakukan *cloning* dari *repository* mahasiswa tersebut, dan menjalankan *source code* tersebut untuk melihat *output* web hasil pekerjaan mahasiswa. Proses ini dilakukan untuk mengurangi tahap yang dirasa tidak diperlukan ketika tidak menemukan aspek yang janggal. Namun, proses penilaian praktikum masih ditentukan oleh preferensi masing-masing asisten praktikum.

Asisten pernah menemukan beberapa hal yang janggal ketika melakukan penilaian. Ada pekerjaan yang hasil *output* pada *screenshot* yang di-*submit* sesuai

dengan *input* yang dimasukkan, namun ketika dicek, ternyata algoritma yang digunakan salah dan *input* selain yang dipakai pada *screenshot* menghasilkan *output* yang salah. Pada kasus yang lain, ada mahasiswa yang tidak mendapati *error* ketika awal pengerjaan praktikum dan baru mendapati *error* pada akhir praktikum. Namun, mahasiswa tersebut melakukan *submit screenshot* dari hasil *output* yang muncul pada awal pengerjaan praktikum. Kedua kejanggalan tersebut ditemukan dengan menggunakan proses penilaian yang sudah disebutkan sebelumnya. Walaupun demikian, asisten menyadari bahwa proses tersebut sepenuhnya bergantung pada kejelian asisten dalam menemukan kejanggalan dan preferensi asisten dalam melakukan proses penilaian. Asisten juga menambahkan bahwa menjalankan *source code* dari semua mahasiswa akan menghasilkan penilaian yang lebih akurat, namun hal tersebut akan membuang banyak waktu dan tenaga.

Borowski dkk. (2020) menyatakan bahwa ketika pengajar harus mengunduh *archive* dari banyak proyek milik mahasiswa untuk menjalankan semua *source code* tersebut satu per satu, maka proses penilaian tersebut tidak efisien. Perbedaan lingkungan *runtime* antara laptop mahasiswa dan pengajar juga sering kali menyebabkan *source code* yang berjalan lancar di laptop mahasiswa gagal dijalankan di perangkat pengajar sehingga menambah kompleksitas dalam proses penilaian.

Untuk mengatasi masalah tersebut, solusi yang diajukan dalam penelitian ini adalah implementasi *cloud development environment* (CDE) menggunakan GitHub Codespaces dan *platform-as-a-service* (PaaS) menggunakan Koyeb. CDE menyediakan lingkungan pengembangan berbasis *cloud* yang sudah dikonfigurasi dengan semua perangkat lunak yang diperlukan, seperti VSCode, PHP, Composer, MySQL, dan Node.js (jika *framework javascript* diperlukan kedepannya) (GitHub, 2024). Dengan solusi ini, mahasiswa dapat langsung memulai praktikum tanpa perlu melakukan instalasi atau konfigurasi tambahan, cukup dengan login ke GitHub dan menjalankan Codespaces (Snowberger & You, 2024). Selain itu, dengan dukungan akses melalui *web browser*, mahasiswa dapat menggunakan perangkat apa pun tanpa tergantung pada laptop pribadi (Malan dkk., 2023).

Sementara itu, PaaS memungkinkan mahasiswa untuk mendistribusikan hasil pekerjaan praktikum ke internet secara langsung dalam bentuk *website* sehingga asisten praktikum dapat melakukan penilaian tanpa perlu menjalankan *source code* milik mahasiswa satu per satu maupun melakukan *build* Codespace baru untuk setiap mahasiswa (Ghazaly, 2021). PaaS ini juga mendukung integrasi berkelanjutan (CI/CD), menyebabkan setiap perubahan kode yang di-*push* ke GitHub secara otomatis di-*deploy* ke *website* tersebut sehingga mengurangi beban mahasiswa dalam mengelola infrastruktur (Koyeb, 2024).

Melalui pendekatan metode studi kasus, penelitian berfokus pada implementasi dan analisis beberapa CDE dan PaaS, termasuk GitHub Codespaces dan Koyeb, dalam konteks praktikum *Web Application Development* di Program Studi Sistem Informasi Universitas Telkom. Peneliti mengevaluasi sejauh mana sistem rekomendasi yang diuji mampu mengatasi hambatan teknis pada proses pembelajaran. Diharapkan, pendekatan ini tidak hanya menyederhanakan proses teknis bagi mahasiswa dan pengajar, tetapi juga memungkinkan fokus pembelajaran beralih ke pemahaman konsep dan praktik pengembangan web yang lebih substansial.

Dengan latar belakang penggunaan Laravel sebagai *framework* dalam praktikum, serta berbagai tantangan teknis yang telah diuraikan, maka solusi berbasis *cloud* ini menjadi relevan untuk diimplementasikan. Permasalahan yang telah diidentifikasi, mulai dari kendala instalasi, keterbatasan perangkat, hingga kompleksitas dalam penilaian, semuanya bermuara pada kebutuhan akan lingkungan praktikum yang lebih inklusif, efisien, dan dapat diakses dari berbagai perangkat. Oleh karena itu, penelitian ini berfokus pada implementasi GitHub Codespaces dan Koyeb untuk menyederhanakan infrastruktur teknis pendukung praktikum pengembangan aplikasi web, khususnya dalam hal penyediaan lingkungan pemrograman dan *deployment* aplikasi untuk penilaian.

## **I.2 Perumusan Masalah**

Berdasarkan hasil penelaahan latar belakang dari penelitian ini, didapatkan serangkaian rumusan masalah sebagai berikut:

1. Bagaimana kondisi sistem *existing* pada praktikum *Web Application Development*?
2. Apa CDE dan PaaS yang paling sesuai untuk diimplementasikan pada praktikum ini?
3. Bagaimana perbandingan sistem rekomendasi yang dibuat dengan sistem *existing*?

### **I.3 Tujuan Penelitian**

Berdasarkan hasil perumusan masalah sebelumnya, didapatkan serangkaian tujuan penelitian sebagai berikut:

1. Mengetahui bagaimana kondisi sistem *existing* pada praktikum *Web Application Development*.
2. Mengetahui CDE dan PaaS yang paling cocok untuk digunakan dalam rangka mengatasi kendala instalasi, konfigurasi perangkat lunak, memfasilitasi proses *deployment*, dan penilaian hasil praktikum yang dihadapi praktikan dan asisten praktikum pada praktikum ini.
3. Menganalisis perbandingan antara CDE dan PaaS yang paling cocok dengan sistem *existing*.

### **I.4 Manfaat Penelitian**

Terdapat manfaat penelitian yang dapat dirangkai sebagai berikut:

1. Bagi Universitas Telkom dan program studi Sistem Informasi, penelitian ini memberikan manfaat berupa peningkatan efisiensi dan efektivitas proses praktikum *Web Application Development*, khususnya dalam hal pengelolaan lingkungan pengembangan dan penilaian hasil praktikum. Dengan penerapan CDE dan PaaS, praktikan dan asisten praktikum dapat lebih produktif, karena proses instalasi, konfigurasi, serta penilaian hasil dapat dilakukan secara lebih cepat dan akurat. Penelitian ini diharapkan mampu mendukung keberlanjutan digitalisasi dalam proses akademis di lingkungan universitas.
2. Bagi mahasiswa yang menjalani praktikum, penelitian ini memberikan manfaat berupa kemudahan dalam mengakses lingkungan pengembangan

yang homogen, tanpa perlu khawatir akan masalah instalasi dan konfigurasi perangkat lunak.

3. Bagi asisten praktikum, penelitian ini memudahkan proses penilaian, karena hasil pengerjaan dapat diakses dan diverifikasi melalui PaaS, tanpa harus mengunduh dan menjalankan *source code* secara manual.
4. Bagi peneliti lain, pendekatan yang diusulkan dalam penelitian ini dapat menjadi referensi atau dasar untuk penelitian lebih lanjut mengenai penerapan teknologi *cloud* dalam pembelajaran di institusi pendidikan.

### **I.5 Batasan Penelitian**

Terdapat batasan penelitian yang dapat dirangkai sebagai berikut:

1. CDE yang dibandingkan dalam penelitian ini meliputi GitHub Codespaces, Codesandbox, Jetify, dan DevZero. Objek CDE utama adalah GitHub Codespaces.
2. PaaS untuk *deployment* aplikasi web praktikan yang dibandingkan dalam penelitian ini meliputi Koyeb, Render, Northflank, dan Lade. Objek PaaS utama adalah Koyeb.
3. Solusi yang diberikan memenuhi beberapa kondisi, yaitu tidak membutuhkan biaya, cukup *powerful* untuk penggunaan praktikum, dan mudah digunakan oleh mahasiswa.
4. Penelitian ini tidak mencakup aspek lain dari proses pembelajaran praktikum tersebut.

### **I.6 Sistematika Laporan**

Laporan ini memiliki struktur penulisan yang dijelaskan pada subbab ini.

1. Bab I Pendahuluan

Bab I menjelaskan latar belakang, rumusan masalah, tujuan, manfaat, dan batasan penelitian. Bab ini berfokus pada tantangan praktikum *Web Application Development*, seperti instalasi *software* dan penilaian hasil, serta solusi berupa penerapan CDE dan PaaS.

2. Bab II Tinjauan Pustaka

Bab II menjelaskan penelitian terdahulu, metode, dan definisi konsep yang menjadi referensi dalam memahami bagaimana CDE dan PaaS dapat

berkontribusi dalam pengembangan aplikasi web, terutama dalam konteks akademik. Penelitian ini mengutip dan menjelaskan definisi konsep yang dipakai, seperti *cloud computing*, PaaS, Docker, dan CDE.

### 3. Bab III Metodologi Penelitian

Bab III menjelaskan metode dan sistematika penyelesaian masalah pada penelitian ini. Sistematika penyelesaian masalah tersebut dibagi menjadi 4 (empat) fase, yaitu *foundation phase*, *prefield phase*, *field phase*, dan *reporting phase*.

### 4. Bab IV Identifikasi dan Perancangan

Bab IV menjelaskan identifikasi dari sistem *existing* dan sistem rekomendasi, konfigurasi sistem rekomendasi, implementasi sistem, dan spesifikasi kebutuhan teknologi.

### 5. Bab V Pengujian, Evaluasi, dan Analisis

Bab V menjelaskan pengujian, evaluasi, dan analisis yang dilakukan pada CDE, PaaS, dan sistem yang dibandingkan.

### 6. Bab VI Kesimpulan dan Saran

Bab VI menjelaskan kesimpulan dari pengujian yang telah dilakukan dan saran yang ingin diberikan kepada peneliti selanjutnya. Kesimpulan menjelaskan apa CDE dan PaaS yang paling cocok dan bagaimana perbandingan sistem rekomendasi dengan sistem *existing*.