

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi informasi saat ini mengalami peningkatan termasuk dalam bidang pengembangan perangkat lunak. Pendekatan konvensional atau tradisional tidak cukup untuk memenuhi kebutuhan bisnis atau organisasi [1]. Masalahnya dalam pengembangan perangkat lunak konvensional, proses pembangunan (*build*) dan pengujian (*test*) yang dilakukan oleh tim pengembang dan tim operasi dilakukan secara terpisah [2]. Hal ini mengakibatkan waktu yang diperlukan dalam proses pengembangan perangkat lunak menjadi lebih lama [3] dan mengalami beberapa kegagalan termasuk proyek yang terlambat sebab jadwal proyek yang sudah ditetapkan sulit untuk mengubahnya, bahkan jika terjadi perubahan kebutuhan atau kendala selama proses pengembangan [4].

Pengembangan perangkat lunak menerapkan praktik *Continuous Integration / Continuous Deployment* pada organisasi, untuk sering merilis fitur karena pengembangan *website* menjadi semakin kompleks dengan tuntutan untuk merilis perubahan perangkat lunak secara cepat [5]. Dalam prosesnya, setiap kali ada perubahan akan otomatis diuji dan perubahan bisa langsung diterima di lingkungan produksi *production environment* [6]. Metode yang dikembangkan untuk mengatasi masalah dan mempercepat proses pengembangan adalah metode *Agile* yang berarti “bergerak cepat dan mudah” [7]. DevOps diperkenalkan untuk menyatukan antara tim pengembang dan tim operasi, berperan penting dalam proses *build*, *test*, dan *deployment* menggunakan metode *Agile* dilakukan secara otomatis [8].

Proses pengujian menggunakan CI/CD menunjukkan tingkat keberhasilan sebesar 80% dari lima kali percobaan, dengan satu percobaan mengalami kegagalan karena tidak lulus uji akibat adanya *bug* [3]. Hal ini menyebabkan *pipeline* terhenti dan memberikan notifikasi kepada *developer* untuk memperbaiki kode yang bermasalah.

Sebaliknya, dalam pengujian tradisional, tingkat keberhasilan hanya mencapai 60%, di mana dua dari lima percobaan gagal karena adanya *bug* [3]. Pengujian dengan CI/CD mendeteksi kesalahan atau *bug* dalam aplikasi secara langsung melalui *code review*, sementara pengujian tradisional baru mengidentifikasi kesalahan setelah seluruh proses pengujian selesai [3]. Temuan ini menunjukkan bahwa metode pengujian tradisional lebih rentan terhadap kesalahan atau *human error*.

Dalam upaya meningkatkan proses pengembangan perangkat lunak saat ini, organisasi atau perusahaan banyak menerapkan CI/CD [1]. Banyaknya alat CI/CD yang tersedia memberikan berbagai opsi bagi *developer* untuk memilih alat yang paling sesuai dengan kebutuhan proyeknya. [9]. Dampak implementasi CI/CD dalam manajemen proyek dapat dilihat melalui peningkatan efisiensi, kecepatan siklus pengembangan, meningkatkan kolaborasi antar tim, serta mengurangi kesalahan manual [10]. Menurut *survey* JetBrains pada tahun 2023, tentang pentingnya penggunaan alat CI/CD, *survey* ini diisi lebih dari 26.000 *developer* di seluruh dunia, menghasilkan 45% *developer* mengonfirmasi penggunaan alat CI/CD dalam pengembangan perangkat lunak dan alat CI/CD yang paling banyak digunakan yaitu Jenkins 54% dan GitHub Action 51% [11].

Berdasarkan survei yang dilakukan di Universitas Telkom kampus Purwokerto terhadap 108 mahasiswa/i, diketahui bahwa sebanyak 81 mahasiswa/i belum menerapkan CI/CD dalam pengembangan proyek *website*. Sebagian besar mahasiswa menyimpan kode program mereka di GitHub 94 orang, sementara lainnya menggunakan GitLab 2 orang dan menyimpan di perangkat pribadi 9 orang. Terkait frekuensi melakukan perubahan kode ke *repository* proyek, sebanyak 73 orang menyatakan melakukan *push* setiap kali pengerjaan fitur selesai, 20 orang hanya melakukan *push* ketika diperlukan (seperti *bug fixes* atau fitur besar), 10 orang tidak memiliki jadwal tertentu, 3 orang hanya melakukan *push* setelah semua kode selesai, 1 orang tergantung pada sistem yang dibuat, dan 1 orang belum pernah melakukan *push* sama sekali. Data ini menunjukkan bahwa masih banyak mahasiswa yang belum

memanfaatkan CI/CD secara optimal dalam pengembangan proyek mereka, meskipun penggunaan *repository* seperti GitHub cukup umum di kalangan mahasiswa.

Penggunaan alat CI/CD dalam pengembangan perangkat lunak berperan penting dalam pengujian kode secara otomatis. Hal ini membantu mendeteksi *bug* atau masalah lebih awal dalam siklus pengembangan. CI juga menggabungkan kode dari *developer* yang melakukan perubahan ke cabang utama untuk *deployment* [12]. CD berperan dalam menyiapkan perangkat lunak untuk siap dirilis, setelah proses CI selesai, secara otomatis kode program masuk ke lingkungan produksi [12]. Penerapan CD memastikan kode yang berada pada lingkungan produksi selalu versi terbaru [8].

Berdasarkan penjelasan di atas, penelitian ini akan berfokus pada menganalisis performa CI/CD dengan dua alat yaitu, Jenkins dan GitHub. Penerapan ini diikuti dengan pengembangan aplikasi berbasis *website* dengan gambaran proses secara otomatis *build*, *test*, dan *deploy* aplikasi ke *server*. Setelah itu, dibuat analisis berdasarkan waktu eksekusi [3], kendala, dan biaya . Maka dari itu penulis mengambil judul penelitian tugas akhir sebagai berikut “**Analisis Performa Jenkins GitHub Pada Proses *Continuous Integration* dan *Continuous Deployment* Pada Pengembangan *Website***”. Diharapkan dapat memberikan wawasan bagi praktisi, pengembang dan peneliti dalam memilih alat yang paling sesuai untuk kebutuhan proses CI/CD dalam pengembangan *website*.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang dijelaskan di atas, pengembangan *website* membutuhkan proses yang efisien dan efektif. Proses pengembangan yang dilakukan secara tradisional dapat mengakibatkan *human error* dan banyaknya alat CI/CD yang tersedia memunculkan tantangan dalam memilih alat yang paling tepat untuk digunakan. Oleh karena itu, analisis performa Jenkins dan GitHub menjadi penting untuk membantu memilih alat CI/CD yang tepat dan dapat meningkatkan waktu serta kualitas pengembangan *website*.

1.3. Pertanyaan Penelitian

Dengan merumuskan masalah di atas, maka diperoleh pertanyaan peneliti dalam melakukan penelitian ini yaitu:

1. Bagaimana performa Jenkins dan GitHub dalam menjalankan proses CI/CD pada penggunaan *server*?
2. Apakah terdapat perbedaan signifikan dalam hal waktu eksekusi, kendala yang dihadapi, dan biaya operasional antara penggunaan Jenkins dan GitHub dalam proses CI/CD?
3. Bagaimana implementasi proses CI/CD dengan Jenkins dan GitHub?

1.4. Batasan Masalah

Batasan masalah dalam penelitian ini ditetapkan untuk memastikan fokus yang jelas dan ruang lingkup yang terkontrol. Adapun batasan masalah dari penelitian ini adalah sebagai berikut:

1. Berfokus pada pengembangan aplikasi *website static* dan tidak meliputi jenis aplikasi lain seperti aplikasi *mobile* atau *desktop*.
2. Penggunaan alat CI/CD membatasi analisis performa hanya pada kedua alat CI/CD yang telah dipilih, yaitu Jenkins dan GitHub.
3. Berfokus pada sejumlah metrik yang telah ditetapkan, seperti waktu eksekusi, kendala, dan biaya.
4. Bahan yang digunakan dalam analisis performa dapat berasal studi kasus, atau pengalaman penulis sebagai *website developer*.
5. Sistem operasi yang digunakan ubuntu versi 24.04.

1.5. Tujuan Penelitian

Penelitian ini dilakukan untuk mencapai berbagai tujuan yang berfokus pada analisis penggunaan alat *Continuous Integration* dan *Continuous Deployment (CI/CD)*

dalam pengembangan *website*. Beberapa tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengetahui performa penggunaan alat CI/CD dalam pengembangan *website*.
2. Menganalisis perbedaan dalam waktu eksekusi, kendala, dan biaya antara Jenkins dan GitHub.
3. Mengetahui implementasi penggunaan alat CI/CD dalam pengembangan *website*.

1.6. Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi yang bermanfaat baik secara teoretis maupun praktis, khususnya dalam konteks implementasi CI/CD pada pengembangan *website*. Beberapa manfaat penelitian yaitu:

1. Penelitian ini akan memberikan pemahaman tentang performa kedua alat CI/CD seperti Jenkins dan GitHub, dalam pengembangan *website*.
2. Hasil penelitian ini akan membantu *developer* dalam memilih alat CI/CD yang optimal sesuai dengan kebutuhan dan preferensi dalam pengembangan *website*.
3. Penelitian ini akan memberikan pengetahuan praktis bagi praktisi, *developer* dan peneliti dalam memilih dan mengimplementasikan alat CI/CD yang ke dalam proyeknya.