

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam era digital yang terus berkembang, tuntutan terhadap efisiensi dan produktivitas dalam operasional bisnis semakin tinggi. Salah satu aspek penting namun sering terabaikan adalah pengelolaan ruang pertemuan, terutama di perusahaan besar yang memiliki banyak cabang dan lokasi. Pengelolaan ruang pertemuan yang tidak optimal dapat menimbulkan berbagai masalah, seperti pemborosan waktu, jadwal yang tumpang tindih, serta hilangnya peluang bisnis yang seharusnya dapat dimanfaatkan [1].

Untuk menjawab tantangan ini, PT. Indo Trans Teknologi Bandung mengembangkan aplikasi INMEET, sebuah sistem informasi ruang pertemuan berbasis digital. INMEET dirancang untuk mengoptimalkan penggunaan ruang pertemuan dengan menyediakan informasi real-time mengenai ketersediaan ruang, jadwal pemesanan, serta fasilitas yang tersedia di masing-masing ruang. Aplikasi ini juga terintegrasi dengan berbagai sistem kalender, email, dan aplikasi produktivitas lainnya, sehingga memudahkan pengguna dalam mengelola dan mengatur jadwal pertemuan.

Namun, seiring dengan berkembangnya teknologi dan meningkatnya kebutuhan bisnis, pengembangan aplikasi seperti INMEET membutuhkan pendekatan yang lebih efisien dan adaptif. Salah satu solusi strategis adalah dengan mengimplementasikan praktik *Continuous Integration* (CI) dan *Continuous Deployment* (CD), yang memungkinkan tim pengembang untuk melakukan proses *build*, *testing*, dan *deployment* secara otomatis dan berkelanjutan [2]. Praktik ini telah terbukti mampu mempercepat siklus pengembangan perangkat lunak dan meningkatkan kualitas rilis melalui pengujian otomatis [3].

Lebih lanjut, penerapan CI/CD juga memungkinkan tim pengembang untuk menerapkan *pipeline* otomatis yang mendukung rollback, validasi keamanan, dan integrasi langsung ke lingkungan produksi [4]. Studi terbaru menunjukkan bahwa pendekatan CI/CD secara signifikan mengurangi risiko kegagalan sistem di fase *deployment* dan meningkatkan frekuensi rilis aplikasi.

Dengan demikian, otomatisasi pengembangan dan penerapan aplikasi INMEET melalui CI/CD di PT. Indo Trans Teknologi Bandung menjadi sangat penting untuk memastikan bahwa sistem ini dapat terus memenuhi kebutuhan bisnis yang dinamis dan mendukung kelancaran operasional perusahaan [5].

1.2 Rumusan Masalah dan Solusi

Berdasarkan latar belakang yang telah dijelaskan, beberapa masalah dapat diidentifikasi sebagai berikut:

1. Bagaimana cara mengimplementasikan *Continuous Integration* (CI) pada aplikasi INMEET untuk memastikan setiap perubahan kode dapat diuji dan diintegrasikan secara otomatis ?
2. Bagaimana cara mengimplementasikan *Continuous Deployment* (CD) pada aplikasi INMEET untuk memastikan setiap pembaruan aplikasi dapat di-*deploy* secara otomatis dan aman ke lingkungan produksi?
3. Bagaimana cara memastikan bahwa penerapan CI/CD dapat meningkatkan kualitas kode dan mengurangi risiko kesalahan dalam proses pengembangan aplikasi INMEET?
4. Bagaimana cara mengelola dan mengoptimalkan *pipeline* CI/CD untuk mendukung siklus pengembangan yang cepat dan efisien pada aplikasi INMEET?

Untuk menjawab permasalahan di atas, berikut adalah solusi yang ditawarkan:

1. Implementasikan *Continuous Integration* (CI) aplikasi INMEET

Implementasi CI pada aplikasi INMEET dapat dilakukan dengan menggunakan alat GitLab CI. Setiap perubahan kode yang di *push* ke *repository* akan memicu *pipeline* CI yang secara otomatis menjalankan serangkaian tes untuk memastikan kode tersebut dapat diintegrasikan tanpa masalah.

2. Implementasikan *Continuous Deployment* (CD) aplikasi INMEET

Implementasi CD dapat dilakukan dengan memperluas *pipeline* CI untuk mencakup langkah- langkah deployment. Setelah kode berhasil melewati semua tes, *pipeline* CD akan secara otomatis melakukan deployment ke lingkungan produksi menggunakan alat Docker, dengan memastikan bahwa proses deployment aman dan tidak mengganggu layanan yang berjalan

3. Penerapan CI/CD untuk meningkatkan kualitas koden dan mengurangi risiko kesalahan

Penerapan CI/CD meningkatkan kualitas kode dengan memastikan bahwa setiap perubahan kode diuji secara otomatis sebelum diintegrasikan dan di-*deploy*. Ini mengurangi risiko kesalahan karena setiap bug atau masalah dapat terdeteksi lebih awal dalam siklus pengembangan. Selain itu, *code review* dan static code analysis dapat diintegrasikan ke dalam *pipeline* CI/CD untuk memastikan standar kualitas kode terpenuhi.

4. Cara mengelola dan mengoptimalkan *pipeline* CI/CD

Pipeline CI/CD dapat dikelola dan diopmalkan dengan memecah *pipeline* menjadi beberapa tahap yang lebih kecil dan paralel untuk mempercepat proses *build* dan testing. Selain itu, caching dan penggunaan container dapat membantu

mempercepat waktu *build*. *Monitoring* dan analisis *pipeline* juga penting untuk mengidentifikasi bottleneck dan area yang dapat dioptimalkan lebih lanjut.

1.3 Tujuan

Alasan utama pekerjaan ini dilakukan adalah untuk mengatasi berbagai masalah yang dihadapi aplikasi INMEET dalam pengelolaan layanan dengan memanfaatkan teknologi CI/CD. Berikut adalah rincian tujuan yang ingin dicapai:

1. Meningkatkan Kecepatan Pengembangan

Mempercepat siklus pengembangan perangkat lunak dengan otomatisasi proses *build*, *testing*, dan *deployment*, sehingga fitur-fitur baru dapat dirilis lebih cepat.

2. Meningkatkan Efisiensi Tim Pengembang

Mengurangi beban kerja manual pada tim pengembang dengan otomatisasi tugas-tugas rutin, sehingga mereka dapat fokus pada pengembangan fitur dan perbaikan yang lebih penting.

3. Mempercepat Respons terhadap perubahan

Memungkinkan perusahaan untuk lebih cepat merespons perubahan kebutuhan bisnis dan permintaan pelanggan dengan siklus pengembangan dan *deployment* yang lebih cepat.

4. Meningkatkan Kolaborasi Tim

Meningkatkan kolaborasi antar anggota tim pengembang dengan *pipeline* CI/CD yang transparan dan terintegrasi, sehingga semua anggota tim dapat melihat status *build* dan *deployment* secara real-time.

1.4 Batasan Masalah

Ruang lingkup kerja dalam proyek ini mencakup hal-hal berikut:

1. Otomatisasi Pembangunan dan Penerbitan

Proses pembangunan dan penerbitan aplikasi web akan dilakukan secara otomatis menggunakan GitLab CI/CD.

2. Penggunaan Docker

Aplikasi web ini akan dijalankan menggunakan teknologi Docker sebagai dasar operasionalnya

1.5 Penjadwalan Kerja

No	Deskripsi Kerja	Agustus		September				Oktober				November				December			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Perkenalan																		
2	Pemberian materi Docker dan CI/CD																		
3	Memperajari dan mempraktekan materi Docker dan CI/CD																		
4	Setup bahan-bahan untuk keperluan <i>deployment</i>																		
5	Melakukan proses CI/CD pada <i>repository</i> Gitlab																		
6	Setup Monitoring Elastic di <i>server</i>																		

Tabel 1. 1 Tabel Pelaksanaan Kerja