

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam siklus pengembangan aplikasi, tahap terakhir yang dilakukan agar aplikasi rekening nasabah dapat digunakan oleh user adalah *software deployment*. *Software deployment* merupakan salah satu tolak ukur pengukuran *software delivery and operational (SDO) performance*. SDO performance adalah alat ukur yang diteliti oleh DORA (*Digital Operational Resilience Act*) untuk memberikan tingkat ukur terhadap efektifitas praktek *deployment* dan *delivery*. Pada proses pengembangan dilakukan dengan prosedur manual, dimulai dengan proses perancangan (*development*), pengujian (*testing*), dan penerapan (*deployment*) [1]. Dalam pengembangan aplikasi rekening nasabah, proses *continuous integration* (CI) merupakan proses penggabungan atau penambahan fitur dalam code ke *repository* utama. Sedangkan proses *continuous deployment* (CD) adalah proses yang dilakukan secara otomatis dijalankan setelah proses *build* dan *test* berhasil dalam CI. Proses tersebut akan melakukan perilisan dan pendistribusian otomatis dari hasil *build* kedalam lingkungan *staging* atau *Production* [2].

Keuntungan yang didapat dalam mengadaptasi CI/CD *pipeline* adalah proses perilisan dan pendistribusian aplikasi rekening nasabah menjadi lebih konsisten. Selain itu keuntungan lain adalah dari segi waktu pengembang dapat di alih fungsikan tanpa harus melakukan proses perilisan secara manual. Dengan menghemat waktu maka sumber daya manusia yang sebelumnya terpakai untuk proses manual tersebut dapat di alih fungsikan untuk kebutuhan yang lebih penting, dengan mengadaptasi CI/CD cenderung memiliki proses kerja yang berkualitas, memiliki resiko rendah dan hemat waktu dan biaya. Penghematan yang di dapat dari segi waktu adalah, waktu *developer* dapat di alihkan ke tugas yang lebih penting. Penghematan biaya didapatkan dari peningkatan efektifitas waktu *developer*.

Dengan adanya masalah yang timbul tersebut dengan menerapkan CI/CD. Maka akan dibuat *system* integrasi otomatis (CI) dan deployment otomatis (CD) untuk menghemat waktu, biaya dan resiko dalam proses pengembangan selanjutnya [3].

Continuous Integration (CI) merupakan proses otomatisasi untuk pengembangan, dari mulai penulisan kode yang secara teratur dan digabungkan dalam repositori bersama, serta dilakukannya pengujian untuk masuk ke tahap selanjutnya [4].

Continuous Delivery / Continuous Deployment (CD) merupakan proses yang saling berkaitan pada pengembangan aplikasi rekening nasabah. Pada proses ini kode yang telah di tulis di unggah ke repositori dan dilakukan pengujian akan masuk ke tahapan ini untuk dilakukannya proses penerapan (*deployment*) dengan tujuan agar memastikan usaha *men-deploy* kode baru seminimal mungkin dan dapat melakukan upaya pelepasan (*release*) secara otomatis tiap perubahan yang dilakukan oleh pengembang terhadap aplikasi rekening nasabah dari repositori hingga ke tahapan *Product ion* di tiap *environment* yang nantinya akan sampai ke pengguna [5].

Kedua proses tersebut berisi tentang proses otomatisasi yang nantinya akan membentuk suatu *pipeline / cycle* CI/CD berkelanjutan satu sama lain dan mempercepat proses dari tahap perancangan (*development*), pengujian (*testing*), hingga tahap penerapan (*deployment*) serta memberikan efektivitas dan efesiensi waktu pada pengembang. Dengan adanya impelmentasi proses ini mengurangi konflik antar tim serta memberikan kemudahan dan diharapkan kinerja antar tim akan meningkat.

Berdasarkan latar belakang di atas, pengembangan dan distribusi aplikasi rekening nasabah sangat membutuhkan konsistensi, agar proses tersebut dapat menghemat waktu pada tahap pengembangan. Maka penulis mencoba memanfaatkan proses CI/CD dengan mengimplementasikan proses tersebut untuk pengembangan dan distribusi aplikasi rekening nasabah. Oleh karena itu penulis mengambil judul **“Implementasi Metode CI/CD (*Continuous Integration/Continuous Deployment*) dengan menggunakan docker container”**

1.2 Identifikasi Masalah

Berdasarkan latar belakang penelitian di atas dapat disimpulkan identifikasi masalahnya sebagai berikut :

1. Penelitian tidak membahas proses pengembangan aplikasi melainkan hanya proses pengembangan dan implementasi CI/CD.
2. Sering terhambatnya pengembangan dan distribusi aplikasi perhitungan rekening nasabah dikarenakan perubahan dan integrasi *environment*.
3. Banyak proses yang masih dilakukan secara manual dan masih ketergantungan terhadap *human input*.

1.3 Rumusan Masalah

Berdasarkan masalah yang sedang dihadapi, maka rumusan masalah yang dapat saya ambil adalah “Bagaimana menerapkan implementasi CI/CD (Continuous Integration/Continuous Deployment) dengan metode *scrum/agile* untuk memberikan dampak positif yang sifatnya konsisten untuk pengembangan dan distribusi aplikasi rekening nasabah”

1.4 Batasan Masalah

Sesuai dengan masalah yang dihadapi, maka batasan masalahnya sebagai berikut :

1. Bagaimana implementasi proses pengembangan aplikasi rekening nasabah yang terdiri dari proses integrasi *source code*, *testing*, dan *deployment* aplikasi secara manual menjadi otomatis.
2. Bagaimana dampak otomasi yang diperoleh yang ditinjau dari segi waktu, biaya dan resiko.
3. Implementasi proses CI/CD *pipeline* ini ditujukan untuk mengurangi kesalahan yang tidak diperlukan.

1.5 Tujuan dan Manfaat Penelitian

Adapun tujuan dan manfaat dari penelitian ini adalah sebagai berikut :

1.5.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Memberikan rancangan perubahan proses pengembangan aplikasi rekening nasabah dengan menggunakan CI/CD.
2. Membantu proses yang sebelumnya berjalan secara manual menjadi proses otomatis.
3. Mengurangi adanya konflik pada *environment* yang terdiri dari proses *source code*, *testing* dan *deployment*.
4. Memberikan memberikan dampak analisi yang diperoleh ditinjau dari segi waktu, biaya dan resiko.

1.5.2 Manfaat

Adapun manfaat dari penulisan skripsi ini adalah :

1. Penulis dapat keterampilan dalam merancang sebuah sarana dalam mengoptimisasi pada proses pengembangan dan distribusi aplikasi rekening nasabah.
2. Meminimalisir *human error* karna semua sudah di jalankan secara otomatis.
3. Menselaraskan tujuan dan memberikan siklus kerja yang baik antar tim.
4. Membantu meringankan waktu kerja bagi *developer* jika ada *bug fixing*, perubahan yg sifatnya *minor* maupun *major*.

1.6 Sistematika Penulisan

Sistematika penulisan merupakan uraian tentang susunan penulisan itu sendiri yang dibuat secara teratur dan terperinci sehingga dapat memberikan gambaran secara menyeluruh. Adapun sistematika penulisan pada skripsi ini terbagi menjadi lima bab, yaitu sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas tentang latar belakang, identifikasi masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian dan sistematika penulisan.

BAB II LANDASAN TEORI

Landasan Teori mencakup tinjauan pustaka penelitian terdahulu yang berkaitan dengan masalah yang dibahas dan dasar teori yang berhubungan dengan judul.

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang mengenai obyek penelitian, kerangka penelitian, analisis sistem berjalan, analisis usulan sistem, analisis kebutuhan.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang langkah-langkah penerapan sistem secara otomatisasi, *input* dan *output*, serta hasil penerapan sistem secara otomatisasi.

BAB V PENUTUP

Bab ini menjelaskan tentang kesimpulan dan saran yang diusulkan untuk proses pengembangan lebih lanjut agar tercapai hasil yang lebih baik.