1. Pendahuluan

1.1 Latar Belakang

Setiap perbankan memiliki sistem *core banking* yang merupakan sebuah aplikasi inti dan kunci dari sebuah perbankan. Sistem tersebut berfungsi untuk mengelola aktivitas-aktivitas inti perbankan yang meliputi *transaction*, *saving*, *loan*, dan *customer information file* [1]. Aktivitas inti tersebut erat hubungannya dengan nasabah bank yang secara tidak langsung merupakan user yang mengakses *core banking* tersebut. Terdapat momen di setiap tahun dalam satu hari di mana jumlah pengguna yang mengakses *core banking* secara bersamaan meningkat berkali-kali lipat. Pada PT. XYZ sedang melakukan pengembangan *mobile banking* versi terbaru untuk menggantikan *mobile banking* versi lama mereka. Pengembangan *mobile banking* baru tersebut juga diharapkan dapat menangani kenaikan jumlah nasabah setiap tahunnya pada PT. XYZ. Akan menjadi hal yang berbahaya jika sistem tersebut menjadi *down* karena akan dapat mengurangi tingkat kepercayaan nasabah terhadap bank nya dan dapat mengakibatkan penurunan jumlah nasabah pada bank tersebut. Hal tersebut mengharuskan sistem *core banking* tetap berjalan dengan baik di saat itu.

Untuk memastikan sistem *core banking* tersebut dapat berjalan dengan baik maka perlu dilakukan pengujian performa (*performance testing*). *Performance testing* merupakan pengujian yang berfungsi untuk mengetahui seberapa jauh performa dari sebuah aplikasi ketika bejalan di bawah beban kerja yang diharapkan [2]. Hasil *performance testing* dapat dikatakan baik apabila dapat mencapai ekspektasi dari pengguna aplikasi atau sistem yang diuji. Tidak ada patokan yang pasti untuk menentukan hasil yang baik pada *performance test*. Setiap aplikasi memiliki karakteristiknya masing-masing dan baik tidak hasilnya bergantung pada kebutuhan bisnis yang ada pada aplikasi tersebut.

Pada *performance testing* dapat dilakukan dengan bermacam metode pengujian, diantara yaitu *stress testing*, *load testing*, *endurance testing*, *spike testing*, dan lain sebagainya. Adapun metode yang akan digunakan untuk melakukan *performance testing* yaitu *stress testing* dan *load testing*. Metode tersebut dipilih karena pada *load testing* dapat digunakan untuk mengetahui apakah sistem *core banking* PT. XYZ dapat menangani trafik sesuai dengan ekspektasi untuk menghadapi momen kenaikan nasabah setiap tahunnya. Sedangkan dari hasil metode *stress testing* akan dapat membantu menemukan jumlah maksimum pengguna berjalan bersamaan (*concurrent user*) dan juga untuk menghindari terjadinya kegagalan sistem ketika didorong sampai kondisi puncaknya.

Dalam menjalankan pengujian tersebut perlu adanya *performance testing tools* salah satunya adalah Loadrunner. Loadrunner merupakan *performance testing tools* yang dapat mendukung *stress testing* maupun *load testing* dan menyertakan fungsionalitas untuk memudahkan dalam menganalisis hasil dari pengujian yang sudah dijalankan. Di Loadrunner beban kerja user akan ditentukan oleh *virtual user* yaitu *concurrent user* yang akan mengirimkan *request* ke aplikasi yang akan dilakukan pengujian [3].

Performance test pada *core banking* PT. XYZ sudah pernah dilakukan sebelumnya di *environment* dengan spesifikasi dan arsitektur server yang berbeda dengan *environment production*. *Environment production* merupakan versi sistem *core banking* yang saat ini digunakan untuk keperluan aktivitas nasabah yang ada di PT. XYZ. Sehingga PT. XYZ menyediakan *environment core banking* baru yang disebut dengan *environment QA* untuk keperluan *performance testing* ini dengan spesifikasi dan arsitektur server yang lebih menyerupai dengan versi *production*. Karena jika dilakukan *performance test* di *environment* yang sama lagi maka yang terjadi adalah hasil dari performa aplikasi tidak akan menunjukkan hasil yang mencerminkan versi *production*. Hal tersebut akan membuat aplikasi tersebut belum diketahui hasil dari performanya dan tidak terlacak jika ada sumber permasalahan dalam performanya. Dengan spesifikasi dan arsitektur yang menyerupai versi *production*, maka hasil yang akan didapatkan juga akan menggambarkan performa *core banking* yang saat ini sedang digunakan untuk mengelola aktivitas nasabah pada bank tersebut, karena *performance test* yang dilakukan sebelumnya pada *core banking* tersebut memiliki kondisi spesifikasi dan arsitektur yang jauh berbeda dengan versi *production*.

Maka dari itu penelitian ini akan melakukan *performance testing* dengan menggunakan metode *stress testing* dan *load testing* pada sistem *core banking* sebagai *tools* yang akan dipakai.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan maka didapatkan rumusan masalah sebagai berikut.

1. Bagaimana cara penerapan *performance testing* menggunakan metode *load testing* dan *stress testing* pada sistem *core banking* PT. XYZ?
2. Bagaimana pencapaian hasil *performance testing* terhadap objektif yang telah ditentukan?

1.3 Batasan Masalah

Adapun ruang lingkup batasan masalah pada penelitian ini yaitu sebagai berikut.

1. *Performance testing* yang akan dilakukan menggunakan metode *load testing* dan *stress testing*.
2. *Tools* yang akan digunakan untuk pengujian ini adalah Loadrunner.
3. Parameter yang akan dinilai dalam pengujian ini yaitu *response time*, *transactions per second* (TPS), dan *error rate*.

1.4 Tujuan

Berdasarkan rumusan masalah yang ada di atas, maka tujuan dari penelitian ini sebagai berikut.

1. Melakukan *load testing* dan *stress testing* pada *core banking* menggunakan *tools* Loadrunner.
2. Menganalisis hasil *performance testing* yang sudah dilakukan.