

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Perkembangan dalam komputasi dan teknologi nirkabel yang pesat menjadikan semakin banyak perangkat yang terhubung ke jaringan internet. *International Data Corporation* memprediksi objek sensor yang terhubung ke internet akan meningkat hingga 30 milyar pada tahun 2020 dan perangkat yang terhubung ke internet akan bertambah hingga 50 milyar di Amerika Serikat dan 231 milyar perangkat yang dapat digunakan seperti misalnya jam tangan pintar. Semua data dari perangkat dan sensor tersebut akan diproses menggunakan *cloud computing*. Perusahaan besar seperti Google dan Amazon sudah membuat *cloud server* di berbagai tempat untuk menunjang kebutuhan komputasi [1].

Dengan banyaknya data harus diolah pada cloud maka perusahaan penyedia jasa komputasi harus membangun infrastruktur untuk memenuhi kebutuhan. Menurut [2] beberapa perangkat *Internet of Things* (IoT) dirasa kurang cocok untuk menggunakan cloud. Untuk mengatasi masalah tersebut dapat diatasi dengan menambahkan komputasi fog di antara pengguna dan cloud. Keuntungan yang akan didapatkan ketika menambah fog pada jaringan cloud adalah menyediakan *bandwidth* yang besar, mengurangi *latency*, menyediakan cara yang efisien untuk *offloading* data yang akan dikirim ke *core network* [3]. Fog memiliki kemampuan komputasi yang lebih rendah dibandingkan dengan cloud tetapi berada dekat dengan pengguna sehingga akan menghasilkan *delay* yang rendah. Kedepannya, Fog akan terdapat di lokasi vital seperti stasiun, *base transceiver station* (BTS), kafe atau bahkan di rumah pengguna dan wilayah perkantoran. Koneksi antara pengguna dan fog menggunakan akses nirkabel dan hanya menggunakan komunikasi *single hop*.

Berdasarkan jurnal [4] fog dan cloud memiliki layanan yang sama misalnya untuk eksekusi aplikasi dari jarak jauh atau sebagai tempat penyimpanan, tetapi fog lebih dekat dengan pengguna. Karena itu fog dapat memberikan respon yang cepat

untuk aplikasi dan pengguna *mobile*. Fog akan berisi sebuah layanan yang dijalankan pada *container*. Fog memiliki jangkauan wilayah yang terbatas sehingga saat pengguna pindah dari satu tempat ke tempat lain yang berada di luar jangkauan, maka layanan fog juga akan berpindah ke fog yang lain.

Saat pengguna sudah terhubung dengan fog yang baru maka fog tersebut harus melanjutkan layanan yang sudah diberikan oleh fog sebelumnya. Hal inilah yang disebut sebagai *service-handover*. Pada dasarnya *handover* yang dilakukan di fog sama seperti yang dilakukan pada jaringan seluler, caranya adalah dengan menghentikan sementara *container* dan memindahkannya ke fog yang baru. *Handover* yang dilakukan harus memiliki *delay* yang kecil untuk memberikan pengalaman pengguna yang baik. Jurnal [5] menjelaskan *service-handover* pada fog akan dilakukan pada dua kasus. Pertama, mendukung suatu kejadian yang membutuhkan perpindahan layanan dari satu fog ke fog yang lain untuk mendukung mobilitas pengguna. Kasus kedua adalah memungkinkan manajemen sumber daya yang dinamis dari sebuah layanan fog computing.

Pada jurnal [6] menjelaskan bahwa Docker dapat menjadi standar baru karena memberikan peningkatan kinerja dan mengurangi waktu mulai aplikasi yang berjalan dalam *container*. Docker dapat mengurangi *overheads* pada saat menyebarkan aplikasi di fog dan memberikan efisiensi pada perangkat IoT. Oleh karena itu *container* akan lebih menguntungkan dibandingkan dengan *virtual machine*. Berdasarkan penelitian [7] Docker memiliki kemampuan untuk melakukan *checkpoint* yaitu menghentikan sementara *container* yang sedang berjalan dan menyimpan semua dalam sebuah file dan melakukan *restore* yaitu melanjutkan *container* yang sebelumnya dihentikan dan melanjutkan semua sesi yang ada. Proses tersebut biasa disebut sebagai *checkpoint and restore* (CRIU)

Maka dari itu, penulis akan membuat rancangan fog yang akan menjalankan *container* menggunakan Docker dan melakukan *handover* menggunakan proses *checkpoint-restore* dan *snapshot*. Sistemnya menggunakan *hard handover* yaitu membuat aplikasi mobile berbasis Android yang akan memberikan perintah untuk melakukan *handover*.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang, maka rumusan masalah yang didapat yaitu diantaranya:

1. Perancangan sistem handover menggunakan Docker *container*.
2. Perancangan aplikasi Android untuk memberikan perintah *handover*.
3. *Delay* yang dihasilkan pada sistem yang dirancang.

## 1.3. Tujuan dan Manfaat

Berdasarkan latar belakang yang diteliti, tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Menghasilkan sistem *handover* pada Docker *container* menggunakan CRIU dan aplikasi Android sebagai media bagi pengguna untuk berpindah ke fog yang lain.
2. Mengetahui *delay* yang dihasilkan pada sistem yang dirancang.

Manfaat dari penelitian Tugas Akhir ini adalah untuk membuat sistem *hard handover* supaya data yang dihasilkan dari satu fog dapat dipindahkan ke fog yang lain melalui jaringan lokal.

## 1.4. Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Tidak membahas keamanan jaringan.
2. Menggunakan Docker *container*.
3. Pengujian dilakukan tanpa background traffic.
4. Pengujian dilakukan pada *local area network* (LAN)
5. Parameter yang diuji hanya *delay*.
6. Fog diasumsikan sudah melakukan autentikasi.

## 1.5. Metode Penelitian

Metode penelitian yang digunakan dalam penyusunan Tugas Akhir ini sebagai berikut:

1. Studi Literatur

Mencari teori dan konsep yang berhubungan dengan *fog computing*, *handover*. Sumber yang dijarikan referensi adalah paper, jurnal, artikel dan tugas akhir.

## 2. Perancangan Sistem

Pada tahap ini melakukan perancangan berupa aplikasi untuk melakukan *handover* pada fog dan aplikasi untuk pengguna berbasis Android.

## 3. Pengujian

Melakukan pengujian terhadap sistem yang sudah dirancang.

## 4. Analisis

Pada tahap terakhir melakukan analisis dari hasil pengujian yang telah dilakukan dan memberikan kesimpulan dari analisis tersebut.