

## BAB 7 PENUTUP

Bab penutup menjelaskan kesimpulan sesuai dengan proses kebutuhan, perancangan, implementasi dan pengujian yang telah dilakukan. Bab ini juga memberikan saran untuk pengembangan selanjutnya terkait penelitian yang berhubungan.

### 7.1 Kesimpulan

Kesimpulan dari penelitian dijelaskan sebagai berikut:

1. MQTT broker dapat mengimplementasikan metode *failover*. MQTT broker menggunakan *broker cluster* yang terdiri dari *master* dan *slave broker* untuk mengimplementasikan *failover*. Broker cluster diimplementasikan menggunakan fitur dari *message broker ActiveMQ*. *Master* dan *slave broker* menggunakan *Sharing File System*, *kahaDB* dan *Network File System (NFS)* agar kedua *broker* saling terhubung dan membentuk *broker cluster*.
2. Implementasi metode *failover* dengan *ActiveMQ* dapat memperbaiki *availability* dari MQTT *broker* dengan adanya *broker cluster* yang terdiri dari *master* dan *slave broker*. *Slave broker* dapat menggantikan peran *master broker* ketika *master broker* gagal sehingga meminimalisir *downtime* yang terjadi. Kesimpulan hasil analisis dijelaskan sebagai berikut:
  - Nilai rata-rata *downtime* yang didapatkan sebesar 24.3266 detik dari lima percobaan. Waktu tersebut adalah waktu yang dibutuhkan untuk *slave broker* mendeteksi *master broker* telah terhenti dan menjalankan *ActiveMQ*. *Downtime* yang terjadi dipengaruhi *cost* dalam hal *processing delay* dan *overhead ActiveMQ* terkait penggunaan CPU dan *memory*.
  - Nilai *latency* sistem memiliki hasil lebih besar bila dibandingkan dengan sistem MQTT pada umumnya yang menggunakan *Mosquitto*. Hasil pengujian *latency* sistem yang didapatkan adalah 0.37 s, 0.8253 s, 1.4087 s, 1.5114 s, 1.7662 s pada sisi MQTT *publisher* dan 0.048 s, 0.0926 s, 0.1189 s, 0.1450 s, 0.1739 s pada sisi MQTT *subscriber*. Hasil pengujian *latency* dipengaruhi faktor penggunaan *thread* yaitu 20, 40, 60, 80 dan 100. Hasil juga dipengaruhi oleh *overhead* penggunaan CPU dan *memory* yang lebih besar karena mengakibatkan penurunan kecepatan ketika melakukan komunikasi.
  - Nilai *packet loss* ketika terjadi kegagalan pada MQTT *broker* lebih sedikit bila dibandingkan dengan *Mosquitto*. *ActiveMQ* memiliki hasil 20 *packet loss* dalam komunikasi 50 detik dan terjadi kegagalan pada MQTT *broker*. *Packet loss* yang terjadi menggambarkan waktu *downtime* sistem ketika terjadi kegagalan. Sistem dapat meminimalisir terjadinya *packet loss* dengan adanya *slave broker* yang meneruskan proses komunikasi ketika terjadi kegagalan pada *master broker*. Sementara pada sistem dengan broker tunggal seperti *Mosquitto*, proses komunikasi harus terhenti

karena harus menunggu *broker* diperbaiki sehingga terjadi *packet loss* yang besar.

## 7.2 Saran

Saran dari penelitian ini dijelaskan sebagai berikut:

1. Sistem dapat diimplementasikan pada sistem sebenarnya yang memanfaatkan penggunaan protokol MQTT. Contohnya adalah *Internet of Things* (IoT).
2. *Broker* protokol MQTT menggunakan *message broker* lain untuk mengimplementasikan metode *failover*.
3. *Broker* protokol MQTT menggunakan perangkat keras yang memiliki *resource* lebih besar karena penggunaan Raspberry Pi 3 Model B memiliki *resource* yang terbatas dan mengakibatkan peningkatan nilai *downtime*.