

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan penelitian yang telah dilaksanakan, maka didapatkan beberapa jawaban untuk menjawab rumusan masalah yang telah dijabarkan pada awal penelitian. Dari hasil pengujian dan analisis yang telah dilaksanakan maka diperoleh kesimpulan bahwa :

1. Penerapan *load balancing* untuk protokol komunikasi *MQTT* dapat diterapkan. Khususnya *load balancing* pada *broker mosquitto* bisa dilakukan menggunakan *HAProxy*, dimana *HAProxy* akan berjalan pada mode *TCP*, yang akan melakukan listen paket *TCP*, karena protokol *MQTT* berjalan pada protokol *TCP*.
2. Performa *load balancing* yang menggunakan algoritma *round robin* dalam mendistribusikan *load* mampu membagikan beban secara merata ketika semua *client* terkoneksi terus menerus tanpa terjadinya *disconnect*. *Load balancer* juga mampu mengurangi beban *broker* dengan hasil penurunan sebesar 40% ketika digunakan 2 *broker* dan 60% ketika digunakan 3 *broker*. Penurunan tersebut dibandingkan dengan kondisi dimana hanya ada 1 *broker* yang digunakan. Penurunan beban masih bisa bertambah bergantung dengan jumlah *broker* yang digunakan pada sistem.
3. Dengan diimplementasikannya *load balancing* untuk mendistribusikan *request* dari *subscriber*. Maka jika terjadi kegagalan pada *broker*, keseluruhan sistem masih bisa berjalan karena terdapat *broker* lain yang aktif dan dengan adanya *load balancing*, maka *subscriber* dapat melakukan rekoneksi ke *broker* lain. Adapun kecepatan di dapatkan pada penelitian ini adalah sebesar 1,007 detik ketika *traffic* di *broker* sedang tidak sibuk. Kemudian ketika *traffic* sedang tinggi dan *CPU Usage broker* sedang tinggi, maka diperlukan waktu lebih dimana pada percobaan ini didapatkan waktu yang lebih variatif, yaitu waktu 1,005 detik, 2,791 detik, 4,593 detik, dan 2,005 detik.
4. Ketika *broker* membagikan pesan *publish* dengan *broker* lain melalui *bridge*, maka terjadi peningkatan beban kerja *broker* pada *broker* yang bertanggung jawab untuk membagikan pesan. Pada penelitian ini ketika *broker* yang berperan sebagai *bridge* mengirim kan 125 ke 2 *broker* lainnya, *load* yang didapatkan sangatlah tinggi. Yaitu mengalami peningkatan sebesar 50%. Kemudian dari kecepatan distribusi pesan antar *broker* didapatkan hasil 0,362, 0,687, 0,891, 1,199, dan 1,622 detik untuk jumlah pesan *publish* sebesar 25 , 50, 75, 100, dan 125. Sehingga sangat disarankan untuk memberikan spesifikasi yang lebih tinggi untuk *broker* yang berperan sebagai *bridge*. Karena *broker* tersebut berperan sebagai jembatan agar *broker* lain bisa saling bertukar pesan *publish*.

7.2 Saran

Pada penelitian ini masih banyak kekurangan sehingga terdapat beberapa saran yang nantinya bisa digunakan untuk penelitian – penelitian sejenis lainnya ataupun pengembangan terhadap penelitian ini. Saran untuk pengembangan dari penelitian ini antara lain adalah :

1. Algoritma *load balancing* yang digunakan masih menggunakan *round robin*, sehingga untuk penelitian selanjutnya bisa dilakukan implementasi maupun analisis menggunakan algoritma *load balancing* lainnya semisal dengan mendeteksi utilisasi *resource* CPU setiap *broker* untuk menentukan distribusi beban kerja yang ada.
2. Karena penelitian ini menggunakan simulasi pada virtualisasi, maka pada penelitian selanjutnya, penelitian bisa dilakukan secara langsung pada *Raspberrry* ataupun *Arduino*.