

**ANALISIS PERFORMA *LOAD BALANCING* PADA *BROKER*
MQTT MENGGUNAKAN ALGORITMA *ROUND ROBIN***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Kevin Charlie
NIM: 145150200111197



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

ANALISIS PERFORMA LOAD BALANCING PADA BROKER MQTT MENGGUNAKAN ALGORITMA ROUND ROBIN

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh :
Kevin Charlie
NIM: 145150200111197

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Januari 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Rakhmadhany Primananda, S.T, M.Kom
NIK. 201609 860406 1 001



Mahendra Data, S.Kom, M.Kom
NIK. 201503 861117 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T, Ph.D

NIP. 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Desember 2017



Kevin Charlie

NIM: 145150200111197

KATA PENGANTAR

Salam sejahtera bagi kita semua, segala pujian, dan rasa syukur penulis ucapkan kepada Tuhan yang Maha Kuasa karena atas kasih dan berkat rahmat-Nya, penulis dapat menyelesaikan skripsi yang berjudul “ANALISI PERFORMA LOAD BALANCING PADA BROKER MQTT MENGGUNAKAN ALGORITMA ROUND ROBIN” dengan baik. Karena campur tangannya maka penelitian ini dapat diselesaikan. Adapun selama proses penelitian ini penulis telah banyak sekali mendapatkan bantuan dari berbagai pihak yang terus memberikan dukungan dalam bentuk bimbingan, motivasi, dan doa. Oleh karena itu penulis ingin mengucapkan terimakasih kepada :

1. Orang tua dan seluruh keluarga saya yang tidak ada henti – hentinya memberikan dukungan, motivasi, doa, dan energi yang sungguh luar biasa besar kepada penulis selama melakukan penelitian ini.
2. Bapak Rakhmadhany Primananda, S.T., M.Kom dan Bapak Mahendra Data, S.Kom., M.Kom selaku dosen pembimbing selama pelaksanaan skripsi yang selalu memberikan saran dan arahan kepada penulis.
3. Bapak Wayan Firdaus Mahmudi, S.Si., M.T., Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Agus Wahyu Widodo, S.T., M.CS selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Seluruh teman – teman baik saya (Bagus, Andre, Imam, Ayu, Salma, Bayu, Rizqi, Riski, Suhhy, Robi, Ridho, Yosua, Alan, Iskar, Afif, Arsa) yang selalu mendukung saya, membantu saya, hingga membackup saya selama masa pengerjaan skripsi berlangsung.
7. Semua pihak yang tidak disebutkan yang telah membantu penulis dalam kelancaran penelitian skripsi dan memberikan doa.

Penulis menyadari bahwa pada penelitian ini masih terdapat kekurangan dan kesalahan yang disebabkan oleh keterbatasan ilmu dan sumberdaya yang penulis miliki. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari pembaca sekalian. Akhir kata, semoga laporan skripsi ini dapat bermanfaat bagi kita semua. Semoga hari anda bahagia.

Malang, 1 Januari 2015

Penulis,
soratavin27@gmail.com

ABSTRAK

Kevin Charlie, Analisis Performa Load Balancing Pada Broker MQTT Menggunakan Algoritma Round Robin

Pembimbing: Rakhmadhany Primananda, S.T, M.Kom dan Mahendra Data, S.Kom, M.Kom

MQTT merupakan protokol komunikasi yang memerlukan *resource* dan bandwidth yang kecil. *MQTT* menggunakan *broker* yang berperan sebagai penghubung antara 2 *client*, yaitu *client publisher* dan *subscriber*. *Broker MQTT*, khususnya *Mosquitto* tidak memiliki mekanisme untuk mengatasi kegagalan ketika terjadi *overload CPU* ataupun ketika perangkat keras mengalami kerusakan. Selain itu *broker* harus selalu menjaga koneksi yang ada sehingga hal ini membebani kerja *broker*. Sehingga dibutuhkan *load balancer* dan beberapa *broker* untuk mengatasi permasalahan ini. Pada penelitian ini, dilakukan analisis performa *load balancer* yang menggunakan algoritma *round robin* dalam mendistribusikan beban kerja setiap *broker MQTT* yang ada. Algoritma *Round Robin* dipilih karena dapat mendistribusikan beban kerja secara merata ke semua *server* yang ada. Dengan menggunakan *round robin*, beban kerja setiap *broker* akan sama. Penelitian dilakukan pada sebuah *single host* menggunakan virtualisasi. Terdapat 4 *virtual devices* dalam 1 *host*. *Virtual Devices* tersebut adalah 3 *broker* dan 1 *load Balancer*. *Load balancer* menggunakan aplikasi *HAProxy*. Setiap *broker* bisa saling bertukar informasi dari *publisher*. Kemudian *subscriber* berlangganan topik dengan mengakses alamat IP *load balancer*. Pengujian dilakukan menggunakan *load balancer* dan tanpa menggunakan *load balancer* untuk mengetahui dampak yang diberikan *load balancer* terhadap *broker*. Didapatkan hasil saat sistem menangani *request* yang sama ketika menggunakan *load balancer* dan tidak menggunakan *load balancer*. Hasilnya didapatkan bahwa *load balancer* yang menggunakan algoritma *round robin* dapat mendistribusikan beban secara merata pada kondisi semua *client* terkoneksi tanpa ada yang terputus dengan beban sebesar 13,56%, 13,68%, dan 15,12% pada setiap *broker* dengan *request* sebesar 400. Kemudian sistem mampu melakukan *reconnect* ke *broker* lainnya dengan kecepatan 1,007 detik saat *traffic* sedang rendah dan 1,005, 2,791, 4,593, dan 2,005 detik ketika *traffic* sedang tinggi. Terakhir, distribusi pesan di *broker* memerlukan waktu sebesar hasil 0,362, 0,687, 0,891, 1,199, dan 1,622 detik untuk jumlah pesan *publish* sebesar 25, 50, 75, 100, dan 125.

Kata kunci: *load balancer, MQTT, round robin, publisher, subscriber, mosquitto, HAProxy*

ABSTRACT

Kevin Charlie, Analisis Performa Load Balancing Pada Broker MQTT Menggunakan Algoritma Round Robin

Pembimbing: Rakhmadhany Primananda, S.T, M.Kom dan Mahendra Data, S.Kom, M.Kom

MQTT is a communication protocol that requires a small resource and bandwidth. MQTT uses a broker that acts as a liaison between 2 clients, the publisher and subscriber. MQTT brokers, especially Mosquitto, do not have mechanisms to overcome failures that caused by CPU overload or hardware failure. In addition, the broker must always maintain the existing connection that burden the work of the each broker. So it takes load balancer and more than 1 brokers to overcome this problem. In this research, we conduct an analysis for load balancer performance using round robin algorithm to distribute the workload to each MQTT broker. The Round Robin algorithm is chosen because it can distribute the workload evenly to all existing servers. By using round robin, the workload of each broker will be the same. The study was conducted on a single host using virtualization. There are 4 virtual devices in 1 host. Virtual Devices are 3 brokers and 1 load balancer. Load balancer using HAProxy application. Every broker can exchange information from publishers. The subscriber subscribe to the topic by accessing the load balancer IP address. Testing is done using load balancer and without using load balancer to know the impact given load balancer to broker. From this research, it is found that when the system handles the same request when using load balancer and not using load balancer. The obtained result was a load balancer that use round robin algorithm could distribute workload evenly when client is always connected and no disconnect scenario occur with workload of 13.56%, 13.68%, and 15.12% for each broker when the number of request was 400. Then the system is able to reconnect to other brokers with a speed of 1.007 seconds when traffic is low and 1.005, 2.791, 4.593, and 2.005 seconds when traffic is high. Finally, the broker's message distribution takes about 0.362, 0.687, 0.891, 1.199, and 1.622 seconds for the number of publish messages of 25, 50, 75, 100, and 125.

Keywords: load balancer, MQTT, round robin, publisher, subscriber, mosquitto, HAProxy

DAFTAR ISI

| | |
|--|------------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR | iv |
| ABSTRAK..... | v |
| ABSTRACT | vi |
| DAFTAR ISI | vii |
| DAFTAR TABEL | xi |
| DAFTAR GAMBAR | xiv |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar belakang..... | 1 |
| 1.2 Rumusan masalah..... | 3 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan masalah | 3 |
| 1.6 Sistematika pembahasan | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 6 |
| 2.1 Kajian Pustaka | 6 |
| 2.2 Kajian Teori | 6 |
| 2.2.1 <i>Message Queueing Telemetry Transport (MQTT)</i> | 6 |
| 2.2.2 <i>Publish-Subscribe (Pub-Sub)</i> | 7 |
| 2.2.3 <i>Paho</i> | 7 |
| 2.2.4 <i>Mosquitto</i> | 7 |
| 2.2.5 <i>Load Balancing</i> | 8 |
| 2.2.6 <i>HAProxy</i> | 8 |
| 2.2.7 <i>Algoritma Round Robin</i> | 8 |
| BAB 3 METODOLOGI | 9 |
| 3.1 Identifikasi Masalah | 9 |
| 3.2 Studi Literatur | 9 |
| 3.3 Analisis Kebutuhan & Perancangan | 10 |
| 3.4 Implementasi | 10 |

| | |
|--|-----------|
| 3.5 Pengujian dan Analisis | 10 |
| 3.6 Kesimpulan..... | 10 |
| BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN | 11 |
| 4.1 Ruang Lingkup..... | 11 |
| 4.2 Analisis Kebutuhan | 11 |
| 4.2.1 Kebutuhan Fungsional..... | 11 |
| 4.2.2 Kebutuhan Perangkat Keras..... | 11 |
| 4.2.3 Kebutuhan Perangkat Lunak | 12 |
| 4.3 Batasan Perancangan..... | 12 |
| 4.4 Istilah..... | 12 |
| 4.5 Perancangan Sistem..... | 13 |
| 4.5.1 Perancangan Perangkat Keras | 13 |
| 4.5.2 Perancangan Perangkat Lunak..... | 13 |
| 4.5.3 Perancangan Arsitektur Jaringan | 14 |
| 4.5.4 Perancangan Alur Sistem di Sisi <i>Broker</i> | 15 |
| 4.5.5 Perancangan Alur Sistem di Sisi <i>Load Balancer</i> | 16 |
| 4.6 Perancangan Pengujian | 17 |
| BAB 5 IMPLEMENTASI | 20 |
| 5.1 Implementasi | 20 |
| 5.1.1 Spesifikasi Perangkat Keras..... | 20 |
| 5.1.2 Spesifikasi Perangkat Lunak | 21 |
| 5.1.3 Batasan Implementasi..... | 22 |
| 5.1.4 Implementasi Pada <i>Host</i> | 22 |
| 5.1.5 Implementasi Pada <i>Broker</i> | 24 |
| 5.1.6 Implementasi Pada <i>Load Balancer</i> | 28 |
| BAB 6 PENGUJIAN DAN ANALISIS | 31 |
| 6.1 Pengujian waktu untuk melakukan penyebaran pesan antar <i>broker</i> menggunakan <i>bridge</i> | 31 |
| 6.1.1 Tujuan Pengujian..... | 31 |
| 6.1.2 Langkah – langkah..... | 31 |
| 6.1.3 Data yang didapatkan | 31 |
| 6.1.4 Analisis..... | 33 |

| | |
|--|----|
| 6.2 Pengujian CPU <i>usage</i> ketika melakukan penyebaran pesan antar <i>broker</i> menggunakan <i>bridge</i> | 34 |
| 6.2.1 Tujuan Pengujian..... | 34 |
| 6.2.2 Langkah – langkah..... | 34 |
| 6.2.3 Data yang didapatkan | 34 |
| 6.2.4 Analisis..... | 39 |
| 6.3 Pengujian waktu yang dibutuhkan <i>broker</i> untuk menangani <i>request subscribe</i> | 41 |
| 6.3.1 Tujuan Pengujian..... | 41 |
| 6.3.2 Langkah – langkah..... | 41 |
| 6.3.3 Data yang didapatkan | 41 |
| 6.3.4 Analisis..... | 44 |
| 6.4 Pengujian CPU <i>usage broker</i> saat menangani <i>request subscribe</i> | 47 |
| 6.4.1 Tujuan Pengujian..... | 47 |
| 6.4.2 Langkah – langkah..... | 47 |
| 6.4.3 Data yang didapatkan | 47 |
| 6.4.4 Analisis..... | 50 |
| 6.5 Pengujian waktu ketika mengirimkan pesan <i>publish</i> ke <i>subscriber</i> | 51 |
| 6.5.1 Tujuan Pengujian..... | 51 |
| 6.5.2 Langkah – langkah..... | 51 |
| 6.5.3 Data yang didapatkan | 52 |
| 6.5.4 Analisis..... | 52 |
| 6.6 Pengujian untuk mengetahui CPU <i>usage broker</i> dalam menangani banyak <i>client</i> secara paralel dalam melakukan proses <i>publish-subscribe</i> | 54 |
| 6.6.1 Tujuan Pengujian..... | 54 |
| 6.6.2 Langkah – langkah..... | 54 |
| 6.6.3 Data yang didapatkan | 54 |
| 6.6.4 Analisis..... | 62 |
| 6.7 Pengujian integritas data | 63 |
| 6.7.1 Tujuan Pengujian..... | 63 |
| 6.7.2 Langkah – langkah..... | 64 |
| 6.7.3 Data yang didapatkan | 64 |
| 6.7.4 Analisis..... | 65 |

| | |
|--|-----------|
| 6.8 Pengujian rekoneksi ketika <i>broker</i> mengalami kegagalan | 65 |
| 6.8.1 Tujuan..... | 65 |
| 6.8.2 Langkah – langkah..... | 65 |
| 6.8.3 Data yang didapatkan | 66 |
| 6.8.4 Analisis..... | 67 |
| BAB 7 PENUTUP | 68 |
| 7.1 Kesimpulan..... | 68 |
| 7.2 Saran | 69 |
| DAFTAR PUSTAKA | 70 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1 Tabel Istilah | 12 |
| Tabel 4.2 Tabel Rancangan Pengujian | 18 |
| Tabel 5.1 Tabel Spesifikasi <i>Host</i> | 20 |
| Tabel 5.2 Tabel Spesifikasi <i>Broker 1</i> | 20 |
| Tabel 5.3 Tabel Spesifikasi <i>Broker 2</i> | 21 |
| Tabel 5.4 Tabel Spesifikasi <i>Broker 3</i> | 21 |
| Tabel 5.5 Tabel Spesifikasi <i>Load Balancer</i> | 21 |
| Tabel 6.1 Delay Sinkronisasi dengan jumlah <i>Publish 25 Pesan</i> | 31 |
| Tabel 6.2 Delay Sinkronisasi dengan jumlah <i>Publish 50 Pesan</i> | 32 |
| Tabel 6.3 Delay Sinkronisasi dengan jumlah <i>Publish 75 Pesan</i> | 32 |
| Tabel 6.4 Delay Sinkronisasi dengan jumlah <i>Publish 100 Pesan</i> | 32 |
| Tabel 6.5 Delay Sinkronisasi dengan jumlah <i>Publish 125 Pesan</i> | 33 |
| Tabel 6.6 Penggunaan <i>Resource CPU Broker 1</i> aktif dengan 25 <i>Publisher</i> | 34 |
| Tabel 6.7 Penggunaan <i>Resource CPU Broker 1</i> aktif dengan 50 <i>Publisher</i> | 35 |
| Tabel 6.8 Penggunaan <i>Resource CPU Broker 1</i> aktif dengan 75 <i>Publisher</i> | 35 |
| Tabel 6.9 Penggunaan <i>Resource CPU Broker 1</i> aktif dengan 100 <i>Publisher</i> | 35 |
| Tabel 6.10 Penggunaan <i>Resource CPU Broker 1</i> aktif dengan 125 <i>Publisher</i> | 36 |
| Tabel 6.11 Penggunaan <i>Resource CPU Broker 1 & 2</i> aktif dengan 25 <i>Publisher</i> ... | 36 |
| Tabel 6.12 Penggunaan <i>Resource CPU Broker 1 & 2</i> aktif dengan 50 <i>Publisher</i> ... | 36 |
| Tabel 6.13 Penggunaan <i>Resource CPU Broker 1 & 2</i> aktif dengan 75 <i>Publisher</i> ... | 36 |
| Tabel 6.14 Penggunaan <i>Resource CPU Broker 1 & 2</i> aktif dengan 100 <i>Publisher</i> . 37 | |
| Tabel 6.15 Penggunaan <i>Resource CPU Broker 1 & 2</i> aktif dengan 125 <i>Publisher</i> . 37 | |
| Tabel 6.16 Penggunaan <i>Resource CPU Broker 1 ,2, & 3</i> aktif dengan 25 <i>Publisher</i> | 37 |
| Tabel 6.17 Penggunaan <i>Resource CPU Broker 1 ,2, & 3</i> aktif dengan 50 <i>Publisher</i> | 37 |
| Tabel 6.18 Penggunaan <i>Resource CPU Broker 1 ,2, & 3</i> aktif dengan 75 <i>Publisher</i> | 38 |
| Tabel 6.19 Penggunaan <i>Resource CPU Broker 1 ,2, & 3</i> aktif dengan 100 <i>Publisher</i> | 38 |

| | |
|---|----|
| Tabel 6.20 Penggunaan <i>Resource CPU Broker 1, 2, & 3</i> aktif dengan 125 <i>Publisher</i> | 38 |
| Tabel 6.21 Delay waktu 100 <i>Subscriber</i> pada setiap <i>Broker</i> | 41 |
| Tabel 6.22 Delay waktu 200 <i>Subscriber</i> pada setiap <i>Broker</i> | 42 |
| Tabel 6.23 Delay waktu 300 <i>Subscriber</i> pada setiap <i>Broker</i> | 42 |
| Tabel 6.24 Delay waktu 400 <i>Subscriber</i> pada setiap <i>Broker</i> | 42 |
| Tabel 6.25 Delay waktu kecepatan 600 <i>Subscribe</i> menggunakan <i>load balancer</i> . | 42 |
| Tabel 6.26 Delay waktu kecepatan 1200 <i>Subscribe</i> menggunakan <i>load balancer</i> | 43 |
| Tabel 6.27 Delay waktu kecepatan 1800 <i>Subscribe</i> menggunakan <i>load balancer</i> | 43 |
| Tabel 6.28 Delay waktu kecepatan 1800 <i>Subscribe</i> menggunakan <i>load balancer</i> | 43 |
| Tabel 6.29 Perbandingan Delay waktu kecepatan <i>Subscribe</i> | 44 |
| Tabel 6.30 <i>Resource CPU Broker</i> dengan 100 <i>request subscribe</i> untuk setiap <i>broker</i> | 47 |
| Tabel 6.31 <i>Resource CPU Broker</i> dengan 200 <i>request subscribe</i> untuk setiap <i>broker</i> | 48 |
| Tabel 6.32 <i>Resource CPU Broker</i> dengan 300 <i>request subscribe</i> untuk setiap <i>broker</i> | 48 |
| Tabel 6.33 <i>Resource CPU Broker</i> dengan 400 <i>request subscribe</i> untuk setiap <i>broker</i> | 48 |
| Tabel 6.34 <i>Resource Broker</i> terhadap 300 <i>request subscribe</i> melalui <i>load balancer</i> | 48 |
| Tabel 6.35 <i>Resource Broker</i> terhadap 600 <i>request subscribe</i> melalui <i>load balancer</i> | 49 |
| Tabel 6.36 <i>Resource Broker</i> terhadap 900 <i>request subscribe</i> melalui <i>load balancer</i> | 49 |
| Tabel 6.37 <i>Resource Broker</i> terhadap 1200 <i>request subscribe</i> melalui <i>load balancer</i> | 49 |
| Tabel 6.38 Perbandingan Delay waktu <i>Publish</i> ketika menggunakan <i>load balancer</i> | 52 |
| Tabel 6.39 Perbandingan Delay waktu <i>Publish</i> ketika menggunakan <i>load balancer</i> | 52 |
| Tabel 6.40 Penggunaan <i>resource CPU</i> dengan hanya 1 <i>Broker</i> dengan 100 <i>subscriber</i> | 54 |
| Tabel 6.41 Penggunaan <i>resource CPU</i> dengan hanya 1 <i>Broker</i> dengan 200 <i>subscriber</i> | 55 |

| | |
|---|----|
| Tabel 6.42 Penggunaan <i>resource</i> CPU dengan hanya 1 <i>Broker</i> dengan 300 <i>subscriber</i> | 55 |
| Tabel 6.43 Penggunaan <i>resource</i> CPU dengan hanya 1 <i>Broker</i> dengan 400 <i>subscriber</i> | 55 |
| Tabel 6.44 Penggunaan <i>resource</i> CPU dengan hanya 2 <i>Broker</i> dengan 100 <i>subscriber</i> | 56 |
| Tabel 6.45 Penggunaan <i>resource</i> CPU dengan hanya 2 <i>Broker</i> dengan 200 <i>subscriber</i> | 56 |
| Tabel 6.46 Penggunaan <i>resource</i> CPU dengan hanya 2 <i>Broker</i> dengan 300 <i>subscriber</i> | 57 |
| Tabel 6.47 Penggunaan <i>resource</i> CPU dengan hanya 2 <i>Broker</i> dengan 400 <i>subscriber</i> | 58 |
| Tabel 6.48 Penggunaan <i>resource</i> CPU dengan hanya 3 <i>Broker</i> dengan 100 <i>subscriber</i> | 59 |
| Tabel 6.49 Penggunaan <i>resource</i> CPU dengan hanya 3 <i>Broker</i> dengan 200 <i>subscriber</i> | 60 |
| Tabel 6.50 Penggunaan <i>resource</i> CPU dengan hanya 3 <i>Broker</i> dengan 300 <i>subscriber</i> | 60 |
| Tabel 6.51 Penggunaan <i>resource</i> CPU dengan hanya 3 <i>Broker</i> dengan 400 <i>subscriber</i> | 61 |
| Tabel 6.52 Waktu yang dibutuhkan untuk melakukan reconnect saat <i>broker idle</i> | 67 |
| Tabel 6.53 Waktu yang dibutuhkan untuk melakukan reconnect saat <i>broker</i> sibuk | 67 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Struktur <i>Fixed Header</i> Paket MQTT (Oasis, 2015)..... | 7 |
| Gambar 3.1 Alur Penelitian | 9 |
| Gambar 4.1 Arsitektur Jaringan | 15 |
| Gambar 4.2 Diagram Alir Kerja Sistem di Sisi <i>Broker</i> | 16 |
| Gambar 4.3 Diagram Alir Kerja Sistem di Sisi <i>Load Balancer</i> | 17 |
| Gambar 5.1 Implementasi kode <i>Python</i> pada <i>Publisher</i> | 23 |
| Gambar 5.2 Implementasi kode <i>Python</i> pada <i>Subscriber</i> | 24 |
| Gambar 5.3 Konfigurasi Clock Speed <i>Virtual Device Broker</i> | 25 |
| Gambar 5.4 Konfigurasi RAM <i>Virtual Device broker</i> | 25 |
| Gambar 5.5 Konfigurasi <i>network interface</i> pada <i>virtual device</i> | 26 |
| Gambar 5.6 Versi <i>Mosquitto</i> yang digunakan..... | 26 |
| Gambar 5.7 Konfigurasi IP Static pada salah satu <i>broker</i> | 27 |
| Gambar 5.8 Konfigurasi <i>Bridging</i> pada salah satu <i>broker</i> | 27 |
| Gambar 5.9 Tampilan ketika <i>broker</i> berjalan di <i>foreground</i> | 28 |
| Gambar 5.10 Konfigurasi Clock Speed <i>Virtual Device Load Balancer</i> | 28 |
| Gambar 5.11 Konfigurasi RAM <i>Virtual Device Load Balancer</i> | 28 |
| Gambar 5.12 Konfigurasi <i>network interface</i> pada <i>virtual device</i> | 29 |
| Gambar 5.13 Versi <i>HAProxy</i> yang digunakan..... | 29 |
| Gambar 5.14 Konfigurasi IP Static pada Load Balancer | 29 |
| Gambar 5.15 Konfigurasi <i>HAProxy</i> pada Load Balancer | 30 |
| Gambar 6.1 Grafik waktu penyebaran pesan | 33 |
| Gambar 6.2 Grafik Penggunaan <i>Resource Broker 1</i> | 39 |
| Gambar 6.3 Diagram Batang Penggunaan <i>Resource</i> setiap <i>Broker</i> | 40 |
| Gambar 6.4 Arsitektur <i>bridge Mosquitto</i> | 40 |
| Gambar 6.5 Waktu yang dibutuhkan untuk menangani <i>subscriber</i> | 44 |
| Gambar 6.6 Komunikasi antara <i>subscriber, load balancer, dan broker</i> | 45 |
| Gambar 6.7 Waktu yang dibutuhkan <i>broker</i> untuk menangani <i>subscriber</i> | 46 |
| Gambar 6.8 Waktu yang dibutuhkan <i>broker</i> untuk menangani <i>subscriber</i> ketika menggunakan <i>load balancer</i> | 46 |

| | |
|--|----|
| Gambar 6.9 CPU Usage broker untuk menangani subscriber ketika menggunakan load balancer..... | 50 |
| Gambar 6.10 Hasil Capture Wireshark (1) | 51 |
| Gambar 6.11 Hasil Capture Wireshark (2) | 51 |
| Gambar 6.12 Selisih Waktu publish ketika menggunakan load balancer..... | 53 |
| Gambar 6.13 Komunikasi antara publisher, load balancer, dan broker | 53 |
| Gambar 6.14 CPU Usage broker 1..... | 63 |
| Gambar 6.15 Diagram penggunaan resource CPU pada setiap devices | 63 |
| Gambar 6.16 Publisher mengirimkan pesan ke broker..... | 64 |
| Gambar 6.17 Publisher mengirimkan pesan ke broker..... | 64 |
| Gambar 6.18 Subscriber menerima pesan..... | 64 |
| Gambar 6.19 Pesan yang dikirimkan oleh publisher..... | 65 |
| Gambar 6.20 Subscriber melakukan recovery ketika utilisasi CPU broker rendah | 66 |
| Gambar 6.21 Subscriber melakukan recovery ketika utilisasi CPU broker tinggi . | 66 |
| Gambar 6.22 Rekoneksi gagal..... | 66 |