

IMPLEMENTASI KLASTER KOMPUTER MINI RASPBERRY PI METODE LOAD BALANCING MENGGUNAKAN ALGORITMA ROUND ROBIN

Erfan Rohadi¹, Arief Prastyo², Mohammad Faried Rahmat³

^{1,2,3}Teknik Informatika, Teknologi Informasi, Politeknik Negeri Malang
¹erfan@polinema.ac.id, ²tiyok_pras@yahoo.com, ³fariedrahmad@gmail.com

Abstrak

Meningkatnya jumlah pengguna internet saat ini, membuat lalu lintas data meningkat secara signifikan. Ketika server tunggal mendapatkan request dari banyak client, besar kemungkinan akan terjadi overload. Sehingga request dari banyak client tidak dapat ditangani dengan maksimal. Hal ini mengakibatkan kinerja dari server menjadi menurun, Saat ini Untuk membangun sebuah cluster server dibutuhkan biaya yang sangat besar dan penggunaan daya yang besar. Berdasarkan permasalahan ini , maka dilakukan penelitian mengenai Implementasi Klaster Server Penelitian ini sangat menarik untuk dijadikan penelitian

Kata kunci : Raspberry Pi , Cluster Server , Load Balancing , WebServer

1. Pendahuluan

Meningkatnya jumlah pengguna internet saat ini, membuat lalu lintas data meningkat secara signifikan. Ketika server tunggal mendapatkan request dari banyak client, besar kemungkinan akan terjadi overload. Sehingga request dari banyak client tidak dapat ditangani dengan maksimal. Hal ini mengakibatkan kinerja server menjadi menurun.

Saat ini banyak pilihan untuk menangani masalah tersebut salah satunya adalah dengan cara melakukan clustering. Pada clustering menawarkan solusi untuk menangani pemerataan beban akses dari satu server ke server yang lain, salah satunya menggunakan metode load balancing.

Berdasarkan latar belakang tersebut, maka dilakukan penelitian mengenai Implementasi Klaster Server. Penelitian ini sangat menarik untuk dijadikan sebagai penelitian. Hal ini dikarenakan Raspberry Pi sebagai komputer kecil yang memiliki sistem operasi berbasis linux, tidak membutuhkan daya listrik besar.

2. Tinjauan Pustaka

2.1 Raspberry Pi

Raspberry Pi 3 adalah generasi ketiga dari Raspberry Pi. Model ini dirilis pada bulan Februari 2016 menggantikan Raspberry Pi 2 Model B. Dibanding Versi Sebelumnya , Pada Raspberry Pi 3 ini memiliki processor yang lebih baik, yaitu processor Quad-Core Broadcom BCM2837 64bit CPU dengan kecepatan 1,2GHz. Pada Raspberry Pi ini dilengkapi dengan ram sebesar 1 GB. Untuk bentuk dan fitur, pada model ini memiliki kemiripan dengan model Raspberry Pi 2 Model B.

Raspberry Pi Foundation juga menyediakan distro linux debian versi ARM Linux untuk di download, serta aplikasi pendukung untuk pemrograman seperti Python , Perl . Berikut ini Spesifikasi pada Raspberry Pi yang digunakan :

- Soc : Broadcom BCM2837
- CPU : 1,2 GHz , QuadCore ARM Cortex-A53
- GPU : DualCore VideoCore IV Multimedia Co-Processor
- Memory : 1GB LPDDR2
- USB : 4x USB 2.0 Ports
- Arus : 2,5 A
- Tegangan : 5V
- OS : Debian, Raspbian OS , Free BSD

2.2. Web Server

Web Server merupakan perangkat lunak yang memiliki fungsi untuk menerima permintaan berupa halaman web melalui protokol HTTP atau HTTPS dari suatu klien yang lebih dikenal dengan nama browser.

2.3 Cluster Server

ClusterServer Merupakan sekumpulan komputer yang beroperasi serta bekerja bersamaan dan terlihat oleh user jaringan seperti satu buah server.[5]

Kelebihan dari cluster server adalah kecepatan pengaksesan data yang lebih baik daripada komputer tunggal. Kelebihan lain dari cluster server ini adalah processor yang digunakan oleh cluster server dapat terus bertambah sesuai dengan jumlah processor yang di buat cluster

sehingga kemampuan yang diperoleh cluster server akan lebih baik dibandingkan server biasa.

2.4. Load Balancing

Load balancing merupakan sebuah cara untuk mendistribusikan beban traffic pada dua / lebih jalur koneksi secara seimbang, agar traffic data pada server dapat berjalan optimal, mempercepat waktu tanggap dan menghindari pada salah satu jalur koneksi.[1]

Load balancing diperlukan ketika jumlah user yang mengakses server telah melebihi kapasitas maksimal. selain itu load balancing juga berperan untuk mendistribusi beban kerja secara merata pada dua komputer atau lebih untuk mendapatkan sumber daya yang optimal.

2.5. Algoritma RoundRobin

Algoritma Round Robin merupakan salah satu penjadwalan proses, dimana algoritma ini menggilir proses yang ada di antrian.

Proses akan mendapatkan jatah sebesar time quantum. Jika time quantum-nya habis atau proses sudah selesai, CPU akan dialokasikan ke proses berikutnya. Pada penjadwalan proses ini, tidak ada proses yang diprioritaskan, semua proses mendapatkan pembagian waktu yang sama dari CPU.[2]

Kelemahan pada algoritma round robin ini adalah menentukan besarnya time quantum. Jika time quantum yang diberikan terlalu kecil, maka sebagian besar proses tidak akan selesai pada 1 quantum.

2.6. Httpperf

Httpperf merupakan tool yang sering digunakan untuk pengujian kinerja terhadap webserver. Pada Httpperf menyediakan fitur yang cukup lengkap dalam pembuatan beban kerja sesuai parameter yang ditentukan. Parameter yang ditentukan oleh software httpperf adalah Throughput, Response Time , Error , dan CPU Utilization.

Pada pengujian web server sederhana , klient akan membuat sejumlah permintaan kepada server, sehingga akan terukur hasil troughput yang berupa jumlah reply per second dari web server.

Cara paling mudah untuk mengukur performa web server yaitu dengan mengirim permintaan ke server dengan rute tertentu dan mengukur waktu permintaan tiba kembali.

Pengujian dilakukan dengan meningkatkan permintaan ke server tertentu hingga didapatkan server menjadi jenuh (saturated) dan percobaan dilakukan beberapa kali hingga didapatkan server mengalami padat akan traffic jaringan.[2]

Perintah berikut dapat digunakan untuk mengguji web server secara sederhana menggunakan “httpperf” :

```
Httpperf -hog -server [ipserver] - num -conn
100 -ra 50 -timeout 5
```

Perintah ini akan melakukan pengujian terhadap ip server server tertentu dengan menyediakan 100 koneksi dengan membuat 50 koneksi perdetik, dengan waktu timeout 5 detik.

2.7. Algoritma LeastConnection

Algoritma Least Connection adalah algoritma penjadwalan dengan mengarahkan koneksi jaringan dari server yang sedang aktif dengan jumlah koneksi yang paling sedikit.

Algoritma Penjadwalan ini merupakan salah satu algoritma penjadwalan yang bersifat dinamik, dikarenakan real server memerlukan perhitungan dari koneksi yang sedang aktif secara dinamik. Algoritma ini baik digunakan untuk melancarkan pendistribusian ketika request yang datang banyak (Rosada, 2013).

2.8. Nginx

Nginx Adalah server HTTP dan reverse proxy bebas berbasis open-source yang berkemampuan tinggi, juga dapat digunakan sebagai server proxy IMAP/POP3. Perangkat lunak ini diciptakan oleh Igor Sysoev pada tahun 2002, dan dirilis untuk pertama kalinya secara umum pada tahun 2004. Saat ini Nginx digunakan oleh 7,65% (22.8juta) nama domain di seluruh dunia.

2.9. Haproxy

Haproxy merupakan software yang digunakan untuk membangun sebuah High Availability server dan load balance unuk aplikasi berbasis TCP dan HTTP. Haproxy digunakan untuk membangun sebuah high availibility cluster , dimana load balance merupakan jalur utama seluruh req klien ke web server.

2.10. Respon Time

Respon Time merupakan waktu mengakses pada sebuah resource. Untuk mendapatkan response time peneliti menggunakan waktu yang dicatat pada skenario pengujian request. Pengujian request menggunakan tool Webserver Stress Tool.

2.11. Throughput

Throughput merupakan ukuran dari seberapa banyak data yang dapat diterima pada sebuah koneksi dengan interval waktu tertentu . Satuan umum throughput pada sebuah jaringan adalah bits dan byte per satuan waktu. Sistem Throughput biasanya merupakan nilai dari kemampuan dari server yang tersedia. Throughput dipengaruhi beberapa faktor :

1. Karakteristik workload (Beban kerja) yang akan dievaluasi

2. Karakteristik perangkat keras dan perangkat lunak sistem
3. Algoritma yang digunakan
4. Kecepatan perangkat keras dan perangkat lunak sistem.

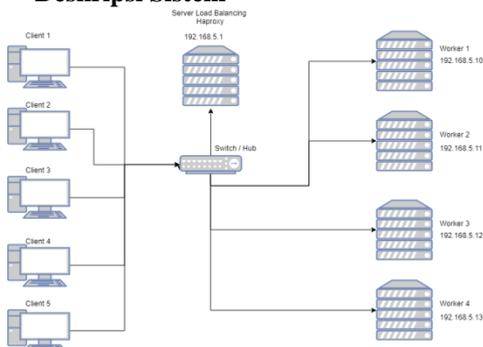
3. Metodologi

Dalam penelitian ini akan dilakukan 5 proses tahapan , yakni Studi Literatur , Analisa Kebutuhan untuk menentukan apa saja yang dibutuhkan untuk membangun cluster server. Selanjutnya dilakukan desain jaringan dengan metode load balancing menggunakan Algoritma round robin dan merealisasikan hasil desain tersebut menjadi suatu sistem yang terintegrasi.

Untuk tahapan selanjutnya adalah pengujian. Setiap unit server akan diuji apakah sesuai dengan rancangan yang telah dilakukan. Kedua , melakukan pengujian dengan cara membandingkan 1 buah webserver dengan 3 buah server raspberry yang di cluster, dan selanjutnya melakukan pengujian secara keseluruhan sebagai evaluasi hasil program dan sistem yang telah dirancang.

4. Implementasi Dan Pengujian

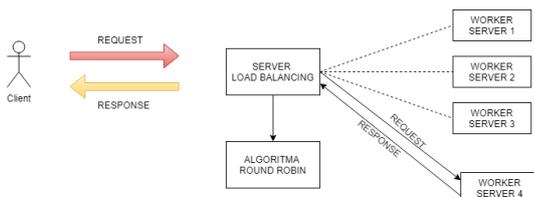
4.1 Deskripsi Sistem



Gambar 1. Topologi Cluster Server

Penulis merancang topologi jaringan cluster server, konsep pada topologi jaringan ini menggunakan 3 buah raspberry pi. 1 buah raspberry sebagai server utama, dan 2 server lagi digunakan untuk penyeimbang ketika server utama mengalami overload.

4.2 Diagram Blok



Gambar 2. Diagram Blok

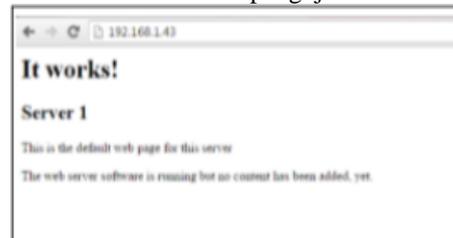
Pada use case ini terdapat client dan server , pada client melakukan request ke server dan server akan mendapatkan response. Client akan melakukan request ke server ke server yang telah di load balancing , ketika server mengalami beban yang

berat *algoritma round robin* akan bekerja untuk menyeimbangkan beban server agar *traffic* koneksi antar server menjadi lancar.

Algoritma Round Robin akan bekerja untuk menentukan giliran proses pada antrian setiap CPU. Proses Perpindahan antar server terdapat waktu sela (time quantum) secara berurutan.

4.3 Pengujian Cluster Server

Pengujian dilakukan didalam jaringan lokal dengan skema pengujian yang dijelaskan pada Gambar 2. Untuk melihat konten web, client harus mengakses IP 192.168.1.43. IP tersebut merupakan load balancer yang meneruskan permintaan dari back-end server. Pada penelitian ini diterapkan pengujian dengan cara membandingkan kinerja trafik dari server yang berjalan dengan load balancing dan tanpa load balancing. Pada Gambar 3 dan 4 dilihat bagaimana tampilan konten web sederhana untuk dilakukan pengujian



Gambar 3. Tampilan Server 1 Server Cluster



Gambar 4. Tampilan Server 2 Server Cluster

Untuk mengetahui bagaimana Raspberry Pi sebagai server dapat menangani lalu lintas data dari web yang diakses dengan menggunakan *load balancer*, dibutuhkan beberapa *user* yang mengakses web secara langsung. Pada penelitian ini dilakukan pengujian dengan 15 user yang mengakses web secara bersamaan.

Pengujian menggunakan httpperf, yaitu sebuah program tambahan untuk mengetahui lalu lintas data yang berjalan secara real-time

```
Httpperf -hog -server [ipserver] -num -conn 100 -ra 50 -timeout 5
```

Berikut adalah hasil yang didapat dari pengujian menggunakan httpperf :

```

Transactions:           13 hits
Availability:          100.00 %
Elapsed time:          59.49 secs
Data transferred:     0.02 MB
Response time:         0.01 secs
Transaction rate:      0.22 trans/sec
Throughput:            0.00 MB/sec
Concurrency:           0.00
Successful transactions: 13
Failed transactions:   0
Longest transaction:   0.02
Shortest transaction:  0.00
    
```

Gambar 5. Pengujian

4.4 Pengujian QOS Cluster Server

4.4.1 Skenario Pengujian WebServer Tunggal

Pada Skenario Pengujian ini Web Server Akan dilakukan proses stressing dengan menggunakan Aplikasi Web Server Stress Tool.

Tabel 1. Pengujian QOS Server Tunggal

Jumlah Klien	Jumlah Request	Waktu Respon	Request Loss %	Throughput
100	1000	78	0	16,33
200	1000	144	0	12,71
500	1000	202	0	8,47

4.4.2 Skenario Pengujian WebServer Cluster Algoritma Round Robin

Pada Skenario Pengujian ini webserver cluster dilakukan pengujian QOS dengan Algoritma Round Robin.

Tabel 2. Pengujian QOS Server Cluster Algoritma Round Robin

Jumlah Klien	Jumlah Request	Waktu Respon	Request Loss %	Throughput
100	1000	67,3	0	62,50
200	1000	95,3	0	76,42
500	1000	182,3	0	79,62

4.4.3 Skenario Pengujian WebServer Cluster Algoritma Least Connection

Pada Skenario Pengujian ini webserver cluster dilakukan pengujian QOS dengan Algoritma Least Connection. Penggunaan Algoritma Lain pada pengujian ini bertujuan untuk membandingkan kinerja antara web server cluster algoritma round robin dengan web server cluster algoritma least connection.

Tabel 3. Pengujian QOS Server Cluster Algoritma LeastConnection

Jumlah Klien	Jumlah Request	Waktu Respon	Request Loss %	Throughput
100	1000	400	0	32,50
200	1000	385	0	46,42
500	1000	350	0	55,40

5. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, dapat dibuktikan bahwasanya Raspberry Pi mampu digunakan sebagai web server dan load balancer dalam batas-batas tertentu. Raspberry Pi sebagai load balancer dengan Algoritma Round Robin mampu meminimalisir beban kinerja dari lalu lintas data web server. Selain itu juga, load balancer mampu menangani apabila ada terjadinya kerusakan diantara salah satu server dengan meneruskan permintaan ke server yang lainnya.

Daftar Pustaka:

Guizhi, Xu (2015). *The Design of a Client-Cloud Computing Collaborative Model*. International Journal of Grid and Distributed Computing, Vol. 8 No. 5

Khairul, Ansharullah.(2016): *Implementasi sistem load balancing dengan algoritma round robin untuk mengatasi beban server di smk negeri 2 kodus*.

Rido, Habi Putra & Sugeng, Winarno.(2016): *Implementasi Cluster Server pada Raspberry Pi dengan Menggunakan Metode Load Balancing*.

Zhou, Zhenji., Lifa, Wu., Zheng, Hong., (2013). *Context-Aware Access Control Model for Cloud Computing*. International Journal of Grid and Distributed Computing, Vol. 6 No. 6