

Perbandingan Algoritma *Round Robin* dan Algoritma *Least Connection* pada *Haproxy* untuk *Load Balancing Web Server*

¹Arip Solehudin, ²Rini Mayasari, ³Garno, ⁴Agung Susilo Yuda Irawan

^{1,2,3,4}Program Studi Teknik Informatika, Universitas Singaperbangsa Karawang

Email: arip.solehudin@staff.unsika.ac.id

Abstract

Sekolah Menengah Atas (SMA), Sekolah Menengah Kejuruan (SMK) or equivalent hold a grade promotion test using the Computer Based Test (CBT). However, in its implementation, there are problems, one of which is the server load. This is caused by user requests that exceed the server's capabilities. To overcome these problems, we use web server load balancing using Haproxy. However, the use of web server load balancing often experiences data inconsistencies per server. Therefore, database replication is used that can guarantee data consistency in web server load balancing. This study uses a descriptive approach to the PPDIIO network system design method. There are two load balancing algorithms compared, namely the round-robin algorithm and the least connection algorithm. Based on the httpperf test results, the throughput parameters of the two algorithms obtain the same average value of 99.5 Kb/second. In testing the response time parameter, the least connection algorithm is superior at 6.9 ms compared to round-robin, which has an average value of 7.2 ms. However, in testing the CPU Utilization parameter, the round-robin algorithm is worth 23.7% superior to the least algorithm connection is 24.3%.

Keywords: *Computer Based Test, load balancing web server, Database Replication*

Abstraksi

Sekolah Menengah Atas (SMA), Sekolah Menengah Kejuruan (SMK) atau yang sederajat menyelenggarakan ujian kenaikan kelas menggunakan Computer Based Test (CBT). Akan tetapi, dalam pelaksanaannya mengalami permasalahan, salah satunya beban server. Hal tersebut diakibatkan oleh permintaan pengguna yang melebihi kemampuan server. Untuk mengatasi permasalahan tersebut digunakan load balancing web server menggunakan haproxy. Namun penggunaan load balancing web server, sering mengalami inkonsistensi data tiap server. Oleh karena itu digunakan replikasi database yang dapat menjamin konsistensi data dalam load balancing web server. Penelitian ini menggunakan pendekatan deskriptif dengan metode perancangan sistem jaringan PPDIIO. Ada dua algoritma load balancing yang dibandingkan, yaitu algoritma round robin dan algoritma least connection. Berdasarkan hasil pengujian httpperf, parameter throughput kedua algoritma memperoleh nilai rata-rata yang sama yaitu 99,5 Kb / second. Pada pengujian parameter response time algoritma least connection lebih unggul yaitu 6,9 ms dibandingkan dengan round robin yang memiliki nilai rata-rata sebesar 7,2 ms, namun pada pengujian parameter CPU Utilization algoritma round robin bernilai 23,7 % lebih unggul dibandingkan algoritma least connection yaitu 24,3 %.

Kata kunci: Computer Based Test, load balancing web server, replikasi database

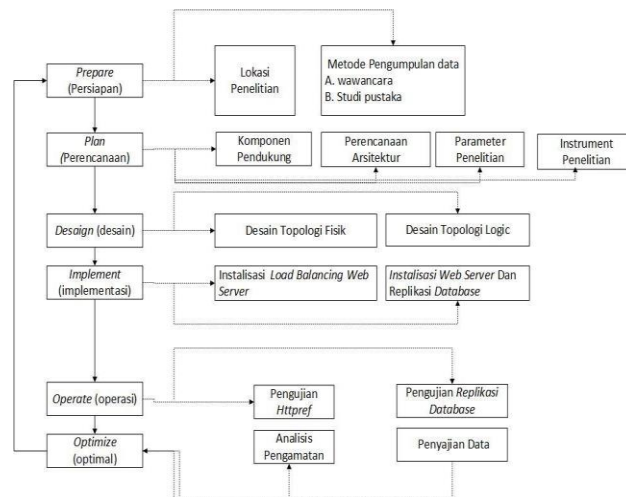
1. PENDAHULUAN

Perkembangan teknologi semakin meningkat, hal ini dibuktikan dengan adanya Ujian Nasional Berbasis Komputer (UNBK) yang dilakukan oleh Kementerian Pendidikan dan Kebudayaan. Berdasarkan data statistik tahun pelajaran 2018/2019 terdapat 13.305 pada jenjang SMK dengan jumlah peserta mencapai 1.515.994 [1] Salah satu permasalahan yang dihadapi dalam menyelenggarakan ujian kenaikan kelas berbasis CBT adalah ketersediaannya infrastruktur teknologi informasi yang kuat dan memadai untuk mengelola ratusan request client yang dilakukan secara bersamaan. Salah satu infrastruktur yang digunakan adalah server. Dengan semakin banyak nya client yang mengakses CBT serta masih menggunakan single server dalam penyedia infrastrukturnya, mengakibatkan server menjadi overload dan crash. Dampak tersebut mengakibatkan beban pada server menjadi meningkat dan waktu respon pada layanan tersebut menjadi lambat. Sehingga dengan adanya permasalahan tersebut, diperlukanya solusi yang digunakan untuk mengatasi beban server serta meningkatkan ketersediaan dan meminimalkan waktu tanggap dari web server [2]. Tidak adanya pembagian prioritas dalam penggunaan akses internet dapat menyebabkan terjadi kepadatan paket data [3] dan server yang mengalami kerusakan [4]. Pengukuran *Quality of Service* [5] dilakukan dengan mengukur jitter, delay, throughput, latency, packet loss [3].

Solusi dari permasalahan tersebut adalah menggunakan teknik Load balancing dengan sistem server clustering yang menggabungkan beberapa server untuk bekerja bersama-sama, sehingga dapat meningkatkan kemampuan server. Penerapan load balancing tidak menjamin konsistensi data, sehingga data base tidak dapat melayani request data client. Untuk mengatasi permasalahan pada penelitian sebelumnya, maka selain load balancing perlu diterapkan replikasi database untuk menjamin konsistensi data sehingga ketika client melakukan request pada server cluster, data request client dapat dilayani oleh node server yang berbeda. Dari uraian diatas, maka akan di lakukan pemabahasan mengenai penelitian yang berjudul “Perbandingan Algoritma Round Robin dan Algoritma Least Connection pada Haproxy untuk Load balancing Web server”.

2. METODE PENELITIAN

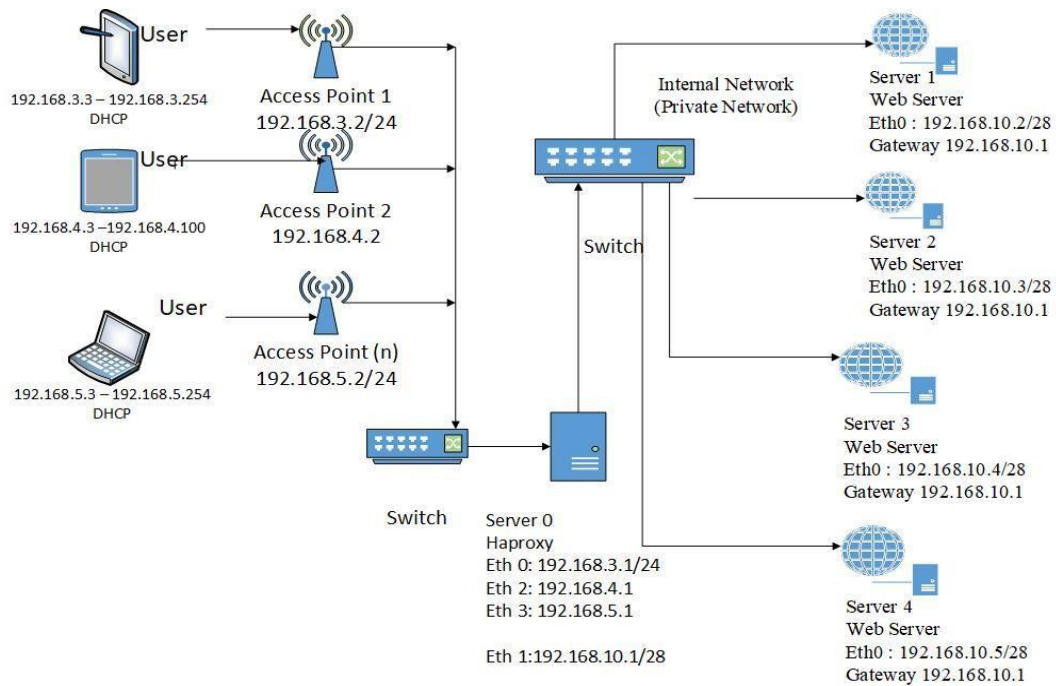
Dalam perancangan penelitian load balancing ini, menggunakan metode pengembangan system dengan tahapan Prepare, Plan, Desaign, Implement, Operate, Optimize (PPDIOO). Metode ini dipilih dengan alasan model pengembangan tahapan dapat mempresentasikan kebutuhan pada perancangan sistem load balancing web server ini. PPDIOO melibatkan enam tahap. Gambar 1. 1 merupakan tahapan yang digunakan dalam penelitian ini, tahapan tersebut dirancang berdasarkan metode PPDIOO.



Gambar 1. Tahapan Rancangan Penelitian

3. HASIL DAN PEMBAHASAN

Gambar 2. Dibawah ini merupakan Desain Topologi merupakan desain dari topologi logic. Server haproxy menggunakan beberapa interface yaitu eth0, eth1, eth2, eth3. Eth 0,2,3 terhubung dengan IP client sedangkan Eth1 terhubung dengan jaringan internal yang terhubung dengan 4 web server

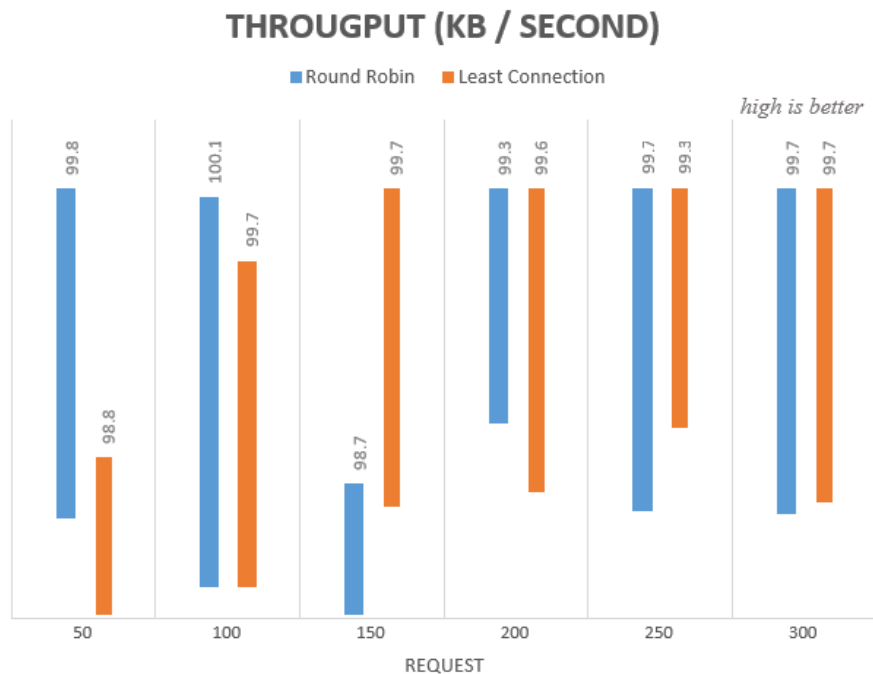


Gambar 2. Desain Topologi Jaringan

Tabel 1. Topologi Logic

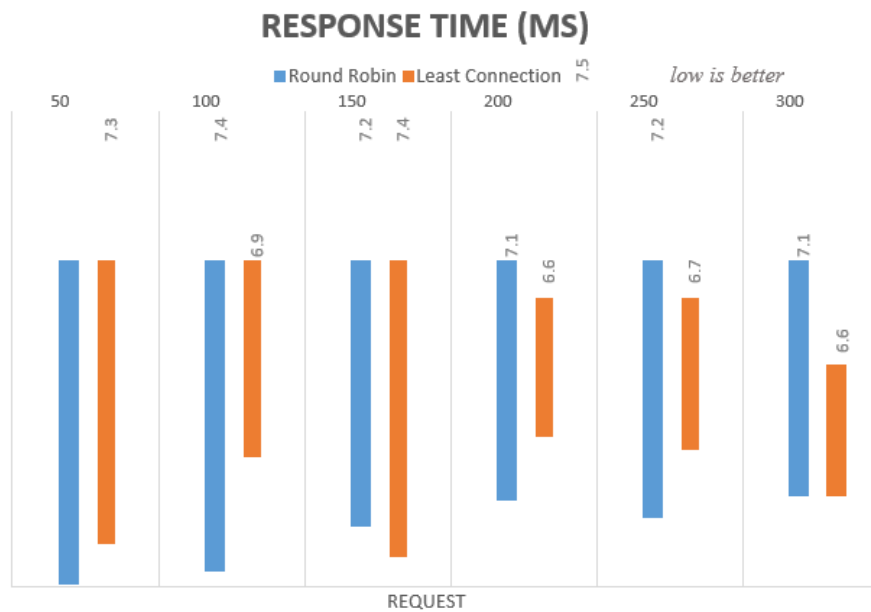
Perangkat	Interface/ IP Address	Keterangan
Server 0 Haproxy	Eth 0 :192.168.3.1/24	IP ke <i>client</i>
	Eth 2 :192.168.4.1/24	IP ke <i>client</i>
	Eth 3 :192.168.5.1/24	IP ke <i>client</i>
	Eth 1: 192.168.10.1/28	IP ke internal / <i>server cluster</i>

Access Point 1	192.168.3.2 – 192.168.3.254 /24	Range IP Koneksi ke <i>client</i>
Access Point 2	192.168.4.2 – 192.168.4.254 /24	Range IP Koneksi ke <i>client</i>
Access Point 3	192.168.5.2 – 192.168.5.254 /24	Range IP Koneksi ke <i>client</i>
Server 1 web server	Eth 0 : 192.168.10.2/28 Gateway : 192.168.10.1	IP web server 1
Server 2 web server	Eth 0 : 192.168.10.3/28 Gateway : 192.168.10.1	IP web server 2
Server 3 web server	Eth 0 : 192.168.10.4/28 Gateway : 192.168.10.1	IP web server 3
Server 4 web server	Eth 0 : 192.168.10.5/28 Gateway : 192.168.10.1	IP web server 4



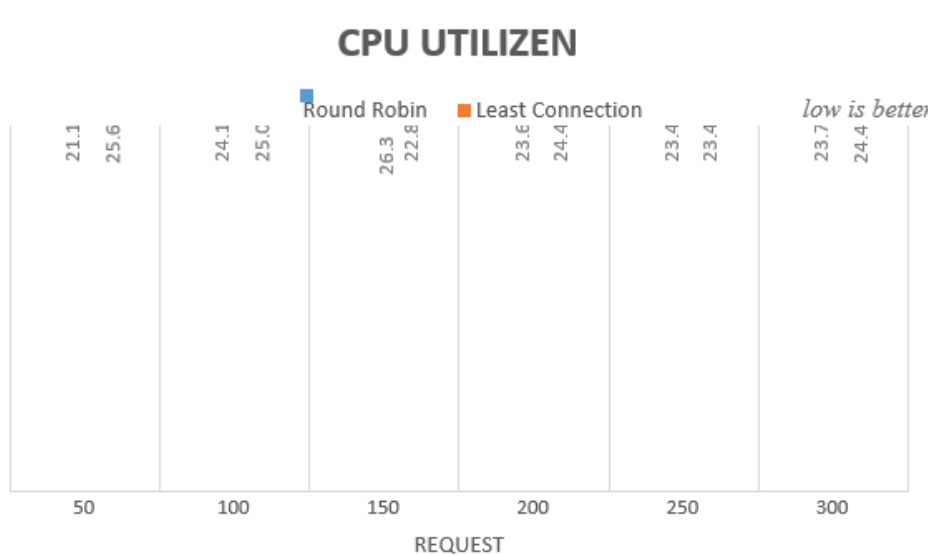
Gambar 2. Grafik Throughput perbandingan dua algoritma

Grafik *Throughput* perbandingan dua algoritma menunjukkan grafik perbandingan nilai *Throughput* algoritma *round robin* dan *least connection*. Pergerakan grafik dimulai dari request 50 pada algoritma *round robin* nilai *Throughput* yang dihasilkan sebesar 99,8 Kb / second. Nilai terendah terdapat pada request 150 dengan nilai *Throughput* yang dihasilkan sebesar 98,7 Kb / second dan nilai *Throughput* tertinggi terdapat pada request 100 sebesar 100,1 Kb / Second. Hal tersebut juga tidak berbeda jauh dengan algoritma *least connection*, nilai *Throughput* terendah terdapat pada request 50 sebesar 98,3 Kb / second dan nilai tertinggi terdapat pada request 100, 150, dan 300 sebesar 99,7Kb / second.



Gambar 3. Grafik *response time* perbandingan dua algoritma

Gambar 3. Menunjukkan grafik perbandingan nilai *response time* algoritma *round robin* dan *least connection*. Pergerakan grafik dimulai dari *request* 50 algoritma *round robin* dengan *response time* yang dihasilkan sebesar 7,5 ms dan mengalami perubahan yang tidak terlalu signifikan hingga *request* 300 yang menghasilkan *response time* sebesar 7,1 ms. Hal tersebut juga tidak berbeda jauh dengan algoritma *least connection*, pergerakan grafik dimulai dari *request* 50 dengan nilai *response time* sebesar 7,3 ms dan mengalami perubahan hingga *request* 300 dengan nilai *response time* sebesar 6,6 ms.



Gambar 4. Grafik *CPU Utilization* perbandingan dua algoritma

Gambar 4. Menunjukkan grafik perbandingan nilai *CPU Utilization* algoritma *round robin* dan *least connection*. Pergerakan grafik dimulai dari *request* 50 pada algoritma *round robin* dengan *CPU Utilization* yang dihasilkan sebesar 21,1 % dan mengalami perubahan yang tidak terlalu signifikan

hingga *request* 300 dan menghasilkan *CPU Utilization* sebesar 23,7%. Hal tersebut juga tidak berbeda jauh dengan algoritma *least connection*, pergerakan grafik dimulai dari *request* 50 dengan nilai *CPU Utilization* sebesar 25,6 % dan mengalami perubahan hingga *request* 300 dengan nilai 24,4 %.

Tabel 2. Hasil Percobaan Replikasi *database*

Skenario	Dampak	Kondisi setelah di pulihkan
1	Normal, <i>Node</i> yang lain tetap dapat beroperasi.	Normal, Replikasi kembali berjalan, Data Sinkron.
2	Normal, <i>Node</i> yang lain tetap dapat beroperasi.	Normal, Replikasi kembali berjalan, Data Sinkron
3	Normal, <i>Node</i> yang lain tetap dapat beroperasi.	Normal, Replikasi kembali berjalan, Data Sinkron.

Dari percobaan yang telah dilakukan berdasarkan skenario 1, skenario 2 dan skenario 3 proses replikasi *database* berjalan dengan normal tanpa adanya *error* serta setiap *node* dapat beroperasi dengan baik.

4. KESIMPULAN

Setelah melakukan penelitian tentang *load balancing web server* menggunakan *haproxy* dan replikasi *database* pada *Computer Based Test* dengan membandingkan algoritma *round robin* dan *least connection* berdasarkan parameter yang telah ditentukan serta menguji konsistensi data pada *server cluster* dengan beberapa skenario pengujian, maka didapatkan kesimpulan sebagai berikut:

1. Berdasarkan hasil pengujian, maka algoritma yang akan digunakan dalam *load balancing web server* menggunakan *haproxy* dan replikasi *database* pada *Computer Based Test* adalah *least connection*. Alasan pemilihan algoritma *least connection* dikarenakan memiliki keunggulan pada *response time* dengan perolehan rata-rata sebesar 6,9 *ms* lebih cepat dari *round robin* yang mendapatkan rata-rata sebesar 7,2 *ms* sehingga siswa tidak akan menunggu lama untuk akses halaman *Computer Based Test*.
2. Penerapan replikasi *database* pada *load balancing web server* dapat mencegah terjadinya inkonsistensi data tiap *node*. Ketika salah satu *node* mengalami gangguan maka akan dibackup oleh *node* lainnya dan saat *node* tersebut dipulihkan, data akan otomatis di perbaharui dengan fitur sinkronisasi *galera cluster*, sehingga konsistensi data dapat terjamin. Kesimpulan harus mengindikasikan secara jelas hasil-hasil yang diperoleh dan kelebihan dan kekurangannya. Kesimpulan berupa paragraf, **tidak boleh** berbentuk point-point atau bullet atau numbering.

DAFTAR PUSTAKA

- [1] Kementrian Pendidikan dan Kebudayaan. (2019, Februari 21).
- [2] Artha, Y. (2017). Penerapan Metode Round Robin Pada Jaringan Multihoming Di Computer Cluster. *IT Journal Research and Development*, 1(2), 26-35.
- [3] A. Solehudin and G. Garno, "PROTOTYPE API PADA APLIKASI PEMBATAHAN AKSES INTERNET DENGAN PEMANFAATAN HAK AKSES USER PROFILE HOTSPOT", *JURNAL REKAYASA INFORMASI*, vol. 6, no. 2, Oct. 2017.
- [4] Euis, E., & Heryana, N. (2018). Analisis Pemanfaatan Cloud Computing Berbasis Software as a Service sebagai Media Penyimpanan Tugas Praktikum. *Syntax Jurnal Informatika*, 7(1), 25–31.
- [5] Arip Solehudin ,Bayu Priyatna ,Nono Heryana "ANALYSIS OF EFFECT OF ZFONE SECURITY ON VIDEO CALL SERVICE IN WIRELESS LOCAL AREA NETWORK" Vol. 6 - Issue 6(1-8) - November - December2019 , *International Journal of Computer Techniques (IJCT)* , ISSN: 2394 - 2231 , www.ijctjournal.org
- [6] Ardiansyah, S., Nur, J., & Mukmin, M. (2017). Rancang Bangun Load balancing Pada Database Cluster Menggunakan Haproxy. *Jurnal Informatika*.