

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Tinjauan pustaka yang digunakan dalam penelitian ini merupakan beberapa penelitian yang telah dilakukan sebelumnya mengenai *load balancing*.

Tabel 2.1 Tinjauan Pustaka

NO	PARAMETER PENULIS	OBYEK PENELITIAN	METODE / TEKNOLOGI	HASIL
1	Edriansyah dan Wedy Freddy (2017)	Rancang Bangun <i>Load Balancing Apache Webserver Dan Database Cluster</i>	<i>HA Proxy, Round Robin</i>	Dengan <i>load balancer</i> layanan web server tetap tersedia ketika salah satu <i>server down</i> , sehingga <i>user</i> dapat terus mengakses seakan-akan tidak ada masalah pada <i>server</i>
2	Andri Duwi Saputro (2017)	Teknologi <i>load balancing database</i> pada aplikasi berbasis web	<i>HA Proxy, Round Robin, Galera Cluster</i>	<i>Round robin</i> pada <i>HA Proxy</i> akan mendahulukan file dengan kapasitas kecil untuk menyesuaikan <i>time quantum</i>
3	Alam Rahmatulloh dan Firmansyah MSN (2017)	Implementasi <i>Load Balancing Web Server</i> menggunakan <i>HA proxy</i> dan Sinkronisasi <i>File</i>	<i>HA Proxy, Round Robin</i>	Dengan implementasi <i>clustering</i> , <i>server</i> dapat meningkatkan <i>respon time web server</i> dan meningkatkan jumlah <i>current connection</i> yang dapat dilayani oleh <i>server</i>
4	Data, M., Ramadhan, G., & Amron, K. (2017)	Analisis Availabilitas dan Reliabilitas <i>Multi-Master Database Server</i> Dengan	<i>MariaDB, Galera Cluster</i>	Dengan menggunakan tiga <i>node</i> dalam satu <i>cluster database</i> , <i>availabilitas</i> dan <i>reliabilitas MariaDB Galera Cluster</i> dapat

		State Snapshot Transfers (SST) Jenis Rsync Pada MariaDB Galera Cluster.		terjaga dengan baik
5	Usulan : Sarwan Hamid (2018)	Implementasi Load Balancing Database Pada Aplikasi Transaksi Central Laundry Berbasis Web	HA Proxy, Least Connection	-

## 2.2 Dasar Teori

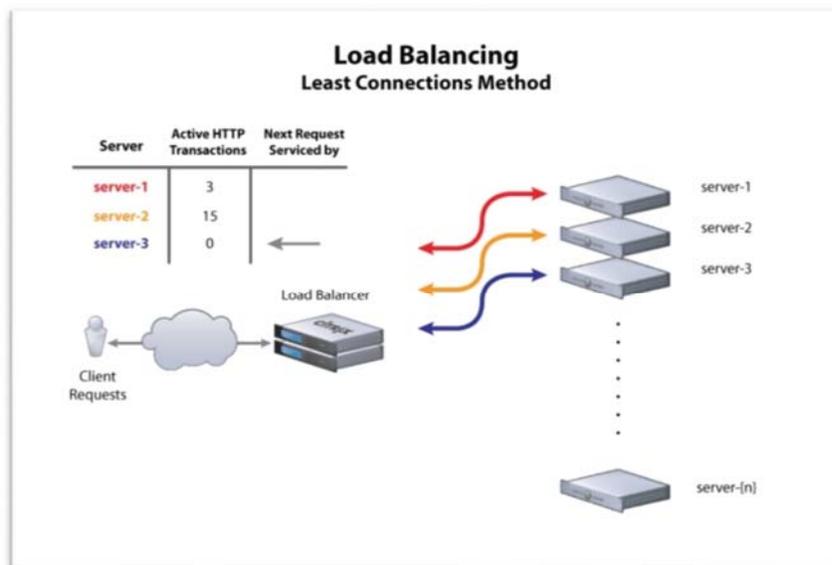
### 2.2.1 Pengertian Load Balancing

*Load balancing* adalah teknik untuk mendistribusikan beban pada dua atau lebih jalur koneksi secara seimbang, agar dapat berjalan optimal, memaksimalkan *throughput*, memperkecil waktu tanggap dan menghindari *overload* pada salah satu jalur koneksi.

*Load balancing* digunakan pada saat sebuah *server* telah memiliki jumlah *user* yang telah melebihi maksimal kapasitasnya. *Load balancing* juga mendistribusikan beban kerja secara merata di dua atau lebih komputer, *link* jaringan, CPU, *hard drive*, atau sumber daya lainnya, untuk mendapatkan pemanfaatan sumber daya yang optimal. Selain itu, Yang (2016) menyatakan tujuan utama *load balancing* adalah untuk membagikan beban secara adil ke seluruh *nodes* yang ada. (Dani & Suryawan, 2017)

### 2.2.2 Least Connection

*Least Connection* merupakan algoritma pembagian beban berdasarkan banyaknya koneksi yang sedang dilayani oleh sebuah *server*. *Server* dengan koneksi yang paling sedikit akan diberikan beban berikutnya, begitu pula *server* dengan koneksi banyak akan dialihkan bebannya ke *server* lain yang bebannya lebih rendah (Ellord, 2010)



Gambar 2.1 least connection method (Ellord, 2010)

*Least Connection* merupakan salah satu algoritma penjadwalan dinamik karena memerlukan perhitungan koneksi aktif untuk masing - masing *real server* secara dinamik. Metode penjadwalan ini baik digunakan untuk melancarkan pendistribusian ketika *request* yang datang sangat banyak.

### 2.2.3 Galera Cluster

*Galera Cluster* adalah sinkronasi *cluster multi-master database* berdasarkan sinkronasi replikasi dan Oracle MySQL / InnoDB. *Galera Cluster* merupakan

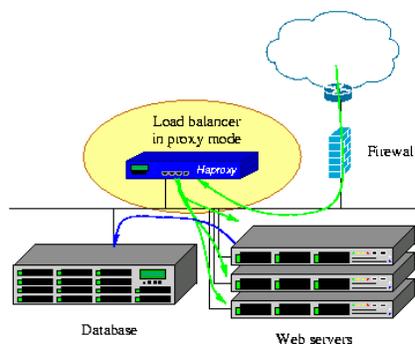
solusi ketersediaan tinggi yang menyediakan sistem yang *uptime* dan tidak ada kehilangan data skalabilitas untuk pertumbuhan.

#### 2.2.4 MariaDB (Database)

*MariaDB* adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Kemampuan *MariaDB* Dapat berjalan stabil pada berbagai sistem operasi. Dapat didistribusikan sebagai perangkat lunak sumber terbuka di bawah lisensi GPL Dapat digunakan oleh beberapa *user* dalam waktu bersamaan Dapat memproses lebih banyak SQL per satuan waktu.

#### 2.2.5 HA Proxy

*HA Proxy* adalah sebuah produk *opensource* yang dibuat secara khusus untuk dapat melakukan pembagian *request* ke arah server. HA sendiri singkatan dari *High Availability*, yang secara teknis artinya adalah kondisi tidak akan mati saat salah satu sumber server mengalami kondisi *down* atau *overload*.



Gambar 2.2 Arsitektur HA Proxy

Fitur - fitur dari HA Proxy :

1. Dapat melakukan *load balancing* untuk banyak *server*

2. Dapat membuat *load balancing master - slave* sehingga apabila *load balancing* yang master mati maka akan digantikan oleh *slave*
3. Dapat digunakan oleh banyak *service* seperti email, *database*, *web server* dan lain - lain.
4. Terdapat statistik *load balancing* sehingga mempermudah kita memantau beban *trafik*
5. Jika salah satu server mati, maka beban *trafik* ke server tersebut akan dihentikan.

### **2.2.6 Web Server**

*Web server* adalah perangkat lunak (*software*) dalam server yang berfungsi untuk menerima permintaan (*request*) berupa halaman web melalui protokol HTTP dan atau HTTPS dari klien (*browser*), kemudian mengirimkan kembali (*respon*) hasil permintaan tersebut ke dalam bentuk halaman-halaman web yang pada umumnya berbentuk dokumen HTML, *image*, CSS, atau *javascript*. *Web server* berguna sebagai tempat aplikasi web dan sebagai penerima *request* dari klien.