Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202184512
Vol. 8, No. 2, April 2021, hlm. 329-332 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 39/E/KPT/2019 e-ISSN: 2528-6579

IMPLEMENTASI ARSITEKTUR *WEB SERVER CLUSTER* MENGGUNAKAN SINGLE BOARD COMPUTER UNTUK MENUNJANG KEBUTUHAN *HIGH AVAILABILITY* SYSTEM

Roisul Setiawan[1], Dany Primanita Kartikasari[*2], Bayu Rahayudi[3]

[1,2,3]Fakultas Ilmu Komputer Universitas Brawijaya
Email: [1]roisulsetiawan@student.ub.ac.id, [2]dany.jalin@ub.ac.id, [3]ubay1@ub.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 16 Desember 2020, diterima untuk diterbitkan: 22 Maret 2021)

Abstrak

Untuk mewujudkan ketahanan pangan, diperlukan mekanisme pengumpulan data secara real-time dari produsen bahan pangan, pendisitribusi bahan pangan sampai pengolah bahan pangan. Namun tidak semua organisasi yang berkecimpung dalam distribusi pangan memiliki infrastruktur sistem informasi yang cukup baik. Untuk mengatasi kendala infrastruktur, penelitian ini mengusulkan untuk membangun arsitektur *web server cluster* yang dapat menunjang kebutuhan *high availability* system menggunakan single board computer. Komponen arsitektur terdiri dari dua tier yaitu: frontend dan backend. Untuk menjamin kehandalan sistem, arsitektur yang diusulkan didukung dengan komponen *load balancing*, mekanisme failover dan replikasi database. Sistem telah diuji berasarkan kebutuhan fungsional dan kebutuhan non-fungsional yang sudah didefinisikan sesuai kebutuhan organisasi. Dari hasil pengujian, tingkat availabilitas yang dihasilkan sebesar 95.83%.

Kata kunci: *high availability, web cluster, single board computer*

IMPLEMENTATION OF WEB SERVER CLUSTER USING SINGLE BOARD COMPUTER TO PROVIDE HIGH AVAILABILITY SYSTEM

Abstract

To achieve food security, a real-time data collection mechanism is needed from food producers, food distribution to food processing. However, not all organizations involved in food distribution have adequate information system infrastructure. To overcome infrastructure constraints, this study proposes to build a web server cluster architecture that can support the needs of a high availability system using a single board computer. The architectural component consists of two tiers, namely: frontend and backend. To ensure system reliability, the proposed architecture is supported by load balancing components, failover mechanisms, and database replication. The system has been tested based on functional requirements and non-functional requirements that have been defined according to organizational requirements. From the test results, the resulting availability level is 95.83%.

.

Keywords: *high availability, web cluster, single board computer*

1. PENDAHULUAN

Dalam undang-undang No. 18/2012, distribusi pangan merupakan tindakan untuk menyalurkan pasokan pangan secara merata setiap saat guna memenuhi kebutuhan pangan masyarakat. (Kementerian Hukum dan HAM Republik Indonesia, 2012) Distribusi pangan menjadi salah satu faktor penentu keberhasilan program ketahanan pangan untuk memastikan setiap penduduk mendapatkan akses pangan yang cukup dan bergizi. Distribusi pangan ini diharapkan dapat terlaksana secara efektif, efisien dan merata di setiap lokasi berlangsungnya transaksi bahan pangan kebutuhan masyarakat. Program ini dapat berhasil bila didukung oleh data akurat dan real time mengenai kondisi produksi dan distribusi pangan. Ketersediaan data produksi dan distribusi pangan menjadi terkendala bila dihadapkan pada beragamnya kondisi daerah di Indonesia. Dalam Undang-undang No 17 tahun 2015 tentang ketahanan pangan dan gizi, dinyatakan bahwa Pemerintah dan Pemerintah Daerah sesuai dengan

kewenangannya bertanggung jawab atas distribusi pangan. (Kementerian Hukum dan HAM Republik Indonesia, 2015) Salah satu bentuk tanggung jawab ini adalah dengan menyediakan infrastruktur yang mendukung pengembangan sistem distribusi pangan yang menjangkau seluruh wilayan secara efektif dan efisien. Untuk mendapatkan data produksi dan distribusi, tentu diperlukan infrastruktur penyedia data yang meliputi seluruh wilayah yang berada dalam lingkup tanggung jawab pemerintah pusat maupun pemerintah daerah. Untuk sebagian industri pangan maupun distribusi pangan yang berskala kecil dan menengah tidak mampu mewujudkan kebutuhan infrastruktur yang memadai. Hal ini disebabkan terbatasnya permodalan yang dimiliki. Untuk mengatasi permasalahan tersebut, diperlukan infrastruktur yang dapat mendukung industri kecil dan menengah di seluruh wilayah Indonesia dalam menyediakan data akurat dan real-time dengan segala keterbatasan yang ada. Sistem ini haruslah bersifat ringan dalam segi biaya dan tingkat kompleksitasnya rendah. Namun memerlukan sistem dengan tingkat ketersediaan tinggi untuk menjamin proses transaksi data yang terjadi tetap dapat berjalan dengan baik.

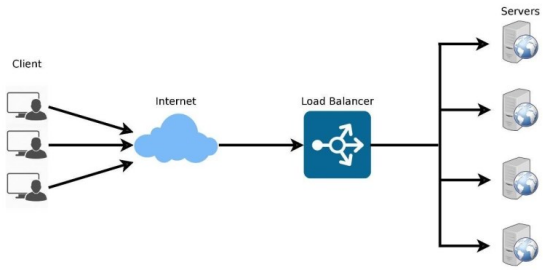
Sistem informasi yang digunakan dalam penyediaan data distribusi pangan umumnya menggunakan mekanisme input data dan akses data sederhana. Arsitektur client-server yang dibangun terdiri dari dua tier, yaitu web server dan database server. Dalam perkembangan teknologi web server, telah dikembangkan metode metode yang dapat menjamin ketersediaan akses pada web server. Usaha untuk mempertahankan kestabilan adalah dengan mengaplikasikan sistem kluster pada web server. (Moon & Cho, 2004) Dengan menggunakan sistem cluster, terdapat dua metode yang dapat diimplementasikan, yaitu: *load balancing* dan failover. (Data, et al., 2019) Metode ini telah dikembangkan dalam beberapa penelitian terdahulu seperti pada penelitian (Sharma, 2017) dan (Li, et al., 2020). Namun metode metode ini dikembangkan untuk regular server dengan perangkat yang memiliki sumberdaya tinggi. Untuk diimplementasikan pada institusi dengan keterbatasan biaya, tentu menjadi sebuah permasalahan yang cukup serius untuk diatasi. Salah satu alternatif pengganti server berbiaya rendah yang sudah dikembangkan adalah perangkat single-board computer. Perangkat ini memiliki kemampuan sebagai mini computer namun cukup untuk digunakan di organisasi dengan penggunaan internal. Penelitian implementasi *load balancing* pada single-board computer telah dilakukan dalam (Maduranga & Ragel, 2016) dan direkomendasikan untuk digunakan dalam web server berbasis cluster.

Penulisan paper ini disusun dengan sebagai berikut: bagian pertama menyampaikan tentang latar belakang penelitian dilakukan dan identifikasi permasalahan. Bagian dua menjelaskan tentang hasil kajian pustaka. Bagian ketiga menyampaikan tentang rancangan arsitektur yang diusulkan. Bagian ke empat berisi tentang implementasi dan pengujian. Bagian ke lima memberikan diskusi pembahasan. Bagian penutup memberikan kesimpulan dari penelitian yang dilakukan.

2. KOMPUTASI KLASTER

Klaster komputer adalah sekumpulan komputer mandiri yang terintegrasi secara software maupun hardware bekerja bersama sama untuk menjalankan tugas bersama. Secara fungsi terdapat tiga kategori klaster server, yaitu: *high performance cluster, load balancing cluster* dan *high availability cluster*. (Yeo, et al., 2006) Penelitian ini menggunakan *high availability* cluster, dengan tujuan untuk mendapatkan tingkat availabilitas yang tinggi.

Untuk mendapatkan tingkat availabilitas yang tinggi diperlukan mekanisme *load balancing* untuk membagi load pekerjaan secara seimbang. Mekanisme *load balancing* dapat diilustrasikan seperti pada Gambar 1.

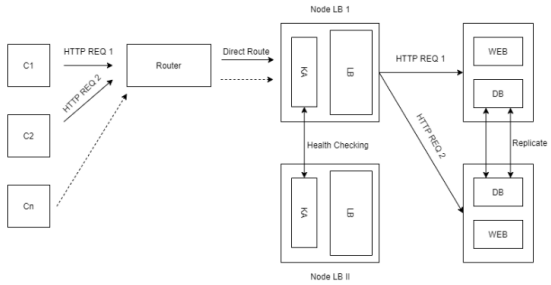


Gambar 1. Ilustrasi fungsi *load balancer*

Metode konvensional untuk mendapatkan sistem yang tangguh adalah menggunakan teknik redudansi. Salah satu teknik redundansi pada database adalah melakukan replikasi database. Replikasi merupakan proses menyalin dan memelihara obyek database. (Pohanka & Pechanec, 2020) Teknik ini cukup efektif dalam menduplikasi data dari server master ke satu atau lebih server slave. Replikasi bekerja dengan cara asinkronous. Terdapat dua jenis teknik replikasi yang dapat digunakan, yaitu *eager replication* dan *lazy replication*. (Gudiu, et al., 2010)

3. RANCANGAN SISTEM

Sistem yang diusulkan menggunakan arsitektur two-tier, terdiri dari lapisan pertama berfungsi sebagai *front-end* dan lapisan kedua berfungsi sebagai *backend.* Ilustrasi dari arsitektur yang diusulkan ditampilkan dalam Gambar 2.



Gambar 2. Arsitektur Sistem

Pada bagian *front end*, setiap node akan memiliki bagian yang berfungsi sebagai *load balancing* node. Pada kedua node ini saling bertukar informasi yang disebut dengan *Health Check*. Informasi ini memberikan keterangan mengenai kondisi masing masing node yang digunakan sebagai dasar memutuskan node yang dipergunakan sebagai node utama. Pada bagian backend, setiap node akan memiliki modul web server dan database yang saling mereplikasi secara berkala. Kedua modul ini akan memproses HTTP REQUEST yang dikirimkan oleh node *load balancing* yang bertugas menerima request.

4. IMPLEMENTASI DAN PENGUJIAN

Pada penelitian ini, sistem diimplementasikan menggunakan Raspberry Pi Zero-W dengan single core CPU 1 Ghz dan menggunakan RAM sebesar 512 MB. Perangkat ini berfungsi sebagai *load balancer* dan backend server. Untuk perangkat lunak yang dipergunakan, ditunjukkan dalam Tabel 1.

Tabel 1. Perangkat Lunak

No	Perangkat Lunak	Versi	Perangkat
1	Rasberry Pi OS (32-bit) Lite	Jan 2020	Semua Perangkat
2	Haproxy	2.1	Load Balancer
3	Keepalived	2.0.19	Load Balancer
4	Nginx	1.18.0	Backend Server
5	PHP	7.3	Backend Server
6	MariaDB	10.3.15	Backend Server
7	Rsync	3.0	Semua Perangkat

Adapun secara fungsional, sistem diharapkan dapat memenuhi beberapa kriteria seperti ditunjukkan dalam Tabel 2.

Tabel 2. Kebutuhan Fungsional

No.	Kebutuhan Fungsional
KF1.	*Cluster server* mampu beroperasi dan melayani permintaan user meskipun ada perangkat yang mengalami kegagalan
KF2.	*Cluster server* mampu menyimpan data secara konsisten meskipun ada perangkat yang mengalami kegagalan
KF3.	*Cluster server* mampu berada pada batas aman penggunaan *resource* yakni 80%

Untuk pengujian kinerja sistem, dilakukan langkah-langkah pengujian dengan skenario seperti tampak dalam Tabel 3.

Tabel 3. Skenario Pengujian

No	Jenis Pengujian	Tujuan Pengujian	Cara Melakukan Pengujian
1	*Performance Testing*	Mendapatkan data utilitas sistem yakni penggunaan CPU dan Memori	Melakukan *request* dengan total koneksi incremental pada data dengan jumlah tertentu
2	*Black Box Testing*	Mendapatkan niai kavalidan terhadap kebutuhan fungsional dan non fungsional	Melakukan *shutdown* terhadap 1 atau lebih perangkat untuk menguji fungsional dan non fungsional
3	Menghitung *Availabilitas*	Mendapatkan persentase tingkat availabilitas selama sistem berjalan	Memantau sistem yang telah berjalan pada periode tertentu

5. PEMBAHASAN

Implementasi sistem sesuai dengan rancangan yang diajukan telah berhasil dilakukan. Untuk menguji kinerja sistem, pengujian dilakukan sesuai dengan skenario pada tabel 3.

Pada hasil pengujian 1 perangkat didapatkan hasil yang bervariasi mulai dari 71 % penggunaan memori sampai 93%. Penggunaan paling optimal untuk 1 perangkat non cluster adalah pada total 20 koneksi. Rekapitulasi hasil pengujian tanpa *load balancing* ditampilkan pada Tabel 4.

Tabel 4. Hasil Pengujian Tanpa Load Balancing

No	Total Koneksi	Total Konkurensi	Jumlah Data pada Basis Data	CPU Usage (%)	Memory Usage (%)
1	10	10	7000	43.1	71
2	20	20	7000	48.6	79
3	30	30	7000	51.3	82
4	40	40	7000	53.6	87
5	50	50	7000	57.4	93

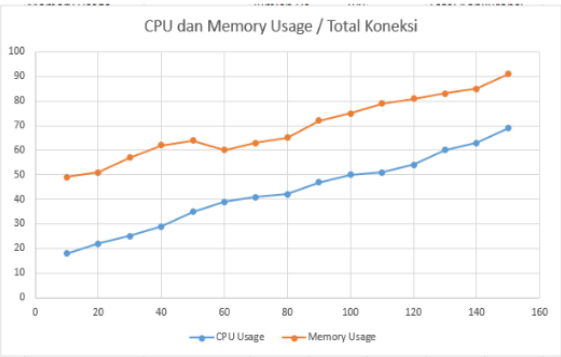
Pengujian tersebut dimulai dengan menguji 10 koneksi secara konkuren dan didapatkan hasil dari penggunaan resource yakni 18.5% CPU dan 49% Memori. pengujian dilakukan dengan kelipatan 10. pada pengujian ke 5 dengan total 41koneksi 50 cluster server ini menggunakan 35.5% CPU dan 64% RAM. Pengujian diteruskan sampai dengan total koneksi yakni 150 dengan penggunaan CPU mencapai 69.7% dan Memori 91% jelas dikatakan sudah melewati batas aman. Penggunaan paling ideal adalah pada total koneksi 110 dengan penggunaan CPU sebesar 51.3 dan memori 79% yang belum menembus batas aman penggunaan perangkat yakni 80%. Dari hasil tersebut terlihat kecenderungan yakni semakin banyak total koneksi dan konkurensi maka akan semakin tinggi pula

penggunaan resource CPU dan memori. Hasil pengujian ditunjukkan dalam Tabel 5.

Tabel 5. Hasil Pengujian dengan Cluter System

No	Total Koneksi	Total Konkurensi	Jumlah Data pada Basis Data	CPU Usage (%)	Memory Usage (%)
1	10	10	7000	18.5	49
2	20	20	7000	22.3	51
3	30	30	7000	25.2	57
4	40	40	7000	29.1	62
5	50	50	7000	35.5	64
6	60	60	7000	39.8	60
7	70	70	7000	41.3	63
8	80	80	7000	42.6	65
9	90	90	7000	47.7	72
10	100	100	7000	50.1	75
11	110	110	7000	51.3	79
12	120	120	7000	54.7	81
13	130	130	7000	60.3	83
14	140	140	7000	63.3	85

Adapun ilustrasi grafik dari hasil pengujian kinerja single-board computer dalam bentuk single server dan cluster server ditunjukkan dalam Gambar 3.



Gambar 3. Hasil Pengujian Sistem

Pengujian tingkat availabilitas dilakukan selama 112 hari dengan mendapatkan nilai availability sebesar 95.83 %.

6. KESIMPULAN

Setelah menguji sistem yang diusulkan, dapat disimpulkan bahwa single board computer mampu melakukan tugas sebagai point pengumpulan data pada organisasi sederhana seperti UMKM dengan tingkat aktifitas transaksi yang cukup tinggi. Arsitektur yang dikembangkan berbasis *high availability* web cluster mampu menjembatani ketidak seimbangan load dari server. Arsitektur yang diajukan berhasil medapatkan tingkat availability 95.83%

DAFTAR PUSTAKA

DATA, M., KARTIKASARI, D. P. & BHAWIYUGA, A., 2019. *The Design of high availability Dynamic web server cluster.* Lombok, IEEE xplore.

GUDIU, A., VOISAN, E. & DRAGAN, F., 2010. *Database replication driven communication model for distributed dedicated web hosting systems.* Timisoara, Romania, IEEE xplore.

Kementerian Hukum dan HAM Republik Indonesia, 2012. *Undang Undang No. 18 tahun 2012 tentang Pangan.* s.l.:s.n.

Kementerian Hukum dan HAM Republik Indonesia, 2015. *Undang Undang No 17 tahun 2015 tentang Ketahanan Pangan dan Gizi.* s.l.:s.n.

LI, B., SHANG, J. T., DONG, M. M. & HE, Y. L., 2020. *Research and Application of Server Cluster load balancing Technology.* Chongqing, China, s.n.

MADURANGA, M. & RAGEL, R., 2016. *Comparison of load balancing Methods for Raspberry –Pi Clustered Embedded Web Servers.* Chiangmai, Thailand, IEEE xplore.

MOON, J. B. & CHO, Y. Y., 2004. *A High-Availability Webserver Cluster Using Multiple Front-End.* s.l., Springer, Berlin, Heidelberg, pp. 752-761.

POHANKA, T. & PECHANEC, V., 2020. Evaluation of Replication Mechanisms on Selected Database Systems. *International Journal of Geo-Information,* 9(4).

SHARMA, D., 2017. *Framework for Achieving load balancing in Web Clusters based on Load Factor.* Gurgaon India, IEEE xplore.

YEO, C. S. dkk., 2006. Cluster Computing: High-Performance, High-Availability, and High-Throughput Processing on a Network of Computers. In: *Handbook of Nature-Inspired and Innovative Computin.* Boston, MA: Springer, pp. 521-551.