

REPLICATION OF COMPARATIVE METHODS FOR SINGLE DATABASE PERFORMANCE IN CENTOS OPERATION SYSTEM WITH MAXSCALE

Merinda Lestandy^{1*}, Mochammad Arif Hidayatulloh², Amrul Faruq²,
Machmud Effendy², M. Irfan²

¹Program Studi D3 Teknik Elektro, Fakultas Teknik, Universitas
Muhammadiyah Malang

²Teknik Elektro, Fakultas Teknik, Universitas Muhammadiyah Malang

E-mail: merindalestandy@umm.ac.id, arief.adx244@gmail.com,
faruq@umm.ac.id, machmud@umm.ac.id, irfan@umm.ac.id

Abstract

Processing data in an institution or organization has become an activity or routine for each meeting. In processing data, of course there are various kinds of problems, ranging from missing data, either because of an error in entering data or the result of a system failure. To overcome these problems, in an agency or company must have data backup, so as to create problems that occur in the data, the institution still has data backup. To call, another method of database replication is used. From the test results obtained data on the value of RTA (Round Trip Average) that is the time required for one time sending and receiving data from the replication database is 5426.22 ms and for compilation service diagrams a query attack occurs with a percentage of 29% in the database applications that produce 1774.45 ms results and for service diagrams at a query attack percentage of 29% a single database estimates the server down.

Keywords: *Database, Maxscale, Replication*

Abstrak

Pengolahan data dalam suatu instansi ataupun organisasi sudah menjadi suatu kegiatan ataupun rutinitas setiap harinya. Dalam melakukan pengolahan data tentu banyak ditemui berbagai macam masalah, mulai dari adanya kehilangan data, baik karena kelalaian dalam memasukkan suatu data ataupun akibat dari suatu kegagalan system. Untuk mengatasi masalah tersebut maka dalam suatu instansi ataupun perusahaan harus mempunyai cadangan data, sehingga ketika terjadi suatu masalah pada data, instansi tersebut masih mempunyai cadangan data lain. Untuk itulah digunakan yang namanya metode replikasi database yang lain. Dari hasil pengujian diperoleh data bahwa nilai RTA (Round Trip Average) yaitu waktu yang dibutuhkan untuk 1x pengiriman dan penerimaan data dari replikasi database adalah sebesar 5426.22 ms dan untuk diagram service ketika terjadi serangan query dengan prosentase 29% pada replikasi database menunjukkan hasil 1774.45 ms dan untuk diagram service pada prosentase serangan query sebesar 29% single database menunjukkan terjadinya server down.

Kata Kunci: *Database, Maxscale, Replikasi*

1. Pendahuluan

Pada masa kemajuan teknologi yang setiap harinya terus menerus mengalami peningkatan, semua pekerjaan yang dilakukan oleh manusia dituntut untuk bisa diselesaikan dengan cepat. Namun, data ataupun objek yang diolah juga semakin banyak salah satunya adalah pengolahan data pada suatu perusahaan ataupun suatu terhubung dan terstruktur sebagaimana semestinya dengan tujuan data yang ada tersebut dapat diolah, digunakan, dan cepat untuk dicari [1]. Tidak hanya terdapat kumpulan data, tetapi isi lain dari database juga bisa terdapat metadata [2]. Terdapat beberapa hal yang dapat dijadikan alasan dalam membangun database, seperti pemakaian bersama (*share*), kehandalan (*reliability*), ketersediaan (*availability*) dan kecepatan pemrosesan query [3]. Salah satu contoh aplikasi database adalah *MySQL*, *Oracle*, *MariaDB* dan lain sebagainya [4]. Perancangan database bertujuan untuk melengkapi informasi kebutuhan pengguna [5][6]. Pengolahan database juga ada berbagai macam jenisnya, salah satunya adalah *MariaDB*.

MariaDB merupakan salah satu database yang mempunyai sifat *open source* (gratis) dan dikerjakan oleh perusahaan *developer* yang sama dengan salah satu aplikasi database populer yakni *MySQL* [7]. Database *MySQL* termasuk kedalam aplikasi database yang bisa dibilang populer untuk digunakan hingga pada akhirnya sekarang telah diambil alih oleh salah satu perusahaan yang lingkup kerjanya berada pada proses pengembangan aplikasi yakni *Oracle* [7],[8]. Salah satu kelebihan dari *MariaDB* adalah karena performanya yang cukup bagus dan tidak berat serta kompatibel dengan *MySQL*. *MySQL* merupakan salah satu perangkat lunak database yang memiliki fungsi untuk melakukan pengolahan berdasarkan pada hubungan antar data ataupun berdasarkan pada objek [9]. *MariaDB* juga kompatibel dengan berbagai macam platform seperti *Linux*, *Windows*, *MacOS*, *Free BSD* dan *Solaris* [10]. Untuk bisa memasang *MariaDB* tentunya dibutuhkan suatu sistem operasi yang nantinya akan digunakan untuk membangun *server* database *MariaDB*.

Dari beberapa *platform* tersebut juga masih memiliki beberapa macam jenis. Salah satunya dari *platform Linux* terdapat beberapa jenis diantaranya adalah *Suse*, *Ubuntu*, *Debian*, *CentOS* dan lain sebagainya. *Linux CentOS (Community Enterprise Operating System)* adalah *Linux* kelas *enterprise* yang tersedia secara bebas, sistem *open source* yang bersumber dari *Linux* jenis *Red Hat Enterprise Linux (RHEL)* dan dikembangkan oleh proyek *CentOS*. *CentOS* proyek adalah lebih dari sekedar sebuah distribusi *Linux* [11][12].

Terdapat berbagai macam cara yang bisa digunakan untuk pengolahan *server* database dalam membantu meningkatkan kecepatan pengolahan suatu database, salah satunya adalah teknik replikasi database. Replikasi database sendiri merupakan suatu metode penggandaan serta suatu cara dalam pendistribusian data maupun objek database yang bersumber dari salah satu atau beberapa database yang didistribusikan ke salah satu ataupun beberapa database lain dengan melakukan penyetaraan antar database sehingga dapat dijamin antara keakuratan dan kesamaan data. Replikasi database dapat menyimpan setiap data secara berkala untuk melakukan *backup* di computer *slave* [13]. Kemampuan yang lain dari metode replikasi adalah mendukung proses kerja aplikasi, pendistribusian data fisik yang sesuai dengan setiap penggunaannya, contohnya dalam proses transaksi *online* ataupun *DSS (Decision Support System)* atau juga dalam proses pendistribusian database pada beberapa server. Adapun keuntungan penggunaan metode replikasi tergantung pada jenis replikasi yang digunakan, akan tetapi seringkali replikasi

dapat membantu ketersediaan data di setiap waktu dan tempat data tersebut dibutuhkan [14]. Teknik replikasi mampu membentuk *cluster* database dengan waktu replikasi kurang dari 0.2 detik [15].

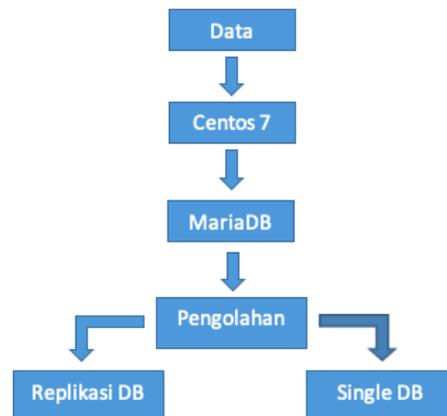
Tingkat performansi dalam proses pengolahan database dapat diukur dengan melakukan pengujian performa antara database yang menggunakan metode replikasi dengan *single* database. Hasil akhir dari pengujian tersebut akan di implementasikan dalam database Universitas Muhammadiyah Malang

Tingkat performansi dalam proses pengolahan database dapat diukur dengan melakukan pengujian performa antara database yang menggunakan metode replikasi dengan *single* database. Hasil akhir dari pengujian tersebut akan di implementasikan dalam database Universitas Muhammadiyah Malang.

2. Metode Penelitian

2.1 Desain Sistem

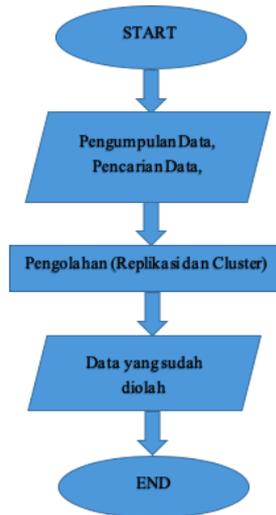
Diperlukan beberapa komponen penting didalam system pengolahan database diantaranya, centos sebagai sistem operasi, MariaDB sebagai database server, replikasi database dan *cluster* database sebagai metode pengolahan database. Pada Gambar 1. dijelaskan blok diagram sistem pengolahan database secara umum.



Gambar 1. Diagram Blok Desain Sistem

2.2 Flowchart Sistem

Dalam proses pengolahan database ini, ada beberapa tahapan yang dimulai dari pengumpulan data, pencarian data, dan permintaan data, setelah itu barulah data diolah menggunakan metode yang sudah ditentukan diawal yaitu replikasi database dan juga cluster database. Untuk lebih jelasnya dapat dilihat pada Gambar 2.



Gambar 2. Flowchart Sistem Pengolahan Database

Gambar 2. menunjukkan bagaimana sistem pengolahan database tersebut akan berhalan dari awal hingga diperoleh hasil akhir. Dimana diawali dari kumpulan data yang sudah ada, permintaan data, ataupun *query* data. Selanjutnya data tersebut akan diolah menggunakan 2 metode yaitu replikasi database dan *cluster* database dengan tujuan untuk menguji manakah yang kerjanya lebih efektif dari 2 metode tersebut. Setelah melakukan perbandingan dari kedua metode tersebut maka metode terbaik akan diterapkan pada lingkungan Universitas Muhammadiyah Malang sebagai data *warehouse*.

2.3 Flowchart Replikasi Database

Metode kedua yang digunakan dalam pengolahan database adalah replikasi database. Di dalam replikasi database juga terdapat beberapa database didalamnya, dimana database tersebut ada yang berfungsi sebagai *master* (utama) dan ada juga yang berfungsi sebagai *slave* (kedua), seperti yang dijelaskan pada Gambar 3.

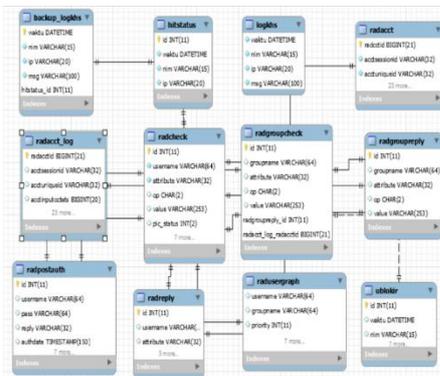


Gambar 3. Flowchart Replikasi Database

Prinsip kerja dari replikasi database adalah data ketika ada *query* dari pengguna (*user*), permintaan *query* tersebut akan diolah oleh replikasi database yang didalamnya terdapat *master* database dan *slave* database. Sesuai dengan Gambar 3. ketika mendapat perintah untuk melakukan *query* maka data akan diolah menggunakan metode replikasi database. Dimana database *master* akan mencatat perubahan perintah yang diterima pada *binary log*. Kemudian database *slave* akan menyalin aktivitas dari *binary log* kedalam *relay log* yang ada pada database *slave*. Di mulai dengan susunan pekerja, yang disebut *thread slave I/O*. *Thread I/O* membuka koneksi klien agar bisa terhubung ke *master*. Setelah itu *slave* akan merespon proses *relay log* dan selanjutnya mengambil semua data di *master*, dan *thread* ini diam menunggu *master* memberi sinyal lagi ketika ada events baru. Lalu *thread I/O* ini mencatat aktivitas ke *relay log slave*. *Thread* ini membaca dan merespon aktivitas dari *relay log*, sehingga memperbarui data *slave* agar sesuai dengan *master*.

2.4 Desain Database

Pada Gambar 4. merupakan modelan dari database radius yang nantinya akan direplikasi pada percobaan yang akan dilakukan. Pada percobaan yang akan dilakukan tabel yang akan direplikasi yaitu sebanyak 4 (empat) tabel. Tabel tersebut merupakan tabel data penting yang harus dilakukan replikasi dengan tujuan apabila terdapat kerusakan data pada database *master*, maka masih dapat di *backup* pada database *slave*.



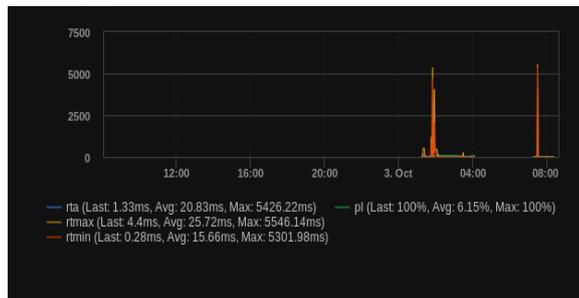
Gambar 4. Relasi antar table

2.5 Standar Performansi

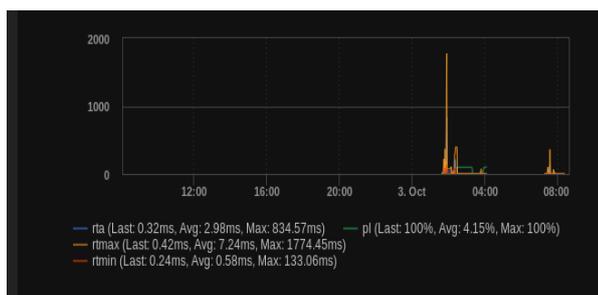
Pada suatu kegiatan pengujian performa tentunya ada beberapa titik yang dijadikan sebagai acuan untuk standar. Dalam pengujian kali ini ada beberapa acuan yang digunakan untuk sebagai standar performansi tentunya dengan menggunakan spesifikasi laptop yang sama dari segi prosesor, RAM, sistem operasi dan aplikasi yang berjalan. Yang pertama adalah dari segi RTA (*Round Trip Average*) merupakan waktu yang dicapai pada saat satu kali pengiriman dan penerimaan data dari *server*. Kedua adalah dari presentase *service* ketika berjalannya proses *query* pada suatu database *server*, yang dapat mengakibatkan terjadinya *server down*.

3. Hasil dan Pembahasan

Setelah dilakukan konfigurasi, maka dilakukan pengujian antara database replikasi dengan single database dan diperoleh hasil pengujian berupa grafik, serta diagram seperti pada Gambar 5.



Gambar 5. Grafik Keluaran Replikasi Database



Gambar 6. Grafik Keluaran Single Database

Dari hasil keluaran grafik pada *Nagiosxi* saat melakukan *query* database dapat dilihat pada Gambar 5 bahwa pada replikasi database tingkat *RTAmax* (*Round Trip Average*) yang dicapai adalah 5426.22ms, *RTA* adalah merupakan waktu yang dicapai pada saat melakukan pengiriman dan penerimaan data dari *server*. Sedangkan pada Gambar 6 dapat dilihat bahwa ketika *single* database melakukan *query* hasil *RTAmax* yang dicapai pada saat melakukan *query* adalah 1774.45ms, dimana hasil tersebut lebih rendah dibandingkan dari database yang terreplikasi. Dengan database yang digunakan sama yakni radius dan juga melakukan *query* yang sama hasil yang didapatkan berbeda, jauh hal ini dikarenakan untuk *server* database yang menggunakan replikasi dalam melakukan *query* karena terdapat 2 database didalamnya yakni *master* database dan *slave* database, yang fungsinya sendiri digunakan adalah sebagai penggandaan dari database yang ada. Dari adanya 2 database *server* inilah yang menyebabkan proses pencarian *query* pada replikasi database membutuhkan waktu yang lebih lama sehingga tingkat *RTAmax* yang dicapai lebih tinggi daripada *single* database. Karena untuk melakukan pencarian sebuah *query* dilakukan pada kedua database yakni *master* dan *slave*, dan hal ini dilakukan secara bergantian sehingga waktu yang dibutuhkan juga lama. Berbeda dengan yang terjadi pada *single* database, *RTAmax* yang dicapai oleh *single* database hasilnya lebih rendah dikarenakan pada saat terjadi pencarian sebuah *query*, *single* database hanya melakukan pencarian pada satu database saja, sehingga *RTA* yang dibutuhkan tidak terlalu tinggi. Jadi dapat disimpulkan dari hasil *RTA* yang ada pada grafik, *single* database hasilnya berada 1 tingkat lebih unggul daripada replikasi database, karena *RTA* dari *single* database nilainya lebih rendah dibandingkan *RTA* dari replikasi database.



Gambar 7. Diagram Service dari Replikasi Database



Gambar 8. Diagram Service Single Database

Dari Gambar 7 yang merupakan hasil diagram dari replikasi database dapat dilihat bahwa kemungkinan terjadinya *server down* masih dapat terjadi dengan presentase sebesar 16,68%, tetapi masih dapat mengatasi tingkat serangan *query* sebesar 29% dengan lancar, tanpa adanya pemberitahuan bahaya. Sedangkan pada Gambar 8 *single* database dapat dilihat bahwa performa ketika berjalan hampir tidak pernah mengalami *server down* 0.00%, walaupun menghadapi tingkat serangan *query* sebesar 29% yang sama dengan database yang sudah menerapkan replikasi.

Diluar dari RTA, untuk tingkat keilangan data jika terjadi kerusakan ataupun *server down* sudah dapat dipastikan bahwa pada *single* database, tidak dapat diselamatkan karena hanya terdiri dari satu database saja sehingga tidak ada database lain yang menggantikan tugas dari database tersebut ketika terjadi kerusakan, berbeda dengan database yang sudah tereplikasi ketika terjadi kesalahan ataupun *server down* dari salah satu database (*master*), maka database yang lain (*slave*) akan bisa digunakan karena sebelumnya sudah terdapat replikasi dari database utama. Berikut adalah bentuk pengujian dari sistem replikasi. Pengujiannya dilakukan dengan cara ditambahkan database baru pada sisi *master*, dan dicek pada sisi *slave* apakah juga terbentuk database baru sesuai pada *master* atau tidak.

```

MariaDB [(none)]> show databases
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.03 sec)
        
```

(a)

```

MariaDB [(none)]> show databases:
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)
        
```

(b)

Gambar 9. (a) Database Master Sebelum Replikasi (b) Database Slave Sebelum Replikasi



Gambar 10. (a) Database Master Setelah Replikasi (b) Database Slave Setelah Replikasi

Dapat dilihat pada Gambar 9. menjelaskan keadaan dari database *master* dan *slave* ketika belum direplikasi. Setelah dilakukan proses replikasi, ketika pada sisi *master* dibuat database baru maka pada sisi juga akan otomatis terbentuk database yang sama seperti pada sisi master seperti pada Gambar 10. Apabila hal ini terjadi dapat diartikan bahwa metode replikasi yang dibuat sudah berhasil berjalan. Dengan diterapkannya sistem replikasi ini, mampu mengurangi terjadinya kehilangan data apabila salah satu database mengalami kerusakan, dan tidak perlu melakukan penggandaan secara manual karena sudah otomatis tergandakan dengan menggunakan metode replikasi. Sehingga dapat memberikan kemudahan kepada pengguna untuk menggandakan database pada suatu instansi ataupun perusahaan berukuran cukup besar karena tidak perlu melakukan penggandaan secara manual yang mungkin saja bisa terjadi kelalaian atau *human error*. Pada tabel 1 dijelaskan perbedaan performa dari hasil pengukuran menggunakan *NagiosXI* ketika dilakukan *query* terhadap replikasi database dan *single* database.

Tabel 1. Hasil Perbandingan Performa Replikasi Database dan Single Database

Hasil Pengujian	Replikasi Database	Single Database
Round Trip Max	5545.24 ms	1774.45 ms
Round Trip Min	5301.98 ms	133.06 ms
Critical	29 %	29 %
Server Up	83.134 %	100 %
Server Down	16.866 %	0 %

4. Kesimpulan

Dari penelitian yang dilakukan adapun beberapa hasil yang sudah didapat dan bisa disimpulkan pada penelitian tentang perbandingan performa dari replikasi database dan *single* database ini adalah dari segi performa antara database tereplikasi dengan *single* database dapat dilihat bahwa, dari faktor RTA (*Round Trip Average*) *single* database menunjukkan hasil 1774.45ms dan untuk diagram *service* pada presentase serangan *query* sebesar 29%, *single* database tidak menunjukkan terjadinya *server down*. Sedangkan pada replikasi database RTA yang diperoleh menunjukkan hasil yang beroleh berada satu tingkat di atas *single* database yakni sebesar 5426.22ms dan untuk diagram *service* ketika terjadi serangan *query* dengan presentase 29%, pada replikasi database mengalami *server down* dengan presentase 16,68%.

Dengan menggunakan metode replikasi database, database yang direplikasi akan otomatis tergandakan pada database lain sehingga dapat mengurangi tingkat terjadinya kehilangan data akibat dari *human error* untuk melakukan penggandaan secara manual, disisi lain juga lebih menghemat waktu bagi para pekerja dan memudahkan pekerja dalam mengatasi masalah ketika terjadi kerusakan data, karena pada *server* database yang lain masih tersimpan hasil penggandaan dari database utama yang telah direplikasi.

Referensi

- [1] R. Ramakrishnan and J. Gehrke, "Database Management Systems solution," *Database Manag. Syst. Solut. Man. Third Ed.*, vol. 8, no. 4, pp. 65–68, 2003, doi: 10.1300/J115v06n04_07.
- [2] Gat, "Perancangan Basis Data Perputakaan Sekolah dengan Menerapkan Model Data Relasional," *Citec*, vol. 2, No. 4, pp. 304–315, 2015.
- [3] J. Grov, L. Soares, A. Correia, J. Pereira, R. Oliveira, and F. Pedone, "A pragmatic protocol for database replication in interconnected clusters," *Proc. - 12th Pacific Rim Int. Symp. Dependable Comput. PRDC 2006*, no. iii, pp. 230–237, 2006, doi: 10.1109/PRDC.2006.11.
- [4] A. Rahmatulloh and F. MSN, "Implementasi Load Balancing Web Server menggunakan Haproxy dan Sinkronisasi File pada Sistem Informasi Akademik Universitas Siliwangi," *J. Nas. Teknol. dan Sist. Inf.*, vol. 3, no. 2, pp. 241–248, 2017, doi: 10.25077/teknosi.v3i2.2017.241-248.
- [5] S. Khotijah, "DESAIN DATABASE SISTEM INFORMASI AKADEMIK PADA Program Studi Teknik Informatika Fakultas Teknik , Matematika dan Ilmu Pengetahuan Alam Universitas Indraprasta PGRI," *Khotijah –Desain Database Sist. Inf.*, vol. 9, no. 2, pp. 154–165, 2016.
- [6] S. Sapate and M. Ramteke, "Survey on Comparative Analysis of Database Replication Techniques," *Int. J. IT, Eng. Appl. Sci. Res.*, vol. 2, no. 3, pp. 2319–4413, 2013.
- [7] M. Data, G. Ramadhan, and K. Amron, "Analisis Availabilitas dan Reliabilitas Multi-Master Database Server Dengan State Snapshot Transfers (SST) Jenis Rsync Pada MariaDB Galera Cluster," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 1, p. 69, 2017, doi: 10.25126/jtiik.201741288.
- [8] A. Mubarak, A. Lawi, and M. Niswar, "Sistem Replikasi Basisdata Terdistribusi Untuk Data Center," pp. 1–14, 2012.
- [9] S. Sucipto, "Perancangan Active Database System pada Sistem Informasi Pelayanan Harga Pasar," *Intensif*, vol. 1, no. 1, p. 35, 2017, doi: 10.29407/intensif.v1i1.562.
- [10] I. WARMAN and R. RAMDANIANSYAH, "ANALISIS PERBANDINGAN KINERJA QUERY DATABASE MANAGEMENT SYSTEM (DBMS) ANTARA MySQL 5.7.16 DAN MARIADB 10.1," *J. Teknoif*, vol. 6, no. 1, pp. 32–41, 2018, doi: 10.21063/jtif.2018.v6.1.32-41.
- [11] S. P. Ramadhan and Teknik, "PERANCANGAN DAN IMPLEMENTASI SISTEM MONITORING JARINGAN MENGGUNAKAN ICINGA DAN POWERDNS DI PT . PACIFIC TELEMATIKA INDONESIA Selamat Puji Ramadhan Teknik Informatika – Universitas Komputer Indonesia," p. 10111381, 2016.
- [12] A. Wicitra, D. Utomo, and H. K. Wardana, "Membangun Infrastruktur Komputasi Awan Privat Single Cluster dan Multi Cluster dengan menggunakan Linux Centos," pp. 185–194.
- [13] M. H. Darmawan, Isnawaty, and Subardin, "Perancangan dan Implementasi Sistem Replikasi Database Terdistribusi pada Fakultas Teknik Universitas Halu Oleo," *semanTIK*, vol. 4, no. 2, pp. 91–98, 2018.
- [14] H. Maulana, "Analisis Dan Perancangan Sistem Replikasi Database Mysql

- Dengan Menggunakan Vmware Pada Sistem Operasi Open Source,” *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 1, no. 1, pp. 32–37, 2016, doi: 10.30743/infotekjar.v1i1.37.
- [15] A. Heryanto and A. Albert, “Implementasi Sistem Database Terdistribusi Dengan Metode Multi-Master Database Replication,” *J. Media Inform. Budidarma*, vol. 3, no. 1, p. 30, 2019, doi: 10.30865/mib.v3i1.1098.