

LANDREAU



1. **Introduction**
 2. **Background**
 3. **Methodology**
 4. **Results**
 5. **Conclusion**
 6. **References**

PERMISSION TO REPRODUCE THIS
MANUSCRIPT IS GRANTED BY THE
AMERICAN SOCIETY OF CLIMATE ENGINEERS

3. **Stressbewusstsein** ist ein zentraler Bestandteil der Stressbewältigung. Es ist die Fähigkeit, die eigenen Stressreaktionen zu erkennen und zu steuern. Dies ist wichtig, um die eigene Gesundheit zu schützen und die Belastbarkeit zu erhöhen.

1. [Friedman, M. \(1952\). "The Case for a Free Market in Money." *Journal of Political Economy*, 60\(2\), 161-178.](#)
 2. [Mises, L. \(1962\). "The Case for a Free Market in Money." *Journal of Political Economy*, 70\(2\), 161-178.](#)
 3. [Mises, L. \(1962\). "The Case for a Free Market in Money." *Journal of Political Economy*, 70\(2\), 161-178.](#)
 4. [Mises, L. \(1962\). "The Case for a Free Market in Money." *Journal of Political Economy*, 70\(2\), 161-178.](#)
 5. [Mises, L. \(1962\). "The Case for a Free Market in Money." *Journal of Political Economy*, 70\(2\), 161-178.](#)
 6. [Mises, L. \(1962\). "The Case for a Free Market in Money." *Journal of Political Economy*, 70\(2\), 161-178.](#)

[illegible]

PROCEEDINGS

For more information, contact: info@thejournal.com or call 1-800-368-6868. The Journal is a member of the American Psychological Association.

[illegible]

Keywords: *depression, self-esteem, self-efficacy, coping, social support, depression, self-esteem, self-efficacy, coping, social support*

141110a. *Agave americana* (L.) (Agavaceae) - a large, succulent-leaved plant, cultivated as an ornamental, and also as a source of fiber and for medicinal purposes. It is native to Central America and the Caribbean.

1. **Introduction**
 2. **Background**
 3. **Methodology**
 4. **Results**
 5. **Conclusion**
 6. **References**

[illegible]

1. <http://www.fishbase.org> (visited, 2009)



LEMBAR KONSULTASI

Nama : AGILAK KANISWARA
Nim : 221420100
Jurusan : Teknik Informatika
Judul : Implementasi Gallery Cluster Menggunakan RDBMS Database Sebagai Koneksi Base dan Tools Database Dengan Metode Master Slave Replication
Penitibimbing : Alif Wijaya

No	Tanggal	Keterangan	Paraf
1	25 Oktober 2023	Proposal judul	A
2	27 November 2023	Revisi proposal dan daftar Pustaka	A
3	28/11/23	Penyusunan P.b. deskripsi	A
		Petunjuk. Pustaka	A
		Pustaka	A
		Can oja program	A

LEMBAR KONSULTASI HASIL KARYA ILMIAH

Nama : Aliflak Kurniawati
NIM : 221420109
Fakultas : Sains Teknologi
Program Studi : Teknik Informatika
Judul : IMPLEMENTASI MASTER-SLAVE SEBAGAI KONFIGURASI RACA DAN
TUNING DATABASE DENGAN METODE REPLICATION
Pembimbing : Alex Wijaya S.Kom, M.T.

No	Tanggal	Mengetahui	Paraf
1		Pengantar	
2		Pengantar	
3		Pengantar	
4		Pengantar	
5		Pengantar	
6		Pengantar	
7		Pengantar	
8/82024		Alex Wijaya S.Kom, M.T.	



Fakultas
Program Studi
Universitas Bina Dharma

Name Abdullah Khan
 NIN 231429109
 Program Studied 1st
 School Government Senior High School, Dera Ismail Khan
Government Senior High School, Dera Ismail Khan
Government Senior High School, Dera Ismail Khan

Original Author(s):
Dorothy Paul H. Johnson

Author: Abhinav Chandra

Anggota Persaud: Adnan, Adha, Adhira

Arceuthobium *Arceuthobium arborescens*

2000-2001

Abstract: **Keywords:**

[illegible]

Downloaded At: 11:53 11 September 2009



Fakultas
Program Studi
Universitas Bina Darma

Name Abdullah Khan
 NIN 231429109
 Program Studied 1st
 School Government Senior High School, Dera Ismail Khan
Government Senior High School, Dera Ismail Khan
Government Senior High School, Dera Ismail Khan

Original Author(s):
Dorothy Paul H. Johnson

Author: Abhinav Chandra

Anggota Persaud: Adnan, Adha, Adha

Arceuthobium *Arceuthobium arborescens*

2000-2001

Abstract: **Keywords:**

[illegible]

2000年12月29日

HALAMAN PENGESAHAN

IMPLEMENTASI GALERA CLUSTER MENGGUNAKAN ROOMS DATABASE SEBAGAI KONEKSI BACA DAN TULIS DATABASE DENGAN METODE MASTER SLAVE REPLICATION

AKHLAK KAMISWARA
221420108

Proposal Skripsi

Dibuat sebagai salah satu syarat untuk melakukan penelitian

Pembimbing

Alex Wijaya, S.Kom., M.IT

Palembang,
Program Studi Teknik
Informatika
Fakultas Ilmu Komputer
Universitas Duta Dharma
Ketua,

Alex Wijaya, S.Kom., M.IT

HALAMAN PERSETUJUAN

Artikel Publikasi Ilmiah berjudul "Implementasi Master-Slave Sebagai Koneksi Baca dan Tulis Database Dengan Menggunakan Metode Replication" yang disusun oleh

Nama : Alhikmah Kamirawati

NIM : 12104201108

Program Studi : Teknik Informatika

telah disetujui untuk dipublikasikan di Seminar Hasil Karya Akhir.

Mengantar
Ketua Program Studi

Alch Wijaya, S.Kom., M.T.

Penerimaan : 8 Agustus 2024

Menerima
Penulis

Alch Wijaya, S.Kom., M.T.

SURAT KETERANGAN LULUS
UJIAN SEMINAR PROPOSAL PENELITIAN SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS TEKNOLOGI
UNIVERSITAS BINA DARMA



Sumber Data
Tanggal

FRM/TA/04
1 Mei 2025 (Rev. 01)



Nama	Akhilak Kaniywar
NIM/NIDM	22142010p
Judul	Implementasi Gadget, Cloud, Storage, dan Rilis Data ke Sahaja, Koneksi Data dan Tinta Digitalisasi Dengan Metode Matriks Klasik Berbasis
Pembimbing Utama	Arik Wijaya, S.Kom., M.T.

Telah mengikuti Ujian Seminar Proposal Penelitian Skripsi Program Studi Teknik Informatika Siswa Baru (Fakultas
Sains Teknologi) Universitas Bina Dharma pada :

Hari/Tanggal
Tempat

1. Senin, 11 Desember 2023
2. Arik Wijaya, S.Kom., M.T.
3. Dewi Dharma, M.Kom.
4. Nurul Aulia Chusma, S.Kom., M.T.

dan dinyatakan dengan LAYAK untuk dilanjutkan ke tahap penelitian. Dengan ini, Universitas Bina Dharma akan dapat
memberikan SK Pembimbing Penelitian untuk melanjutkan penelitian sampai Ujian Kertas (SK) sebagai mahasiswa
tersebut. Atas perhatian dan kerjasamanya Kami mengucapkan terima kasih.

Pembimbing, 01 Agustus 2024
Program Studi Teknik Informatika
Fakultas Sains Teknologi

Arik Wijaya, S.Kom., M.T.

(Lulus secara nilai): LULUS 86.6%

HR

1. Perbaikan SKRIPSI dilakukan paling lambat 1 minggu
2. Wajib dan ujian Program

NR

Diperiksa dan ditandatangani oleh Pembimbing

1. Lulus Seminar, dibuktikan dengan surat keterangan lulus seminar yang terdapat ACC, pengantar dan Ketersetujuan (Adu)
2. Falsafah (diketahui seminar yang terdapat ACC pengantar dan Ketersetujuan (Adu))
3. Surat pernyataan telah dan pembimbing Tugas Akhir yang terdapat ACC pengantar dan Ketersetujuan (Pengantar)
4. Pengantar yang terdapat ACC dan tidak ada PPS (Pengantar)
5. Pengantar, Lulus dan BPP/PPS, Seminar dan Ketersetujuan Pembimbing (Pengantar)
6. Pengantar dan Lulus dan Pembimbing

SURAT KETERANGAN LULUS
UJIAN SARJANA SKRIPSI
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS TEKNOLOGI
UNIVERSITAS BINA DARMA



Nomor Dok : FRM/WDS-01
Tanggal 01 Mei 2006, Rev. 00

Nama Lengkap : Akhdak Kantiworo
NIM : 221420109
Judul : Implementasi Sistem Ganti Sebagai Koneksi Base Dan Tuba
Ditubahi Menggunakan Microsoft Application
Penanggung Jawab : I. Alek Wijaya, S.Kom., M.T.

Teluh mengikuti Ujian Kertas dan Ujian Akhir II Program Studi Teknik Informatika Sarjana Sains Sains
Teknologi Universitas Bina Darma pada

Tanggal : Selasa, 27 Agustus 2024

Dengan ini dinyatakan LULUS dengan nilai 87 (A) dan diberikan 40% keikutsertaan Kary
mengucapkan terima kasih.

Palembang, 09 September 2024
Program Studi Teknik Informatika
Fakultas Sains Teknologi

Alek Wijaya, S.Kom., M.T.

NB

1. Surat untuk mendaftar Winda
2. Informasi Pendaftaran Winda Hubungi Pusat Pelayanan Mahasiswa
3. Nanti Ditanda tangani oleh Ka. Prog. Studi



Implementasi *Master-Slave* Sebagai Koneksi Baca Dan Tulis Database Dengan Metode *Replication*

Aldhak Kamiswara^{2,1}, Alek Wijaya¹

^{1,2}Teknik Informatika, Universitas Bina Darma, Palembang, Indonesia

¹Email Penulis Korespondensi: aldhakamiswara@gmail.com

Abstrak

Kemajuan dalam komputasi *internet of things*, data yang dihasilkan mesin volume data kian meledak. Memiliki data saja tidak cukup. Seseorang perlu memiliki kemampuan untuk melakukan manajemen database melibatkan keterampilan dan pengetahuan yang diperlukan untuk merancang, mengimplementasikan dan memelihara sistem database. Bagaimana penerapan database dengan metode *master-slave replication* sebagai koneksi tulis dan baca sehingga dapat membantu untuk memisahkan proses koneksi saat melakukan penulisan dan pembacaan data sebagai pembebanan kerja server dan meningkatkan *execute time* pada client. Sehingga dapat mengetahui hasil dari pengujian metode *master-slave replication* sebagai koneksi baca dan tulis database dan membandingkan hasil tersebut dengan metode *master-only* sebagai koneksi baca dan tulis database. Pada bentuk *master-slave* penulis membuat master sebagai server yang aktif melakukan segala bentuk penulisan data metode *HTTP POST* dan slave yang *standby* akan mendapatkan data yang didapatkan hasil dari replikasi master digunakan untuk proses pembacaan data metode *HTTP GET* ini membuat proses penulisan data ke database tidak akan terganggu oleh proses pembacaan data ke database sehingga *CPU Usage* dan *Memory Usage* server akan terjaga dan meningkatkan *execution time* dan *total request*. Dengan adanya metode *master-slave replication* ini akan membantu mengurangi persaingan untuk resource dan memungkinkan aplikasi untuk melakukan lebih banyak operasi baca dan tulis secara paralel.

Kata kunci—Database MariaDB, *Master-Slave Replication*, *Execution Time*, *Total Request*

Abstract

The rapid advancements in computing, IoT, machine-generated data, and overall data volumes have created an unprecedented demand for efficient data management. Merely possessing data is insufficient; the capability to effectively manage databases, encompassing skills and knowledge in design, implementation, and maintenance, is crucial. This study investigates how implementing a *master-slave replication* method as a write and read connection can enhance database performance. By separating write and read operations, this method aims to balance the workload, reduce server load, and improve client-side execution time. The research compares the performance of the *master-slave replication* method with a traditional *master-only* approach. In the *master-slave* setup, the master server handles all write operations using *HTTP POST*, while the slave server, which receives replicated data from the master, manages read operations using *HTTP GET*. This separation ensures that write operations are not hindered by

read operations, optimizing CPU and memory usage, and consequently improving execution time and overall request handling. Ultimately, master-slave replication mitigates resource contention, enabling applications to execute more read and write operations concurrently.

Keywords—Database, MariaDB, Master-Slave Replication, Execution Time, Total Request

1. PENDAHULUAN

Data dalam istilah *the next new oil* merupakan sebuah aset yang berharga. Ibarat sumber daya minyak, bagi mereka yang melihat nilai fundamental data dan belajar mengekstrak serta menggunakannya ada imbahnya yang sangat besar. Kita berada dalam era *data driven* dimana data menjadi lebih berharga. Namun jika tidak diokeh, maka data tersebut tidak dapat digunakan.

Jadi, data harus dipecah, dianalisis agar memiliki nilai. Karena kemajuan dalam komputasi, *internet of things*, data yang dihasilkan mesin, volume data kini meledak. Memiliki data saja tidak cukup. Seseorang perlu memiliki kemampuan untuk melakukan manajemen *database* melibatkan keterampilan dan pengetahuan yang diperlukan untuk merancang, mengimplementasikan, dan memelihara sistem *database* [1],[2]. Dalam melakukan proses penulisan dan pembacaan data ke *database* diperlukan metode tertentu untuk melakukan proses tersebut seperti menggunakan metode protokol *HTTP POST* untuk mengirimkan data dan metode *protocol HTTP GET* untuk membaca data antara *client* dan *server* [3].

Master-slave replication merupakan sebuah metode penduplikasian data dimana jika terdapat data yang ditulis ke *master database*, data tersebut juga akan masuk ke *database slave* yang *standby*. Pada bentuk *master-slave* penulis membuat *master* sebagai *server* yang aktif dan akan melakukan proses penulisan data sedangkan *slave* yang *standby* akan mendapatkan data dari replikasi *master* yang digunakan untuk melakukan proses pembacaan data. Ini yang membuat proses penulisan data kehadiran tidak akan terganggu oleh proses pembacaan data ke *database* sehingga dapat meningkatkan *total request* dan *execution time* [4].

Hal ini yang membuat penulis menggunakan teknologi replikasi data dan melakukan pengujian pada *database* yang menggunakan metode *master only* dan *master-slave* lalu membandingkannya. Beberapa penelitian terdahulu yang menerapkan implementasi replikasi pada *database* diantaranya, membuat *auto promote server slave* menjadi *master* di saat *server master* mengalami *error* [5]. Kemudian penelitian [6] melakukan *load testing* untuk mendapatkan perbandingan performa antara arsitektur *monolithic* dan *microservices*. Penelitian lainnya oleh [7] implementasi *Zabbix* untuk *monitoring resource* perangkat komputer dan jaringan. Berdasarkan dari penelitian terdahulu maka dalam penelitian ini akan dilakukan implementasi *master-slave replication* untuk membagi beban proses pembacaan dan penulisan data ke *database*.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Adapun tahapan penelitian data yang digunakan penulis yaitu:

1) Analisis

Tahap awal ini dilakukan analisis kebutuhan, serta permasalahan yang muncul sehingga dilakukan tahap menganalisa untuk melakukan perancangan.

2) Design

Tahap desain ini akan membuat gambar desain topologi *server database* yang akan dibangun dan perancangan *database*.

3) Implementation

Tahapan ini akan lebih lama dari tahapan sebelumnya, dikarenakan penulis menerapkan yang telah dianalisa dan dirancang sebelumnya. Pada tahapan inilah terlihat bagaimana metode *master-slave replication database* yang akan dibangun akan memberikan perubahan dengan metode yang hanya menggunakan *database master only*.

4) *Testing dan Monitoring*

Tahapan ini akan dilakukan pengujian sistem berupa uji coba sistem yang telah dibangun, pengujian akan dilakukan pada metode *master-slave replication database* dimana *master* akan digunakan untuk koneksi tulis sedangkan *slave* akan digunakan untuk koneksi baca dan *master database only* akan digunakan untuk koneksi baca dan koneksi tulis. *Database* akan dilakukan *load testing* menggunakan aplikasi *locust* dan aplikasi absensi yang bisa melakukan penulisan data dan menampilkan hasil absensi menggunakan bahasa pemrograman *PHP* [8], [9]. Tahapan ini akan mengamati *database* yang menggunakan *master-slave replication* dan *master only* untuk melihat sumber daya *server database* seperti *CPU usage* dan *memory usage* menggunakan aplikasi *zabbix* dan melihat *response time* serta banyaknya *request* menggunakan aplikasi *locust*, sehingga dapat dibandingkan hasil dari implementasi tersebut [10].

2. 2 *Analisis*

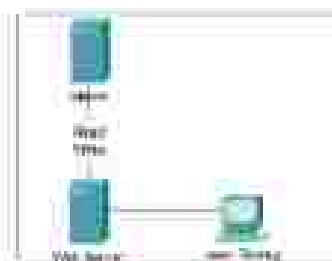
Berdasarkan pengamatan yang dilakukan, proses kegiatan absensi kehadiran biasanya memiliki waktu maksimal mengabsen sehingga absen tepat waktu menjadi sangat penting untuk catatan kehadiran seseorang. Biasanya absensi menggunakan identitas seperti nama dan nomor induk, ketepatan waktu sangat penting sehingga menyebabkan orang-orang melakukan kegiatan absensi kehadiran secara bersamaan dengan massal sehingga mempengaruhi kerja *database*, yang dapat menyebabkan turunya performa *database* dan terjadinya kegagalan dalam pencatatan data absensi.

2. 3 *Desain*

Desain merupakan gambaran dari proses kerja pengujian *database* yang akan dibuat, selain itu juga sebagai gambaran secara menyeluruh terhadap pembuatan untuk pengujian *database master only* dan *master-slave*. Tujuan dari desain ini adalah suatu desain sistem yang dilakukan dalam mengerjakan suatu sistem yang akan diujikan dan diimplementasikan dalam suatu bentuk konsep awal pengujian *database master only* dan *master-slave*. Dengan adanya desain *database* yang terkonsep maka pengujian *database master only* dan *master-slave* akan terarah.

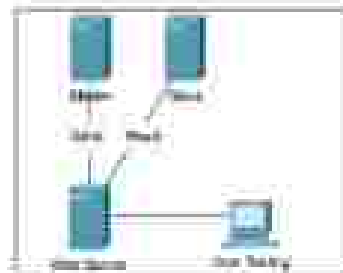
2. 3.1 *Desain Topologi Server*

Desain topologi menggambarkan bagaimana proses *data get* dan *post* terjadi dengan menggunakan metode *master only* dan *master-slave*, hal ini dapat dilihat pada Gambar 1.



Gambar 1 Topologi Master Only

Pada *database master only* proses pembacaan data dan penulisan data terjadi di *database master*, dimana rancangan arsitekturnya dapat dilihat pada Gambar 2. Pada *database master-slave* proses penulisan data terjadi di *database master* dan pembacaan data terjadi di *database slave*.



Gambar 2 Topologi Master-slave

2.3.2 Desain Database Desain

Menurut [11], proses desain database terdiri dari 3 tahap yaitu Model Konseptual, Model Logikal dan Fisik. Penulis menggunakan 3 tahap tersebut untuk mendesain database sesuai dengan kegiatan absensi kehadiran.

a. Model Konseptual

Penulis membuat desain model konseptual database berdasarkan kegiatan absensi kehadiran pada umumnya. Desain model konseptual terdiri dari entitas dan relasi. Sebelum menentukan entitas maka diperlukan identifikasi kegiatan absensi kehadiran. Penulis hanya menggunakan 1 table untuk pengujian database metode *master only* dan *master-slave* sehingga penulis tidak membuat relasi untuk pengujian ini. Entitas dapat dilihat pada Tabel 1.

Tabel 1 Tabel Entitas

Nama Entitas	Keterangan
Kehadiran	Berisi informasi mengenai data kehadiran

b. Desain Model Logikal

Tahapan ini merupakan lanjutan setelah desain model konseptual dibuat. Hasil dari identifikasi pada desain model konseptual menghasilkan informasi entitas *domain*. Adapun hasil dari identifikasi tersebut menghasilkan *table* kehadiran dengan rincian entitas dapat dilihat pada Tabel 2.

Tabel 2 Rancangan Tabel Kehadiran

Nama Tabel	Atribut <i>Primary Key</i> (*)
Kehadiran	*Id_kehadiran, badge, finger_node, waktu

Setelah melakukan proses normalisasi, selanjutnya penulis mengidentifikasi atribut, *domain* dan *primary key* pada entitas *table* kehadiran. Hasil analisis identifikasi atribut, *domain* dan *primary key* pada *table* kehadiran dapat dilihat pada Tabel 3.

Tabel 3 Identifikasi Atribut, Domain dan Primary Key

Entitas	Atribut	Domain	Primary Key
Kehadiran	id_kehadiran	<i>int</i>	id_kehadiran
	badge	<i>Varchar</i>	
	finger_node	<i>Enum</i> ('0', '1')	
	waktu	<i>datetime</i>	

c. Desain Model Fisik

Tahap akhir dari desain database adalah membuat rancangan model fisik yang bertujuan untuk menggambarkan struktur database. Desain dapat dilihat pada Gambar 3 model fisik dari database untuk *table* kehadiran.

Kehadiran
id_kehadiran (PK)
nama_kehadiran (N)
tanggal_kehadiran (N)
status_kehadiran

Gambar 3 Model Fisk Database

2.4 Implementation

Untuk melakukan pengujian database metode *master only* dan *master-slave* ini penulis menggunakan *operating system VMware vSphere Hypervisor* sebagai server virtualisasi dan *Ubuntu 22.04 LTS 64 bit* sebagai server *virtual machine*, sedangkan pada sisi database penulis menggunakan *Mariadb server* sebagai database server [12]. Pengujian database ini menggunakan *PHP* sebagai bahasa pemrograman yang digunakan.

2.4.1 Implementation API

Pada implementasi API ini, penulis membuat *script coding* menggunakan bahasa pemrograman *PHP* yang akan dihosting di web server menggunakan *Apache2* [13]. *Script coding PHP* yang telah dibuat ditempatkan pada folder */var/www/html* untuk web server *apache2* dapat membaca *script coding* tersebut sehingga dapat dihosting [14].

a. Script coding get method

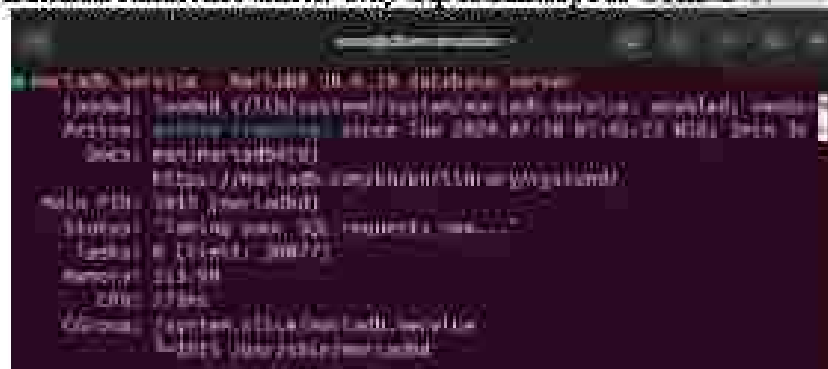
Script ini berfungsi untuk melakukan pembacaan data ke database dengan membaca tabel kehadiran dengan *limit 30 rows*.

b. Script coding post method

Script ini berfungsi untuk melakukan penulisan data ke database dan data masuk ke table kehadiran dengan menggunakan *syntax transaction* dan *mysqli_realconnect API* ini dapat mengurangi proses penulisan data jika terjadi kegagalan dalam penulisan data ke tabel.

2.5 Implementasi Master Only

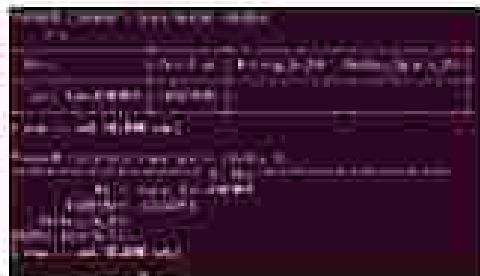
Hasil instalasi database *master only* dapat dilihat pada Gambar 4.



Gambar 4 Status MariaDB Master Only

2.6 Implementasi Master-Slave

Hasil instalasi database *master-slave replication* dapat dilihat pada Gambar 5 dan Gambar 6.



Gambar 5 Status Database Master



Gambar 6 Status Database Slave

2. Implementasi Locust

Implementasi *locust* menggunakan *script* dan *script* ini berfungsi untuk melakukan pembacaan *API* mana yang akan diuji. *Script* yang koneksi ini di beri nama *locustfile.py*. Setelah dibuat *script*nya dapat digunakan untuk melakukan *testing*.

3. HASIL DAN PEMBAHASAN

Hasil dari *Master-slave replication* membuat *master* sebagai *server* yang aktif melakukan segala bentuk penulisan data metode *HTTP POST* dan *slave* yang *standby* akan mendapatkan data yang didapatkan hasil dari replikasi *master* digunakan untuk proses pembacaan data metode *HTTP GET* ini membuat proses penulisan data kehadiran tidak akan terganggu oleh proses pembacaan data ke *database* sehingga *CPU Usage* dan *Memory Usage server* akan terjaga dan meningkatkan *execution time* dari *slave* baru, sehingga membantu mengurangi persaingan untuk sumber daya dan memungkinkan aplikasi untuk melakukan lebih banyak operasi baca dan tulis secara *parallel* [15], [16].

Pada penelitian ini juga dilakukan perbandingan pengujian menggunakan metode *master-slave replication* sebagai koneksi baca dan tulis *database* dengan metode *master only* sebagai koneksi baca dan tulis *database*.

3.1 Testing dan Monitoring

Pengujian ini menggunakan 2 juta baris *record* dengan melakukan perbandingan metode pengujian menggunakan metode *master-slave replication* dan metode *master only*. Pengujian ini juga melibatkan aplikasi *Locust* sebagai aplikasi yang digunakan untuk melakukan *load testing* tetapi dengan *API* yang berbeda.

3.1.1 Testing Master Only

Pengujian *master only* dilakukan dengan menggunakan 2 juta baris *record* pada *table* dengan menggunakan *API* untuk melakukan pengujian.

a. Struktur Tabel Master Only

Struktur *table master only* akan menampilkan *field record table* yang akan digunakan pada *master only*. Struktur *table master only* dapat dilihat pada Gambar 7.

ID	Name	Age	Height	Weight	Blood Pressure
1	John Doe	30	175	70	120/80
2	Jane Smith	25	160	55	110/70
3	Bob Johnson	45	180	85	130/90

Gambar 7 Struktur Tabel

b. *Record Table Master Only*

Record table master only yang digunakan pada *master only* menampilkan data *record* dengan menggunakan 2 juta baris *record*, seperti dapat dilihat pada Gambar 8.

ID	Name	Age	Height	Weight	Blood Pressure
1	John Doe	30	175	70	120/80
2	Jane Smith	25	160	55	110/70
3	Bob Johnson	45	180	85	130/90

Gambar 8 *Record Table Master Only*

c. *Pengujian Menggunakan Locust*

Pengujian dilakukan dengan menggunakan browser dengan URL *localhost8089*. Pengujian dilakukan menggunakan 512 pengguna selama 30 menit yang akan diarahkan ke server *API master only*.

d. *Hasil Pengujian*

Hasil Pengujian yang telah dilakukan menampilkan hasil *total request* sebanyak 57517 dengan rata-rata *response time* 12388.34 ms. Hasil dari pengujian dapat dilihat pada Gambar 9.

Test	URL	Request	Response	Success	Failure	Response Time
1	localhost8089	57517	12388.34	100%	0%	12388.34
2	localhost8089	57517	12388.34	100%	0%	12388.34
3	localhost8089	57517	12388.34	100%	0%	12388.34

Gambar 9 Hasil Pengujian *Master Only*

e. *CPU Master Only*

Penggunaan *CPU* pada server *master only* selama 30 menit menampilkan rata-rata 98,9462 % dengan maksimal penggunaan 100 % dan minimal penggunaan 68,3853 %. Hasil grafik penggunaan *CPU Usage* dapat dilihat pada Gambar 10.

Gambar 10 Grafik *CPU Usage Master Only*

f. *Memory Master Only*

Penggunaan *memory* pada server *master only* selama 30 menit menampilkan rata-rata 17,3016 % dengan maksimal penggunaan 18,1289 % dan minimal penggunaan 16,352 %. Hasil grafik penggunaan *memory usage* dapat dilihat pada Gambar 11.

Gambar 11. Grafik *Memory Usage Master Only*

3.1.2 Pengujian Master-Slave Replication

Pengujian *master-slave replication* dilakukan dengan menggunakan 2 juta baris *record* pada *table* dengan menggunakan *API* untuk melakukan pengujian.

a. Struktur Table Master-Slave

Struktur *table master-slave* akan menampilkan *field record table* yang akan digunakan pada *master-slave*, seperti dapat dilihat pada Gambar 12.

```

mysql> show tables;
+-----+
| table |
+-----+
| table |
+-----+

mysql> describe table;
+-----+
| table |
+-----+
| id    | int(11) | NO | PRI | NULL | UNSIGNED |
| name  | varchar(255) | YES | | NULL | |
| email | varchar(255) | YES | | NULL | |
| phone | varchar(255) | YES | | NULL | |
| address | varchar(255) | YES | | NULL | |
+-----+

```

Gambar 12. Struktur Table Master-Slave

b. Record Table Master-Slave

Record table master-slave yang digunakan pada *master-slave* menampilkan data *record* dengan menggunakan 2 juta baris *record* seperti dapat dilihat pada Gambar 13.

```

mysql> show tables;
+-----+
| table |
+-----+
| table |
+-----+

mysql> describe table;
+-----+
| table |
+-----+
| id    | int(11) | NO | PRI | NULL | UNSIGNED |
| name  | varchar(255) | YES | | NULL | |
| email | varchar(255) | YES | | NULL | |
| phone | varchar(255) | YES | | NULL | |
| address | varchar(255) | YES | | NULL | |
+-----+

```

Gambar 13. Record Table Master-Slave

c. Pengujian Menggunakan Locust

Pengujian dilakukan dengan menggunakan *browser* dengan *URL localhost8089*. Pengujian dilakukan menggunakan 512 pengguna selama 30 menit yang akan diarahkan ke *server API master-slave*.

d. Hasil Pengujian

Hasil pengujian yang telah dilakukan menampilkan hasil *total request* sebanyak 146576 dengan rata-rata *response time* 5648,33 ms. Hasil dari pengujian dapat dilihat pada Gambar 14.

```

Locust - Summary
+-----+
| User | Method | URL | Status | Count | Success | Failures | Response Time |
+-----+
| GET | /api/users | 200 | 200 | 146576 | 100% | 0 | 5648.33ms |
+-----+

```

Gambar 14. Hasil Pengujian Master Only

e. CPU Master metode Master-Slave

Penggunaan CPU pada *server master only* selama 30 menit menampilkan rata-rata 1,9373% dengan maksimal penggunaan 2,4285% dan minimal penggunaan 0,1252%. Hasil grafik penggunaan CPU Usage dapat dilihat pada Gambar 15.



Gambar 15 Grafik CPU Usage Master-Slave Node Master

f. CPU Slave metode Master-Slave

Penggunaan CPU pada *server master only* selama 30 menit menampilkan rata-rata 97,0538 % dengan maksimal penggunaan 100 % dan minimal penggunaan 11,6153 %. Hasil grafik penggunaan CPU Usage dapat dilihat pada Gambar 16.



Gambar 16 Grafik CPU Usage Master-Slave Node Slave

g. Memory Master metode Master-Slave

Penggunaan *memory* pada *server master only* selama 30 menit menampilkan rata-rata 19,1504 % dengan maksimal penggunaan 19,2583% dan minimal penggunaan 18,9499%. Hasil grafik penggunaan *memory usage* dapat dilihat pada Gambar 17.



Gambar 17 Grafik Memory Usage Master-Slave Node Master

h. Memory Slave Metode Master-Slave

Penggunaan *memory* pada *server master only* selama 30 menit menampilkan rata-rata 21,9014 % dengan maksimal penggunaan 22,0583% dan minimal penggunaan 21,8303%. Hasil grafik penggunaan *memory usage* dapat dilihat pada Gambar 18.

Gambar 18 Grafik *Memory Usage Master-Slave Node Slave*

3.1.3 Perbandingan *Master Only* dan *Master-Slave*

Pengujian *master only* dan *master-slave* menghasilkan data yang dapat dibandingkan. Data yang dihasilkan dari pengujian adalah *response time*, *total requests*, *cpu utilization* dan *memory utilisation*.

a. *Response Time*

Perbandingan antara *master only* dan *master-slave* dalam kecepatan melayani request. Melihat dari hasil perbandingan pada Tabel 14 didapatkan hasil peningkatan rata-rata nilai *response time* pada saat menggunakan metode *master-slave*, performa kecepatan melayani request lebih cepat dibandingkan menggunakan *master only*.

Tabel 14 Perbandingan *Response Time*

<i>Response Time</i>	<i>Master Only</i>	<i>Master-slave</i>	Peningkatan
Rata-rata <i>Read</i>	19237.02 ms	8133.14 ms	236.52 %
Rata-rata <i>Write</i>	5455.84 ms	3152.06 ms	173.08 %
Total	12388.24 ms	5648.33 ms	219.32 %

b. *Total Request*

Perbandingan antara *master only* dan *master-slave* dalam menerima banyaknya request. Melihat dari hasil perbandingan pada Tabel 15 didapatkan hasil peningkatan banyaknya *response* yang dikirim pada saat menggunakan metode *master-slave*, performa melayani request lebih banyak dibandingkan menggunakan *master only*.

Tabel 15 Perbandingan *Total Request*

Banyak <i>Request</i>	<i>Master Only</i>	<i>Master-slave</i>	Peningkatan
<i>Read</i>	28933	74459	257.34 %
<i>Write</i>	28584	74117	259.29 %
Total	57517	148576	258.31 %

c. *CPU Utilization*

Perbandingan antara *master only* dan *master-slave* rata-rata *CPU utilization*. Melihat dari hasil perbandingan pada Tabel 6 didapatkan hasil peningkatan penggunaan CPU, saat pengujian pada *master only* terjadi peningkatan *CPU usage* di awal pengujian sebesar 68.88% sedangkan *master-slave* hanya 11.61%. Rata-rata penggunaan *master only* sebesar 98.94% dan *master-slave* sebesar 97.05% terjadi selisih 1.89% pada rata-rata penggunaan CPU. Maksimal penggunaan CPU *master only* dan *master-slave* sama-sama 100% ini menandakan bahwa *requests* yang terjadi pada database cenderung menggunakan CPU untuk melakukan proses penulisan dan pembacaan data.

Tabel 6 Perbandingan *CPU Utilization*

<i>CPU Usage</i>	<i>Master Only</i>	<i>Master-Slave</i>	<i>Slave Master-Slave</i>
<i>Min</i>	68.88 %	0.12 %	11.61 %
<i>Max</i>	100 %	2.42 %	100 %
<i>Average</i>	98.94 %	1.93 %	97.05 %

d. Memory Utilization

Perbandingan antara *master only* dan *master-slave* rata-rata *memory utilization*. Melihat dari hasil perbandingan pada Tabel 17 didapatkan hasil peningkatan penggunaan *memory*, peningkatan penggunaan *memory master only* dan *master-slave* ini menandakan bahwa *request* yang terjadi pada *database* cenderung menggunakan *memory* lebih sedikit untuk melakukan proses penulisan dan pembacaan data.

Tabel 17 Perbandingan *Memory Utilization*

<i>Memory Usage</i>	<i>Master Only</i>	<i>Master Master-slave</i>	<i>Slave Master-slave</i>
Min	16.35 %	18.94 %	21.83 %
Max	18.12 %	19.25%	22.05 %
Average	17.30 %	19.15 %	21.83 %
Peningkatan	1.77 %	0.31 %	0.22 %

4 KESIMPULAN

Adapun kesimpulan dari pengujian *database master only* dan *master-slave*, yaitu:

- Operasional *database* cenderung menggunakan CPU dibandingkan *memory*.
- Total *request* yang dapat diterima menggunakan *master-slave* meningkat dengan nilai rata-rata 258.31 % dibandingkan menggunakan *master only*.
- Response time proses *write* dan *read* yang didapat menggunakan *master-slave* meningkat dengan nilai rata-rata 219.32 % dibandingkan menggunakan *master only*.

5 SARAN

Perkembangan sistem informasi yang pesat dan memerlukan skalabilitas yang lebih besar, penggunaan *database master-slave replication* dapat dikombinasikan dengan *database sharding*. *Sharding* membagi data menjadi beberapa *subset* yang disimpan di *server* yang berbeda, sehingga meningkatkan skalabilitas dan mengurangi beban pada *master server*.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Tim Redaksi Jurnal Teknik Politeknik Negeri Sriwijaya yang telah memberikan kesempatan, sehingga artikel ilmiah ini dapat diterbitkan.

DAFTAR PUSTAKA

- [1] W. Murni Wijaya dan Z. Mutaqin Subekti, "Penerapan Aplikasi Database pada Kegiatan Manajemen Sekolah," vol. 3, no. 1, hlm. 159–167, 2019.
- [2] A. Setiawan dan W. Muthia Karsha, "Development of Cluster Database System Using Galera Cluster Application at Vocational School of IPB University," vol. 11, no. 2, hlm. 49–59, 2021, doi: 10.29244/jst.v11i2.49.
- [3] F. Huzairi, I. Gunawan, D. Cahya Purnomo, M. Yanti, dan N. Kriedayantiabede Teknik Elektro Sekolah Tinggi Teknologi Ronggolawe Cepu Pemilik Korenspondensi, "Analisis Keamanan Data Pada Website Dengan Wire shark," 2021. [Daring]. Tersedia pada <http://sttrcepu.ac.id/sinkap/login>.
- [4] M. Harbi dan N. R. Saputra, "Analisis Quality Of Service (QOS) Jaringan Internet Kantor Pusat King Bukopin dengan Menggunakan Wire shark," 2021. [Daring]. Tersedia pada <https://jurnal.uns.ac.id/index.php/just-t/index>.
- [5] E. Asriyar dan T. Sutendi, "Implementasi Sistem Replikasi Data Base Postgresql Master-Slave Repmgr dengan Auto Promote Master DB," 2019.

- [6] F. S. Redha, R. P. Sari, dan S. Rahmayuda, "Perbandingan Performa Web Services Yang Dibangun Menggunakan Arsitektur Monolitik Dan Microservices Pada Sistem Point of Sales," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, 2023.
- [7] A. B. Cahyo, T. K. Hariadi, dan Y. Ardiyanto, "Implementasi Zabbix Server untuk Memonitor Kondisi Jaringan Komputer di Dinas Komunikasi dan Informatika Kabupaten Pekalongan," 2020.
- [8] D. Intan Permatasari¹, "Pengujian Aplikasi Menggunakan Metode Load Testing dengan Apache Jmeter pada Sistem Informasi Pertanian," *Jurnal Sistem dan Teknologi Informatika*, vol. 8, 2020.
- [9] R. Hermiati, Asnawati, dan I. Kenedi, "Pembuatan E-Commerce Pada Raja Komputer Menggunakan Bahasa Pemrograman PHP dan Database MySQL," *Jurnal Media Informatika*, vol. 17, 2021.
- [10] R. Wiji Wahyuningrum dan E. Haenullah, "Analisis Monitoring Sistem Jaringan Komputer Menggunakan Aplikasi Spiceworks," *PROSSKO*, vol. 9, 2022.
- [11] R. Nur, D. Aryani, T. Elektro, dan P. Negeri Ujung Pandang, "Pemodelan Basis Data Pada Sistem Informasi Laporan Kinerja Program Studi (LKPS) Berbasis Instrumen Akreditasi Program Studi (IAPS 4.0)," 2020.
- [12] K. Christiono dan H. Sama, "Studi Komparasi Database Management System Antara Maria DB Dan PostgreSQL Terhadap Efisiensi Penggunaan Sumber Daya Komputer," 2020, [Darimg]. Tersedia pada: <http://jurnal.uib.ac.id/index.php/cbscit>
- [13] A. V. Chandra, "Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request," *Jurnal Sistem dan Informatika (JSI)*, vol. 14, no. 1, hlm. 48-56, Nov 2019, doi:10.30864/jsi.v14i1.248.
- [14] N. K. Akmal dan M. N. Dasaprawira, "Rancang Bangun Application Programming Interface (API) Menggunakan Gaya Arsitektur GraphQL Untuk Pembuatan Sistem Informasi Pendaftaran Anggota Unit Kegiatan Mahasiswa (UKM) Studi Kasus UKM Starkbs," *Jurnal Sistem Informasi dan Teknologi (SITBCH)*, 2022, [Darimg]. Tersedia pada: <http://www.jurnal.unik.ac.id/emach>
- [15] R. Farah Humainah, A. Fikri Tri Firmansyah, S. Iman Ramadhani, dan D. Aribowo, "Pengaruh Kapasitas Memori RAM (Random Access Memory) Terhadap Kecepatan Memori Pada Laptop," *Jurnal Elektronika dan Teknik Informatika Terapan (JENTIK)*, vol. 1, hlm. 178-186, 2023.
- [16] D. Andira Sulaiman, I. Nurul Syabani, M. Ashof Amia Anka, dan D. Aribowo, "Ruang Lingkup Organisasi dan Arsitektur Komputer," *Jurnal Elektronika dan Teknik Informatika Terapan (JENTIK)*, vol. 1, hlm. 164-177, 2023.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET,
DAN TEKNOLOGI

POLITEKNIK NEGERI SRIWIJAYA
JURNAL TEKNIKA

Jalan Sriwijaya Pangreh 30011 Berto, Palembang 30139, Indonesia
Telp. +62 311-733434 Fax. +62 311-733803

E-mail: jurnal.teknika@ptn-sriwi.ac.id Website: www.ptn-sriwi.ac.id/jurnal-teknika

Indonesian Journal of Engineering Technology - Jurnal Teknik Elektro - Jurnal Teknik Industri - Jurnal Teknik Kimia - Jurnal Teknik Sipil - Jurnal Teknik Mesin - Jurnal Teknik Informatika



JURNAL TEKNIKA



BUKUT KETERANGAN PENTAHAPAN JURNAL TEKNIKA

Journal of Engineering Technology
No. 01/2021/01/07/PTN-SRIWIJAYA

Dewan penyunting Jurnal PTN Sriwijaya telah menerima artikel

Nama Penulis	Achmad Kamiswara, Alif Wijaya
Judul	Implementasi Master-Slave Sebagai Kontrol Busbar Pada Sistem Tenaga Dengan Metode Regulasi
Asal Institusi	Universitas Huta Datar

Mengatakan bahwa artikel tersebut telah diterima oleh Dewan Penulisan Jurnal TEKNIKA Politeknik Negeri Sriwijaya dan akan diterbitkan pada Jurnal TEKNIKA volume 18 nomor 1 Tahun 2021 Tahun 2021. Dengan ini kami mengucapkan terima kasih kepada

Palembang, 22 September 2021

Dr. M. Miftahul Izzah, S. Kom, M. Eng

[Signature]

Indonesian



[Signature]

[Signature]

[Signature]

[Signature]



	FORMULIR Permohonan Pengajuan Judul & Pembimbing Karya Akhir	Nama Pokok	VERITA/2002
		Nama Belakang	WJ
		Tgl. Berlaku	01 Oktober 2022
		Kelas/SPK	

Perihal : **Permohonan Judul & Pembimbing Karya Akhir**

Kepada Yth.
Ketua Program Studi Teknik Informatika
Fakultas Ilmu Teknologi
Universitas Bina Dharma
Palembang

Dengan hormat,
Saya yang bertanda tangan di bawah ini, mahasiswa Program Studi Teknik Informatika
Universitas Bina Dharma Palembang.

Nama : Akhla Kartasura
Nim : 22420106
Semester : 3
Program Studi : Teknik Informatika
Kelompok Studi :

Sehubungan dengan akan berakhirnya studi saya, maka dengan ini bermaksud mengajukan permohonan judul dan pembimbing tugas akhir. Adapun judul yang saya usulkan sebagai berikut:

1. Implementasi Keycloak Sebagai Single Sign-On Authen pada N Aplikasi Menggunakan Metode OAuth2 Connect
2. Implementasi Galera Cluster Menggunakan RDMS Database Sebagai Simulasi Data dan Tinja Database dengan Metode Master-Slave Replication
3. Implementasi Galera Cluster Sebagai Asynchronous Deployment Pada Docker Swarm Cluster

Atas perhatiannya, saya ucapkan terima kasih.

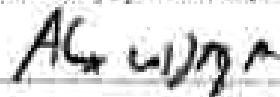
Kelompok Studi,


Hormat saya,


(Akhla Kartasura)


Ketua Program Studi,
(Akhla Kartasura)

Pembimbing Karya Akhir



Spesifikasi Program Studi:

1. Format dan layout yang digunakan mengikuti format yang tertera di bawah ini.
2. Format Judul II yang tertera di atas dan Pembimbing Akhir yang tertera di bawah ini.
3. Format dan layout yang tertera di bawah ini.
4. Format dan layout yang tertera di bawah ini.
5. Format dan layout yang tertera di bawah ini.
6. Format dan layout yang tertera di bawah ini.
7. Format dan layout yang tertera di bawah ini.
8. Format dan layout yang tertera di bawah ini.
9. Format dan layout yang tertera di bawah ini.
10. Format dan layout yang tertera di bawah ini.
11. Format dan layout yang tertera di bawah ini.
12. Format dan layout yang tertera di bawah ini.
13. Format dan layout yang tertera di bawah ini.
14. Format dan layout yang tertera di bawah ini.
15. Format dan layout yang tertera di bawah ini.
16. Format dan layout yang tertera di bawah ini.
17. Format dan layout yang tertera di bawah ini.
18. Format dan layout yang tertera di bawah ini.
19. Format dan layout yang tertera di bawah ini.
20. Format dan layout yang tertera di bawah ini.



FORMULIR
Permohonan
Penggantian Judul TA/Skripsi

Nama Dik	IRMA/16.04111
Nama Resmi	IRI
Tgl. Berlaku	01 Februari 2017
Kelas	7.3

Palembang, 7 Agustus 2024

Perihal : Permohonan Penggantian Judul & Pembimbing
Tugas Akhir/ Skripsi

Kepada Yth
Ketua Program Studi Teknik Informatika
Fakultas Sains Teknologi Universitas Bina Darma
di-

Palembang

Dengan hormat,

Saya yang bertanda tangan di bawah ini, mahasiswa Fakultas Sains Teknologi Universitas Bina Darma Palembang :

Nama : Akhlak Kamtawara
Nim : 221420106
Semester : 4
Program Studi : Teknik Informatika

Dengan ini bermaksud mengajukan permohonan penggantian judul dan pembimbing tugas akhir/ skripsi.
Adapun judul yang saya ajukan sebagai berikut :

Judul Lama :
IMPLEMENTASI GALERA CLUSTER MENGGUNAKAN RDBMS DATABASE SEBAGAI KONEKSI
BACA DAN TULIS DATABASE DENGAN METODE MASTER SLAVE REPLICATION

Judul Baru
IMPLEMENTASI MASTER-SLAVE SEBAGAI KONEKSI BACA DAN TULIS DATABASE DENGAN
METODE REPLICATION

Alasan Perubahan Judul

Judul penelitian terlalu panjang

Atau perhatiannya, saya ucapkan terima kasih.

Mengetahui,
Ketua Program Studi Bina Skripsi

(Alik Wijaya, S.Kom., M.I.T)

Pembimbing I : Alik Wijaya, S.Kom., M.I.T

Pembimbing II :

Hormat Saya,

(Akhlak Kamtawara)