

PENERAPAN REPLIKASI DATA PADA BASIS DATA MYSQL UNTUK MENJAGA KETERSEDIAAN DATA

IMPLEMENTING DATA REPLICATION IN MYSQL DATABASES TO MAINTAIN DATA AVAILABILITY

**Evan Fernando^{1*}, Ghifari Hendrawansyah², Muhammad Kafka Nafisah³,
Siti Raudha Al-Zahra⁴, Suci Adri Avrillia⁵, Uus Firdaus⁶**
¹⁻⁶Universitas Djuanda

***Email Correspondence: evanfernando294@gmail.com**

Abstract

Data availability is a crucial parameter in information system infrastructure, where single server failure is a major risk. To mitigate this risk, database replication is recognized as an effective solution because it can increase data availability and enable the distribution of the same data across multiple locations. This study aims to implement and analyze the effectiveness of MySQL data replication techniques to ensure data continuity and availability. The method used is an experimental study by building a Master-Slave replication architecture using an asynchronous replication mechanism. Testing focused on two main metrics: replication lag time as an indicator of synchronization, and handover time as an indicator of system recovery speed. The implementation results show stable system operation with a minimal average replication lag of 0.00-0.01 seconds in a 1 Gbps network environment. In a failure simulation, the failover process to the replica server was successfully completed within 12-15 seconds. This implementation proves that MySQL data replication is effective in increasing fault tolerance and ensuring continuous data availability.

Keywords: Implementation, Data Replication, Mysql Database.

Abstrak

Ketersediaan data merupakan parameter krusial dalam infrastruktur sistem informasi, di mana kegagalan server tunggal menjadi risiko utama. Untuk memitigasi risiko tersebut, replikasi basis data diakui sebagai solusi efektif karena dapat meningkatkan ketersediaan data dan memungkinkan distribusi data yang sama di beberapa lokasi. Penelitian ini bertujuan mengimplementasikan dan menganalisis efektivitas teknik replikasi data MySQL untuk menjamin kontinuitas dan ketersediaan data. Metode yang digunakan adalah studi eksperimental dengan membangun arsitektur replikasi Master-Slave menggunakan mekanisme asynchronous replication. Pengujian difokuskan pada dua metrik utama: waktu lag replikasi sebagai indikator sinkronisasi, dan waktu pengalihan sebagai indikator kecepatan pemulihan sistem. Hasil implementasi menunjukkan sistem beroperasi stabil dengan rata-rata replication lag yang minimal, yaitu 0.00-0.01 detik dalam lingkungan jaringan 1 Gbps. Dalam simulasi kegagalan, proses failover menuju server replika berhasil diselesaikan dalam waktu 12-15 detik. Penerapan ini membuktikan replikasi data MySQL efektif meningkatkan toleransi kesalahan dan menjamin ketersediaan data secara berkelanjutan.

Kata Kunci: Penerapan, Replikasi Data, Basis Data Mysql.

PENDAHULUAN

Perkembangan teknologi informasi telah menjadikan data sebagai aset strategis yang harus tersedia kapan pun dibutuhkan. Berbagai sektor seperti perbankan,

kesehatan, e-commerce, telekomunikasi, dan pemerintahan sangat bergantung pada sistem basis data yang andal. Ketidaktersediaan data dapat menyebabkan layanan terganggu, kehilangan transaksi, gangguan operasional, dan reputasi buruk bagi organisasi.

Sistem basis data yang berjalan pada satu server fisik memiliki risiko tinggi terhadap single point of failure. Ketika server tersebut mengalami gangguan, seluruh layanan yang bergantung pada basis data akan berhenti. Oleh karena itu, diperlukan mekanisme redundansi yang memungkinkan sistem tetap beroperasi meskipun salah satu komponen mengalami kegagalan. Replikasi asynchronous pada basis data terdistribusi merupakan solusi yang efektif untuk meningkatkan keamanan dan ketersediaan data.

MySQL sebagai salah satu DBMS populer menyediakan fitur replikasi yang dapat digunakan untuk menjaga ketersediaan data. Replikasi master-slave memungkinkan server sekunder menerima salinan data dari server utama secara otomatis, sehingga dapat mengambil alih proses baca ketika master mengalami gangguan. Implementasi replikasi database MySQL telah terbukti efektif untuk menjaga integritas data pada sistem informasi.

Berbagai penelitian telah menunjukkan efektivitas replikasi MySQL dalam konteks ketersediaan data. Analisis kinerja replikasi database menggunakan metode Master-Slave menunjukkan bahwa arsitektur ini mampu meningkatkan performa sistem serta mengurangi risiko kehilangan data. Perancangan replikasi basis data dengan mekanisme pengamanan menggunakan enkripsi SSL juga dapat meningkatkan keamanan transfer data antar server replikasi.

Penelitian serupa juga dilakukan pada sistem basis data lain. Penerapan replikasi synchronous pada aplikasi online menggunakan PostgreSQL menemukan bahwa meskipun replikasi synchronous memberikan konsistensi data yang lebih tinggi, namun memiliki overhead performa yang lebih besar dibandingkan asynchronous. Penerapan sistem replikasi dan integrasi basis data terdistribusi pada pangkalan data pendidikan tinggi menunjukkan bahwa replikasi efektif dalam menjamin konsistensi dan ketersediaan data di berbagai lokasi geografis.

Meskipun berbagai penelitian telah dilakukan, masih diperlukan kajian lebih mendalam mengenai implementasi praktis replikasi MySQL dengan fokus pada pengukuran replication lag dan waktu failover sebagai indikator ketersediaan data. Penelitian ini bertujuan untuk mengisi celah tersebut dengan mengimplementasikan dan menganalisis sistem replikasi MySQL master-slave secara komprehensif.

Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah penelitian ini adalah :

- 1) Bagaimana merancang dan mengimplementasikan arsitektur replikasi MySQL master-slave yang efektif untuk menjaga ketersediaan data?
- 2) Bagaimana performa sistem replikasi MySQL dalam hal replication lag dan waktu failover?
- 3) Seberapa efektif replikasi MySQL dalam meningkatkan toleransi kesalahan dan kontinuitas layanan?

Tujuan Penelitian

Penelitian ini bertujuan untuk :

- 1) Merancang dan mengimplementasikan arsitektur replikasi MySQL master-slave sesuai standar operasional.
- 2) Mengukur dan menganalisis performa sistem replikasi melalui metrik replication lag dan failover time.
- 3) Mengevaluasi efektivitas replikasi MySQL dalam menjaga ketersediaan dan kontinuitas data.

METODE

Pendekatan Penelitian

Penelitian ini menggunakan pendekatan eksperimen terkontrol dengan mengimplementasikan replikasi MySQL dalam lingkungan uji untuk menganalisis kinerja dan ketahanannya terhadap kegagalan system

Tahapan Penelitian

Penelitian dilakukan melalui lima tahapan sistematis :

- 1) Pengumpulan Data dan Studi Literatur
Tahap ini mencakup pengumpulan referensi dari jurnal ilmiah, dokumentasi resmi MySQL, serta penelitian terdahulu terkait replikasi database dan konsep ketersediaan data. Studi literatur dilakukan untuk memperoleh landasan teori dan memahami metode terbaik dalam implementasi replikasi.
- 2) Perancangan Arsitektur Replikasi
Pada tahap ini dirancang topologi replikasi master-slave dengan satu server master dan satu server slave. Perancangan meliputi penentuan spesifikasi server, konfigurasi jaringan, rencana pengujian, serta parameter evaluasi seperti replication lag dan waktu failover.
- 3) Implementasi Konfigurasi MySQL
Tahap implementasi mencakup instalasi MySQL Server 8.x, pengaktifan binary log, konfigurasi master dan replica, pembuatan user replikasi dengan privilege REPLICATION SLAVE, serta sinkronisasi awal data. Keamanan koneksi replikasi juga diperhatikan dengan menerapkan prinsip pengamanan yang tepat.
- 4) Pengujian dan Simulasi Gangguan
Sistem diuji melalui berbagai skenario, termasuk pengujian sinkronisasi data, pengujian pada beban tinggi, simulasi failover dengan mematikan master secara mendadak, dan pengukuran replication lag.
- 5) Analisis Hasil dan Validasi
Data yang diperoleh dari pengujian dianalisis untuk mengevaluasi efektivitas replikasi MySQL. Hasil penelitian dibandingkan dengan referensi dan teori yang relevan untuk memastikan validitas temuan serta menarik kesimpulan terkait kontribusi replikasi terhadap peningkatan ketersediaan data.

Perangkat dan Lingkungan Uji

Penelitian dilakukan pada lingkungan uji dengan spesifikasi sebagai berikut :

Sistem Operasi : Ubuntu Server 22.04 LTS dipilih karena stabilitasnya dan kompatibilitas yang baik dengan MySQL.

Database Server : MySQL Server versi 8.x yang mendukung fitur replikasi modern seperti asynchronous replication, semi-synchronous replication, serta peningkatan pada binary log.

Infrastruktur Server : Menggunakan dua buah virtual machine dengan spesifikasi CPU multi-core, RAM minimal 2-4 GB, dan penyimpanan berbasis SSD. Server pertama berfungsi sebagai Master yang menerima seluruh operasi tulis, dan server kedua sebagai Slave yang menerima salinan data.

Jaringan : Pengujian dilakukan menggunakan jaringan internal dengan kecepatan 1 Gbps untuk memberikan bandwidth tinggi dan latensi rendah, mendekati kondisi lingkungan produksi.

Tools Monitoring : MySQL Admin dan MySQL Enterprise Monitor digunakan untuk memantau status server, replication lag, aktivitas query, dan mendeteksi anomali secara real-time.

Skenario Pengujian

Empat skenario pengujian dirancang untuk mengevaluasi efektivitas system :

- 1) Pengujian Sinkronisasi Data
Memastikan setiap perubahan data pada master dapat direplikasi secara konsisten ke slave dengan menjalankan operasi INSERT, UPDATE, dan DELETE, kemudian memverifikasi kesesuaian data.
- 2) Pengujian pada Beban Baca-Tulis Tinggi
Menguji stabilitas replikasi saat sistem diberi beban tinggi dengan query baca dan tulis secara simultan untuk menilai pengaruhnya terhadap throughput, latency, serta beban CPU dan I/O.
- 3) Simulasi Failover
Mematikan server master secara mendadak untuk mensimulasikan kondisi kegagalan nyata, kemudian mengamati kemampuan slave untuk mengambil alih dan mengukur waktu pemulihan.
- 4) Pengukuran Replication Lag
Mengukur selisih waktu antara perubahan di master dan penerapannya di slave menggunakan parameter Seconds Behind Master untuk menilai kecepatan sinkronisasi.

HASIL DAN PEMBAHASAN

A. Hasil

1) Arsitektur Replikasi MySQL

Arsitektur replikasi yang diimplementasikan menerapkan model master-slave dengan mekanisme asynchronous replication. Server master menangani seluruh operasi penulisan seperti INSERT, UPDATE, dan DELETE, serta mencatat setiap perubahan ke dalam binary log. Server slave melakukan replikasi menggunakan dua komponen utama: I/O Thread yang membaca binary log dari master dan menyimpannya ke relay log, serta SQL Thread yang mengeksekusi relay log sehingga data menjadi sinkron dengan master. Implementasi arsitektur ini mengikuti prinsip bahwa replikasi asynchronous memberikan performa tinggi karena master tidak menunggu konfirmasi dari replica setelah menulis ke binary log. Dalam implementasinya, permintaan dari client diarahkan sesuai kebutuhan: operasi write ke master dan operasi read ke slave untuk mengurangi beban master.

2) Implementasi Replikasi

Implementasi dilakukan secara bertahap dengan langkah-langkah berikut: Pertama, mengaktifkan binary log pada master dengan menambahkan konfigurasi log_bin dan server-id pada file mysqld.cnf. Kedua, membuat user khusus untuk replikasi dengan privilege REPLICATION SLAVE. Ketiga, menentukan server-id yang unik untuk setiap server agar MySQL dapat membedakan masing-masing node. Keempat, melakukan dump database dari master menggunakan mysqldump dengan parameter all-databases dan master-data untuk mendapatkan snapshot awal. Kelima, mengimpor dump ke server slave. Keenam, menjalankan konfigurasi replikasi pada slave dan memverifikasi status dengan perintah SHOW SLAVE STATUS. Proses implementasi ini mengadopsi praktik terbaik dalam implementasi replikasi database untuk menjaga integritas data.

3) Hasil Pengukuran Replication Lag

Pengukuran replication lag dilakukan untuk mengetahui kecepatan sinkronisasi antara master dan slave. Parameter yang diukur adalah nilai Seconds_Behind_Master yang menunjukkan selisih waktu antara eksekusi event pada master dan penerapannya pada slave. Berdasarkan hasil pengamatan selama pengujian pada

lingkungan jaringan internal 1 Gbps, diperoleh hasil sebagai berikut : Replication Lag Minimum mencapai 0.00 detik yang menunjukkan tidak ada keterlambatan, sedangkan Replication Lag Maksimum sebesar 0.01 detik yang menunjukkan keterlambatan minimal. Rata-rata Replication Lag berada di angka 0.00 hingga 0.01 detik. Hasil ini menunjukkan bahwa proses replikasi berjalan hampir real-time dan mampu menjaga keselarasan data antara master dan slave dengan sangat baik. Nilai lag yang sangat rendah disebabkan oleh tiga faktor utama: bandwidth jaringan yang tinggi memungkinkan transmisi binary log berlangsung cepat, beban transaksi yang terkontrol membuat slave mampu mengeksekusi event binary log tanpa penundaan, dan spesifikasi perangkat keras yang seimbang sehingga tidak terjadi bottleneck pada proses SQL Thread.

4) Hasil Simulasi Failover

Simulasi failover dilakukan dengan mematikan server master secara tiba-tiba untuk mensimulasikan kondisi kegagalan nyata seperti crash atau pemadaman listrik. Hasil pengukuran menunjukkan: Waktu Deteksi Kegagalan memerlukan 3 hingga 5 detik, Waktu Promosi Slave ke Master memerlukan 8 hingga 10 detik, sehingga Total Waktu Failover berkisar antara 12 hingga 15 detik. Proses failover berhasil diselesaikan dengan downtime yang sangat minimal. Setelah slave dipromosikan menjadi master baru, sistem dapat kembali melayani request dari aplikasi. Konsistensi data tetap terjaga karena slave telah menerima sebagian besar perubahan dari master sebelum kegagalan terjadi.

5) Pengujian Konsistensi Data

Verifikasi konsistensi data dilakukan dengan membandingkan checksum data antara master dan slave sebelum simulasi kegagalan. Hasil menunjukkan bahwa 100 persen data pada slave konsisten dengan data di master, mengkonfirmasi bahwa tidak ada kehilangan data selama proses replikasi dalam kondisi normal.

B. Pembahasan

1) Efektivitas Replikasi dalam Menjaga Ketersediaan Data

Hasil penelitian menunjukkan bahwa implementasi replikasi MySQL dengan arsitektur master-slave efektif dalam meningkatkan ketersediaan data. Dengan replication lag rata-rata hanya 0.00 hingga 0.01 detik, sistem dapat memastikan bahwa data yang tersedia di server slave hampir real-time dengan data di master. Hal ini membuktikan bahwa replikasi asynchronous, meskipun memiliki potensi kehilangan data minimal, tetap dapat diandalkan untuk aplikasi yang memerlukan ketersediaan tinggi. Kecepatan sinkronisasi yang tinggi ini memungkinkan server slave untuk mengambil alih fungsi pembacaan data tanpa risiko menyajikan data yang sudah usang. Kemampuan sistem untuk menjaga integritas data sambil meningkatkan ketersediaan merupakan indikator utama keberhasilan implementasi replikasi.

2) Performa Failover dan Pemulihan Sistem

Waktu failover sebesar 12 hingga 15 detik menunjukkan bahwa sistem memiliki kemampuan pemulihan yang cepat saat terjadi kegagalan pada server master. Dibandingkan dengan sistem tanpa replikasi yang dapat mengalami downtime berjam-jam hingga berhari-hari untuk pemulihan, waktu 12 hingga 15 detik merupakan peningkatan yang sangat signifikan. Hasil ini konsisten dengan temuan bahwa arsitektur master-slave memungkinkan implementasi mekanisme failover yang cepat karena replica selalu mengikuti perubahan pada master. Meskipun ada periode singkat downtime, waktu ini dapat diterima untuk sebagian besar aplikasi

bisnis yang memerlukan high availability.

3) Perbandingan dengan Jenis Replikasi Lain

Penelitian ini menggunakan replikasi asynchronous yang memberikan performa tinggi dengan replication lag minimal. Berbeda dengan replikasi synchronous yang diterapkan pada sistem basis data lain, penelitian ini menunjukkan bahwa replikasi asynchronous pada MySQL memberikan keseimbangan yang baik antara performa dan ketersediaan data. Replikasi synchronous memberikan konsistensi data yang lebih tinggi karena master menunggu konfirmasi dari replica, namun dengan trade-off berupa latensi yang lebih besar dan penurunan throughput. Sebaliknya, replikasi asynchronous yang diimplementasikan dalam penelitian ini memberikan throughput yang lebih tinggi dengan risiko kehilangan data yang sangat kecil, terutama dalam kondisi jaringan yang stabil.

4) Distribusi Beban dan Skalabilitas

Implementasi replikasi master-slave memungkinkan distribusi beban query baca ke server slave, sehingga mengurangi beban pada master dan meningkatkan performa sistem secara keseluruhan. Dengan mengarahkan semua operasi tulis ke master dan operasi baca ke slave, sistem dapat menangani lebih banyak concurrent users tanpa degradasi performa yang signifikan. Penelitian sebelumnya membuktikan bahwa distribusi beban ini dapat meningkatkan throughput sistem hingga 40 hingga 60 persen tergantung pada rasio operasi baca terhadap tulis. Dalam konteks aplikasi dengan beban baca yang tinggi seperti e-commerce atau sistem informasi akademik, manfaat ini sangat signifikan.

5) Implementasi pada Basis Data Terdistribusi

Penerapan sistem replikasi pada pangkalan data yang terdistribusi di berbagai lokasi geografis menunjukkan bahwa replikasi tidak hanya meningkatkan ketersediaan data, tetapi juga memungkinkan akses data yang lebih cepat dari lokasi yang berbeda karena data tersedia secara lokal. Meskipun penelitian ini dilakukan dalam lingkungan lokal, prinsip yang sama dapat diterapkan untuk implementasi basis data terdistribusi. Dengan menambahkan server slave di lokasi geografis yang berbeda, sistem dapat menyediakan akses data yang cepat dan andal untuk pengguna di berbagai wilayah, sambil tetap menjaga konsistensi data melalui mekanisme replikasi.

6) Aspek Keamanan dalam Replikasi

Aspek keamanan sangat penting dalam implementasi replikasi database, khususnya penggunaan enkripsi SSL untuk melindungi data yang ditransfer antara master dan slave. Meskipun penelitian ini dilakukan dalam jaringan internal yang terpercaya, untuk implementasi produksi, terutama yang melibatkan transfer data melalui jaringan publik, enkripsi SSL sangat direkomendasikan. Pengamanan tambahan yang perlu dipertimbangkan meliputi autentikasi yang kuat untuk user replikasi, pembatasan akses jaringan menggunakan firewall, dan monitoring berkelanjutan untuk mendeteksi anomali atau upaya akses tidak sah. Kombinasi replikasi untuk ketersediaan data dengan mekanisme keamanan yang kuat akan menghasilkan sistem yang tidak hanya andal tetapi juga aman.

7) Keterbatasan dan Tantangan

Meskipun replikasi MySQL terbukti efektif, terdapat beberapa keterbatasan yang perlu diperhatikan. Pertama, replikasi asynchronous memiliki risiko kehilangan data transaksi terakhir jika master gagal sebelum replikasi selesai. Untuk aplikasi yang sangat sensitif terhadap kehilangan data, perlu dipertimbangkan penggunaan semi-synchronous replication atau Group Replication. Kedua, kompleksitas administrasi

meningkat dengan bertambahnya jumlah node replikasi. Monitoring dan pemeliharaan sistem replikasi memerlukan keahlian khusus dan tools yang memadai. Ketiga, replikasi memerlukan sumber daya server tambahan, yang berarti peningkatan biaya infrastruktur. Pemilihan jenis replikasi harus disesuaikan dengan kebutuhan spesifik aplikasi, mempertimbangkan trade-off antara performa, konsistensi, dan biaya. Untuk aplikasi yang memerlukan konsistensi absolut, replikasi synchronous mungkin lebih tepat meskipun dengan performa yang lebih rendah.

8) Implikasi Praktis

Dari sisi praktis, implementasi replikasi MySQL dengan arsitektur master-slave memberikan beberapa keuntungan signifikan : Pertama, mengurangi risiko kehilangan data akibat kegagalan hardware atau software pada server master. Kedua, memungkinkan distribusi beban pembacaan ke beberapa server slave, meningkatkan skalabilitas sistem. Ketiga, menyediakan mekanisme disaster recovery yang cepat dengan waktu pemulihan yang minimal. Keempat, biaya implementasi yang relatif rendah dibandingkan solusi high availability komersial lainnya. Replikasi database MySQL merupakan investasi yang cost-effective untuk organisasi yang memerlukan sistem basis data dengan tingkat ketersediaan tinggi namun memiliki keterbatasan budget untuk solusi enterprise yang lebih mahal.

9) Rekomendasi Pengembangan

Berdasarkan hasil penelitian dan pembahasan, beberapa rekomendasi untuk pengembangan lebih lanjut meliputi : Pertama, implementasi mekanisme automatic failover menggunakan tools seperti MySQL Router, ProxySQL, atau Orchestrator untuk mengurangi waktu failover dan meminimalkan intervensi manual. Kedua, penambahan monitoring komprehensif dengan sistem alerting untuk deteksi dini masalah replikasi. Ketiga, eksplorasi penggunaan semi-synchronous replication untuk aplikasi yang memerlukan jaminan konsistensi data yang lebih tinggi. Keempat, implementasi backup berkala sebagai lapisan keamanan tambahan untuk mengantisipasi kegagalan sistemik atau disaster yang mempengaruhi semua node. Kelima, pengujian pada skala yang lebih besar dengan berbagai skenario beban kerja yang lebih kompleks untuk memvalidasi performa sistem dalam kondisi produksi yang sebenarnya.

KESIMPULAN

Berdasarkan hasil implementasi dan analisis yang telah dilakukan, dapat ditarik beberapa kesimpulan penting : Pertama, replikasi MySQL dengan arsitektur master-slave menggunakan mekanisme asynchronous replication terbukti efektif dalam menjaga ketersediaan data. Sistem mampu menyediakan redundansi data yang memadai untuk mengatasi risiko single point of failure yang merupakan ancaman utama dalam infrastruktur basis data tunggal. Kedua, performa replikasi sangat baik, ditunjukkan dengan replication lag rata-rata 0.00 hingga 0.01 detik dalam lingkungan jaringan 1 Gbps. Nilai ini mengindikasikan bahwa proses sinkronisasi data berjalan hampir real-time dan mampu menjaga konsistensi data antara master dan slave dengan optimal. Ketiga, kemampuan failover sistem sangat baik, dengan total waktu pemulihan berkisar 12 hingga 15 detik saat terjadi kegagalan pada server master. Waktu ini menunjukkan bahwa sistem dapat meminimalkan downtime dan mempertahankan kontinuitas layanan, sesuai dengan prinsip high availability. Keempat, distribusi beban kerja dapat dilakukan dengan efektif dengan

mengarahkan operasi tulis ke master dan operasi baca ke slave, sehingga mengurangi beban pada master dan meningkatkan throughput sistem secara keseluruhan. Manfaat ini sangat signifikan untuk aplikasi dengan rasio pembacaan data yang tinggi. Kelima, meskipun efektif, implementasi replikasi memerlukan perencanaan yang matang, termasuk konfigurasi yang tepat, monitoring berkelanjutan, dan pemahaman tentang keterbatasan metode asynchronous yang memiliki risiko kehilangan data minimal. Untuk aplikasi yang sangat kritis, perlu dipertimbangkan penggunaan semi-synchronous replication atau kombinasi dengan mekanisme backup berkala.

DAFTAR PUSTAKA

- Belluano, P. L. L. (2017). Penerapan sistem replikasi dan integrasi basis data terdistribusi pada pangkalan data pendidikan tinggi (PDPT). *ILKOM Jurnal Ilmiah*, 9(1), 42–48.
- Muliawan, D., & Iskandar, D. (2023). Penerapan replikasi synchronous pada aplikasi data KRS online pada Universitas Jabal Ghafur menggunakan PostgreSQL. *Future Academia: The Journal of Multidisciplinary Research on Scientific and Advanced*, 1(1), 31–45.
- Rahayu, S., & Yunitasari, L. (2018). Analisis kinerja replikasi database MySQL menggunakan metode Master-Slave. *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi*.
- Riadi, I., & Erpani, R. (2020). Implementasi replikasi database MySQL untuk menjaga integritas data pada sistem informasi. *INTECOM: Jurnal Information Technology and Computer Science*, 3(2), 178–186.
- Widyasto, D. (2016). Penerapan replikasi asynchronous pada basis data terdistribusi untuk keamanan data. *Jurnal Ilmiah Teknologi dan Informasi*, 6(1), 62–74.
- Yuliansyah, H. (2014). Perancangan replikasi basis data MySQL dengan mekanisme pengamanan menggunakan SSL Encryption. *Jurnal Informatika*, 8(1), 826–836.