LOCUS JOURNAL: Research & Devotion
Volume 4 No. 9 September 2025
E-ISSN 2829-7334| P-ISSN 2829-5439
Homepage: https://locus.rivierapublishing.id/index.php/jl

# Evaluation and Optimization of MySQL Master-Slave Performance with Quantitative Methods on the Database of Psychology Test Applicants SIM PT XYZ at Polda Metro Jaya

**Iin Sahuri Haryoto*, Gerry Firmansyah, Budi Tjahjono, Agung Mulyo Widodo, Habibullah Akbar, Nenden Siti Fatonah**
Universitas Esa Unggul, Indonesia
Email: iin@student.esaunggul.ac.id* , gerry@esaunggul.ac.id, budi.tjahjono@esaunggul.ac.id, agung.mulyo@esaunggul.ac.id, habibullah.akbar@esaunggul.ac.id, nenden.siti@esaunggul.ac.id

Abstract:
The development of information technology and cloud computing has enabled organizations to manage data on a scale efficiently. This study analyzes the performance problems of the master-slave MySQL database at PT XYZ, which is engaged in psychological tests for Driver's License (SIM). The research problem identified was the declining performance of the database system, with a data volume of 4,000–5,000 entries per day, which led to slow response times for read and write queries and interfered with the operational efficiency of psychology officers. The purpose of this study is to optimize the performance of MySQL databases by adjusting server configurations and specifications to improve the efficiency of the master-slave replication system. The research method uses an experimental quantitative approach with benchmarking tools to analyze performance before and after optimization. The test results showed a significant improvement in system performance after server configuration optimization, with CPU usage dropping from 30.9% to 1.7% and memory usage from 33.9% to 11.0%. The response time of CRUD operations has also improved, with the CREATE operation reaching 0.0245 seconds, GET 0.0010 seconds, UPDATE 0.0021 seconds, and DELETE 0.0017 seconds. The implications of this study contribute to the development of large-scale database optimization strategies and can serve as a practical reference in the implementation of MySQL replication systems for organizations with high data volumes.
**Keywords:** MySQL, master-slave replication, database optimization, system performance, cloud computing.

*Corresponding*: Iin Sahuri Haryoto
E-mail: iin@student.esaunggul.ac.id

## INTRODUCTION

The development of information technology with internet media is essential to increase competitiveness in various fields such as education, politics, economics, and business (Fatimah et al., 2023; Sutrisno et al., 2023). The internet, which was initially only a medium for sending data and information, has now evolved into a strategic tool that supports innovation, transparency, decision-making, and organizational growth (Bacca-Acosta et al., 2023; Bai et al., 2021; Tirno, 2024; Zhang, 2024; Zhang et al., 2023; Asif et al., 2024; Yu et al., 2025). Therefore, every organization requires an integrated system that aligns with digital transformation to sustain development and competitiveness.

Cloud computing delivers hardware and software as virtual services, freeing users from low-level system administration. However, migrating web applications and integrating them into existing infrastructure is complex and creates new challenges across technical, cultural, legal, regulatory, and social levels (Althani, 2025; Fahmideh et al., 2020; Najana & Ranjan, 2024; Ali et al., 2025; Anand, 2024). Organizations must innovate paradigms and practices to handle these multidimensional challenges effectively.

Cloud computing providers, including Amazon Web Services (AWS), Salesforce.com, and Google App Engine, provide users with the option to deploy their applications over an unlimited network of resource pools with virtually no capital investment and with simple operating costs comparable to actual usage. By leveraging cloud services to host web applications, organizations can gain advantages such as elasticity, pay-per-use, and abundant resources. However, organizations tend to avoid or delay migrating web applications to the cloud due to various hurdles (Menzel & Ranjan, 2012).

The master-slave architecture is a traditional solution that has been used for a long time in the database world, where one master handles writing data and multiple slaves are used to read the data. In many applications, a large number of end users require data reading services from the database, while a small number (mostly system administrators) perform data writes for database updates. The system uses a read-write separation mechanism in which the main database server (master) is used for data writing, while multiple slave database servers are balanced for data reading. Thus, the architecture supports read scalability, as the number of slaves can be easily scaled in the cloud as needed at any given time (Shrestha, 2017).

PT XYZ is a company engaged in psychological tests for Driver's License (SIM) applicants. The driver's license psychological test is one of the stages that must be passed by applicants who want to get a new driver's license or extend their license, as stipulated in Article 81 Paragraph 4 of Law 22 of 2009 concerning Road Traffic and Transportation. In its implementation, applicants are required to register through a registration officer. Furthermore, applicants conduct psychological tests online through the PSI test web provided. If the applicant does not pass the test, they must repeat it. Conversely, if passed, the test result data will be verified by psychological officers stationed at several locations such as SATPAS, GERAI, or SIMLING. Each officer uses one application to verify the applicant's test result data. All applicant data is stored in the database.

In this company, the server used is cloud-based with a pay-as-you-go scheme. The system has implemented master-slave replication to support database performance. However, with the increase in data volume and number of users, the system's performance is still not optimal, especially regarding the time of read and write queries. The data is managed on a fairly large scale, with an average of 4,000 to 5,000 entries per day. However, several obstacles have arisen in the implementation of this system, such as slow database performance. When data is added and data retrieval is increasing, it is necessary to optimize the database to be more efficient.

The research problem identified was the decline in the performance of the MySQL master-slave database, which caused slow system response times, especially in read and write query operations. The urgency of this research lies in its direct impact on the operational efficiency of psychological officers in verifying the test results of driver's license applicants, which can interfere with public services. Previous research by Shrestha (2017) showed that traditional master-slave architectures have limitations in handling high workloads. However, research gaps remain related to server configuration optimization and proper hardware specifications for data volumes of 4,000–5,000 entries per day. The novelty of this research lies in its experimental quantitative approach that integrates the optimization of software and hardware configurations simultaneously in a cloud computing environment with auto-scaling. The aim of the research is to optimize the performance of MySQL databases through adjustment of server configurations and hardware specifications to improve the efficiency of the master-slave replication system.

The benefits of this research theoretically contribute to developing large-scale database optimization strategies, practically help PT XYZ address system performance issues, and provide a reference for other organizations with similar challenges in managing high-volume databases.

Based on these problems, a solution is needed to optimize the performance of the MySQL database. This optimization is expected to solve performance-related issues so that the system can run more effectively and efficiently.

Based on the description of the background of the problem above: a) How can the problem of the master-slave database always being slow when adding or retrieving data be solved? b) How does the slow collection and addition of data interfere with the efficiency of psychological officers in verifying test results?

The objectives to be achieved in this writing are as follows: optimize the performance of MySQL databases by improving the system's read and write query response times to be more efficient; determine solutions to minimize latency by evaluating options to reduce replication lag and improve data processing speed; ensure system reliability by increasing the stability and reliability of the database system to support evolving operations; improve MySQL database performance to make data retrieval faster as data volume continues to grow.

The benefits obtained from this study include theoretical contributions to the science of database optimization, especially in MySQL management; practical benefits for users (SIM applicants) through a faster and smoother process; benefits for PT XYZ by addressing performance issues, making the system more efficient; and benefits for the author by providing convenience in implementing improvements within the company.

The problem limitation in this study is that the proposed solution will focus on optimizing performance and efficiency from the database side, excluding changes or updates to other applications or systems integrated with the database. This research will focus on managing increasing data growth in the PT XYZ database at medium and large data scales but will not cover even larger data scales.

## RESEARCH METHOD

This study used a quantitative approach with an experimental method to analyze and optimize the performance of MySQL databases at PT XYZ, a company engaged in SIM psychology tests at POLDA METRO JAYA with 14 office employees and 102 field officers. The company managed an average of 4,000 to 5,000 SIM psychology test applicants per day, or around 120,000 data entries per month, requiring optimization of the MySQL database system that implemented master-slave replication. The research stages included problem identification, initial data collection using benchmarking tools, literature study of optimization solutions, design and implementation of optimization solutions including the implementation of MySQL Group Replication, post-optimization testing, data analysis with descriptive and inferential statistics, and drawing conclusions to provide recommendations for improving the performance of the company's database system.

## RESULTS AND DISCUSSION

## Master and Slave Preparation on JagoanCloud

### *MySQL Installation on Master Server and Slave Server*

In this part, the installation process of master and slave servers is carried out as part of the implementation of the database replication system. After installation, the specifications of the two servers are equalized to have an equivalent configuration, thus ensuring optimal performance in the replication process. The harmonized specification includes key components such as RAM and CPU, to avoid imbalances in data processing. Details of the specifications and configurations can be seen in the following figure.

### *Master Server Database Installation*

In this section, the database installation process on the master server has been successfully completed. The installation is carried out according to the planned configuration, which uses 4 GB of RAM, 12.8 GHz CPU, and 50 GB of storage. The system also comes with an auto scaling feature, which allows the server to automatically increase its capacity when resource usage reaches a specified threshold. If the server load is close to maximum capacity, the system will increase its specifications to 12.5 GB of RAM and 40 GHz CPU to maintain performance and stability of the service.

### *Installation of Database Server Slave*

In this section, the database installation process on the slave server has been successfully completed. The installation is carried out according to the planned configuration, which uses 4 GB of RAM, 12.8 GHz CPU, and 50 GB of storage. Slave servers also come with an auto scaling feature, which allows for automatic capacity increases when resource usage reaches a predetermined threshold. If the server load is close to maximum capacity, the system will increase its specs to 12.5 GB of RAM and 40 GHz CPU to ensure optimal performance.

### *Master Server Configuration*

In this section, a configuration is performed on the master server to support the database replication system. This configuration process includes setting key parameters, such as log bin, server ID, and user replication, to ensure that data synchronization runs optimally.

### *Creating User Replication on the master server*

In this section, the process of creating a user account on the master server is carried out for the purpose of replication and database access. This configuration includes creating users with specific access rights that allow communication between the master and slave servers in the replication system.

### *Replica Configuration on the master server*

In this section, replication configuration is performed on the master server to allow communication and synchronization of data with the slave server. This configuration process includes the use of the CHANGE MASTER TO command, which serves to assign connection parameters to the master server, including:

a. MASTER_HOST (The address of the server to be replicated).
b. MASTER_USER (User account that has access rights for replication).
c. MASTER_PASSWORD (Password for replication account authentication).
d. MASTER_LOG_FILE (Binary log files used in the replication process)

e. MASTER_LOG_POS (The last position in the binary log that is the starting point for replication)

This configuration aims to ensure that the slave server can receive and apply data changes from the master server in real-time.

***Slave Server Configuration***

In this part, configuration is carried out on the master server to ensure that the database replication system runs optimally. The configuration process includes setting key parameters such as binary logging, server ID, as well as granting replication access rights to slave servers. These configuration steps aim to ensure accurate and efficient data synchronization between the master and slave servers.

***Creating User Replication on a slave server***

In this section, user accounts are created on the slave server to support the replication and database management process. This account is configured with specific access rights to be able to receive and apply data changes sent from the master server. This process aims to ensure that the slave server can operate safely and optimally in a replication system.

***Replica Configuration on the slave server***

In this section, a replica configuration is performed on the slave server to allow synchronization of data from the master server. This configuration includes the CHANGE MASTER TO command, which is used to assign connection parameters to the master server, including:

a. MASTER_HOST (master server address)
b. MASTER_USER (user account for replication)
c. MASTER_PASSWORD (replication account password)
d. MASTER_LOG_FILE (binary log files used for replication)
e. MASTER_LOG_POS (last position in the binary log)

This configuration aims to ensure that the slave server can receive and apply data changes in real-time from the master server.

***Testing the connection of both servers***

a. Master server: In the following image, a database with the name psi_payment100 has been added to the master server. The addition of this database is done as part of the replication system, so that the data stored on the master server can be passed to the slave server automatically.
b. Server slave: In the following image, after running the SHOW DATABASES command, psi_payment100 database has been successfully displayed on the slave server. This indicates that the replication process is going well, where the database added to the master server is already synchronized to the slave server.

***Check Replication Status***

a. Master Server: In the image below, the replication status has been checked using the SHOW SLAVE STATUS command. The results show that Slave_IO_Running:Yes and Slave_SQL_Running:Yes, which indicates that the replication process between the master and slave servers has gone well without any problems. This state ensures that the slave server receives and applies data changes from the master server in real-time.

b. Slave Server: In the image below, a replication status check is performed on the slave server using the SHOW SLAVE STATUS command. The results show that Slave_IO_Running: Yes and Slave_SQL_Running: Yes, which means the replication process has run successfully. This state signifies that the slave server can receive data from the master server and implement it without any problems. Details of the replication status can be seen in the following image.

***Attribute Database***

In the image below, some of the attributes used in the payment table are shown. These attributes include the fields used to store payment transaction information, such as PaymentId, noTransaction, Test number, nik, and other attributes that support the payment management system.

**Database Replication Testing with MySQL Server**

In this section, testing is carried out on the CREATE, GET, and DELETE operations in the database. This test aims to ensure that any changes made to the master server, such as new data creation (CREATE), data retrieval (GET), and data deletion (DELETE), can run properly and are automatically replicated to the slave server. This process ensures that the replication mechanism works optimally and that the data remains in sync between the master and slave servers. Details of the test results can be seen in the following figure.

***Stuart T***

a. Master Database: In the image below, the process of creating (CREATE) 20 data on the master server using a query through the phpMyAdmin web interface. The purpose of this data addition is to test whether the replication process is running properly, so that the data entered on the master server can be automatically synchronized to the slave server.
b. Slave Database: In the image below, the data that has been created on the master server database has been successfully replicated to the slave server database of 20 data. This indicates that the replication mechanism is working well, where any changes that occur to the master server are automatically synchronized to the slave server.

***Get***

a. Master Database: In the image below, 20 data retrievals (GET) are carried out on the master server using a query through the phpMyAdmin web interface. This process uses indexing based on the Payments id attribute to ensure that the data can be accessed quickly and efficiently. This test aims to ensure that the data stored on the master server can be retrieved correctly before being replicated to the slave server.
b. Slave Database: In the image below, 20 data retrievals (GET) are carried out on the slave server using a query through the phpMyAdmin web interface. This process uses indexing based on the idPayments attribute to ensure efficiency in data search. This test aims to ensure that the data that has been replicated from the master server can be properly accessed on the slave server.

***Update***

a. Master Database: In the image below, an update (UPDATE) was performed to the createdAt attribute for as many as 20 records on the master server using a query through the phpMyAdmin web interface. This process is done by indexing based on the idPayments attribute to ensure data updates are done efficiently. This test aims to ensure that changes made to the master server can be properly synchronized to the slave server through a replication mechanism.

b. Database Slave: In the image below, the data that has been updated on the createdAt attribute in the master server database is automatically synchronized to the database of the slave server. This shows that the replication mechanism works well, ensuring that any changes to the master server are also applied to the slave server without any problems.

***Delete***

a. Master Database: In the image below, 20 data on the master server database were deleted using the idPayment attribute as a reference. This process aims to test whether the replication mechanism is working properly, so that the deleted data on the master server will also be automatically deleted on the slave server.
b. Database Slave: In the image below, the data on the slave database server has been empty after deleting (DELETE) 20 data on the master server database. This indicates that the replication mechanism is working properly, ensuring that any data deleted on the master server is also automatically deleted on the slave server.

***CPU Server Master***

In the image below, the results of monitoring the master server's resource usage when executing CRUD activities on the database using the top -u mysql command are shown. The monitoring results show that CPU usage is 0.7% and Memory is 11.3% during the process. This data provides an overview of how the server performs when handling database operations and ensures that the system is running within optimal limits.

***CPU Server Slave***

The results of monitoring the resource usage on the slave server when executing CRUD activities on the database using the top -u mysql command. The monitoring results showed that CPU usage was 0.3% and Memory was 11.3% during the process. This data shows that the slave server can handle the replication process and query execution efficiently.

## Result

After testing the CRUD (Create, Read, Update, and Delete) operation, which includes creating new data, retrieving data, updating data, and deleting data, the researchers have observed and analyzed the performance of the system. The results of these tests provide an overview of the efficiency and effectiveness of each of these operations on the database. Based on the tests performed, we conclude the following results, which are presented in the following table:

**Table 1. Analysis Results**

| No | Name | Total Civil | Server Master | Server Slave | Time used |
|---|---|---|---|---|---|
| 1 | CREATE | 20data/min | Live | Live | 0.0306 seconds |
| | CREATE | 20data/min | Live | Die | 0.0291 seconds |
| | CREATE | 20data/min | Die | Live | 0.0245 seconds |

| | | | | | |
|---|---|---|---|---|---|
| 2 | GET | 20 Dates | Live | Live | 0.0010 seconds |
| | GET | 20 Dates | Live | Die | 0.0010 seconds |
| | GET | 20 Dates | Die | Live | 0.0011 seconds |
| 3 | UPDATE | 20 Dates | Live | Live | 0.0021 seconds |
| | UPDATE | 20 Dates | Live | Die | 0.0023 seconds |
| | UPDATE | 20 Dates | Die | Live | 0.0021 seconds |
| 4 | DELETE | 20 Dates | Live | Live | 0.0017 seconds |
| | DELETE | 20 Dates | Live | Die | 0.0018 seconds |
| | DELETE | 20 Dates | Die | Live | 0.0018 seconds |

The test results show that the configuration and specification settings on the server play a very important role in running a database system with a master-slave architecture. Apparently, the accuracy of determining the appropriate server configuration, both in terms of hardware capacity and software settings, greatly affects the overall performance of the system. Every element, from processor speed, memory count, to replication settings between servers, must be carefully tuned to keep the system running optimally. Therefore, it is important to ensure that the server specifications used can support the expected workloads and reduce the likelihood of latency or disruption that could affect database performance, especially on master-slave systems that handle large volumes of data such as those used in these systems.

After searching for the previous problems, the researchers found that the configuration and specifications of the server used were still less than optimal compared to the configuration needed to support better system performance. However, once the researchers made adjustments and reconfigured the master and slave servers, the system was able to run more smoothly and efficiently, showing a significant improvement in performance compared to previous conditions.

a. Results before evaluation and optimization: In the image below trying to see the highest Ram/Memory and CPU usage, it shows the results before evaluation and optimization on the master database server, there is a showing, namely with Ram or Memory used 33.9%, then it shows that with a CPU of 30.9%. This could happen jumping with the highest number.
b. The results have been evaluated and optimized: In the image below trying to see the highest use of Ram/Memory and CPU, it shows the results after evaluation and optimization on the master database server, there is a showing, namely with Ram or Memory used 11.0%, then showing that with a CPU of 1.7%.

## CONCLUSION

This study demonstrated that the performance of MySQL database systems using master-slave replication architecture is significantly affected by server configuration and hardware specifications. By adjusting both hardware and software settings, the system achieved smoother operation, reduced latency, and improved overall performance, with efficient replication settings playing a key role in minimizing lag and ensuring reliability. Future research should explore advanced server optimization strategies, including aligning CPU, RAM, and storage capabilities with system demands, and fine-tuning replication settings to further reduce latency. Additionally, investigating the implementation of MySQL Group Replication could enhance availability and reduce downtime compared to traditional architecture. Regular system monitoring, increasing the number of slave servers to distribute read query loads, and adopting horizontal scaling or cluster database approaches are also recommended to support system scalability as data volume continues to grow.

## REFERENCES

Ali, S., et al. (2025). Security and privacy in multi-cloud and hybrid environments. *Computers & Security*. https://doi.org/10.1016/j.cose.2025.103912

Althani, B. (2025). Migration challenges of legacy software to the cloud: A socio-technical perspective. *Cogent Business & Management, 12(1),* 2503421. https://doi.org/10.1080/23311975.2025.2503421

Anand, L. G. (2024). Regulatory compliance and legal challenges in responding to cloud ransomware in the financial sector. *SSRN.* https://doi.org/10.2139/ssrn.5393722

Asif, M., Yang, L., & Hashim, M. (2024). The role of digital transformation, corporate culture, and leadership in enhancing corporate sustainable performance in the manufacturing sector of China. *Sustainability, 16(7),* 2651. https://doi.org/10.3390/su16072651

Bacca-Acosta, J., Gómez-Caicedo, M. I., Gaitán-Angulo, M., Robayo-Acuña, P., Ariza-Salazar, J., Mercado Suárez, Á. L., & Alarcón Villamil, N. O. (2023). The impact of digital technologies on business competitiveness: A comparison between Latin America and Europe. *Competitiveness Review, 33(7),* 22–46. https://doi.org/10.1108/CR-10-2022-0167

Bai, H., Zhang, D., & Li, J. (2021). Role of digital technology in transforming organizational capabilities: Moderation of product knowledge hiding. *Frontiers in Psychology, 12,* 792550. https://doi.org/10.3389/fpsyg.2021.792550

Fahmideh, M., Daneshgar, F., Beydoun, G., & Rabhi, F. (2020). Challenges in migrating legacy software systems to the cloud: An empirical study. *Journal of Information Systems.* [Working paper]. https://arxiv.org/abs/2004.10724

Fatimah, Y. A., Kannan, D., Govindan, K., & Hasibuan, Z. A. (2023). Circular economy e-business model portfolio development for e-business applications: Impacts on ESG and sustainability performance. *Journal of Cleaner Production*, *415*, 137528. https://doi.org/10.1016/j.jclepro.2023.137528

Menzel, M., & Ranjan, R. (2012). CloudGenius: Decision support for web server cloud migration. In *WWW'12 – Proceedings of the 21st Annual Conference on World Wide Web* (pp. 979–988). https://doi.org/10.1145/2187836.2187967

Najana, D. S. M., & Ranjan, P. (2024). Compliance and regulatory challenges in cloud computing: A sector-wise analysis. *International Journal of Global Innovations and Solutions (IJGIS)*. https://doi.org/10.21428/e90189c8.68b5dea5

Shrestha, R. (2017). High availability & performance of database in the cloud: Traditional master-slave replication versus modern cluster-based solutions. In *CLOSER 2017 – Proceedings of the 7th International Conference on Cloud Computing and Services Science* (pp. 385–392). https://doi.org/10.5220/0006294604130420

Sutrisno, S., Kuraesin, A. D., Siminto, S., Irawansyah, I., & Almaududi Ausat, A. M. (2023). The Role of Information Technology in Driving Innovation and Entrepreneurial Business Growth. *Jurnal Minfo Polgan*, *12*(1), 586–597. https://doi.org/10.33395/jmp.v12i1.12463

Tirno, R. R. (2024). Effect of business intelligence on organizational competitiveness:

Exploring the mediation of technology anxiety. *Computers in Human Behavior Reports, 16,* 100536. https://doi.org/10.1016/j.chbr.2024.100536

Yu, F., & Li, H. (2025). Digitalization policy and enterprise digital transformation. *Journal of Asian Economics, 87,* 101632. https://doi.org/10.1016/j.asieco.2025.101632

Zhang, F., Wang, Y., & Liu, X. (2023). Digital technology usage, strategic flexibility, and business model innovation in manufacturing firms. *Technological Forecasting and Social Change, 193,* 122102. https://doi.org/10.1016/j.techfore.2023.122102

Zhang, T. (2024). Research on business model innovation of media industry in the era of "Internet Plus." *International Journal of E-Collaboration, 20(1),* 1–12. https://doi.org/10.4018/IJeC.349963