IJAIT (International Journal of Applied Information Technology)
http://journals.telkomuniversity.ac.id/ijait/

Application of Index for Optimization Query Data in Graduate Information Systems

Agus Sifaunajah [a, *], Rosa Tarbiyatul Khusna [b]

[a] *Dept. of Information System, KH. A Wahab Hasbullah University, Jombang, East Java, Indonesia*
[b] *Dept. of Informatic, KH. A Wahab Hasbullah University, Jombang, East Java, Indonesia*
agus.syifa85@gmail.com, rosatarbiyatul08@gmail.com

ARTICLE INFO

Received September 26th, 2021
Revised January 7th, 2022
Accepted January 13th, 2022
Available online June 25th, 2022

Keywords:
index, query optimization, join

ABSTRACT

Graduates from educational institutions within the foundation of the Bahrul Ulum Islamic Boarding School each year can be more than 1000 people. The potential for this critical data has not been supported by good data collection. The data collection process is still done manually. Therefore, the level of accuracy is still fragile. In this study, an index is applied to perform query optimization. The object of this research is the database of the graduate data information system at the Bahrul Ulum Islamic Boarding School Foundation. Data collection method using literature study and interviews. The experimental stage of this research is done by determining the target output by compiling query commands. Each query command is executed ten times to find the average execution time and runs on two different hardware platforms to determine the effect of hardware on the query process. The study results show that the index can be used for data query optimization but only if it is supported by good hardware specifications so that the command execution time is optimal. When only using the index application, the average difference in execution time is 0.5 seconds.

* Corresponding author at:
Faculty of Information Technology, KH. A. Wahab Hasbullah University
Garuda Street No.09, Tambakberas, Jombang, East Java
Indonesia
E-mail address: agus.syifa85@gmail.com

ORCID ID:
First Author: 0000-0002-3923-0925

https://doi.org/10.25124/ijait.v5i02.4194
Paper_reg_number IJAIT000050203 2022

1. Introduction

In 2017, the Grand Conference of the Bahrul Ulum Islamic Boarding School Foundation gave a task mandate to the elected management to collect data on graduates, students, and employees in the related environment. This mandate is necessary because the Islamic boarding school is over a century old but lacks a valid database for the data in question. The total number of graduates of the Bahrul Ulum Islamic Boarding School is hundreds of thousands of students. The vast potential of data requires data optimization techniques in the database query process.

Optimization is a measure that leads to the achievement of goals. In general, optimization means searching for the best value of a given function in a given context [1]. It can also be interpreted as a strategy to use query paths more effectively and efficiently to minimize the total time when the query process occurs. The essence of optimizing a query is to minimize the search "path" to find data stored in a physical location [2].

Due to the rapid development of technology, humans have learned to process and search for data effectively and efficiently [3]. Query optimization makes it easier to search for and manage data. Data management can be done by accessing data in the database. Processing the data is done by performing queries on the database with the DBMS [4].

The database is an essential piece of information. Like today, all human needs have been met through digital means, and almost all people rarely use paper to record important things. By using electronic devices, data can be stored in a computer. The need for electronic devices is natural for humans during a pandemic like this. With the electronic devices we have, many positive things can be done, for example, shopping from home or other transaction activities [5].

Data processing and storage systems on computers are usually referred to as "databases." A database collects data on a computer that usually describes the activities of actors in an organization [6]. Big data takes a long time to search or process records [7]. Therefore, we must choose the correct optimization technique [8]. Data collected can be processed with various techniques and answer queries with different optimization times [9].

Optimization techniques can be done in several ways. In this study, optimizing data using the index application can increase the search speed of the desired record. However, selecting fields to be indexed must be selective because not all fields can be indexed. The implementation of the index in this optimization is carried out so that the performance of the information system can be created properly. An application that requires adequate access to this research is a database in information system data collection for the graduate of the Bahrul Ulum Islamic boarding school database, especially if the data stored is extensive. With the application of the index in the optimization of query data, it is expected to speed up the process of searching for data.

2. Methodology

2.1 The Research Object

The object of this research is a database in the graduate information system of the Bahrul Ulum Islamic Boarding School. With this database, it is expected that the data accuracy can be enhanced.

2.2 Method of Collecting Data

This research uses experimental research methods. The data collection techniques used are literature study and interviews. A literature study is used to collect data related to query optimization. The interviews were used to dig up technical-related data running in the related information system.

2.3 Experiment Stage

The experimental stages in this study are generally divided into five stages. The first stage is to analyze the database structure. In the analysis of this database structure, the selection of tables that will be related to query optimization is carried out. The second stage is to determine the expected output target. At this stage, a sample of the output target of the query process is taken, then proceeds to the third stage, namely the preparation of the query command to get the output target in the second stage. At the same time, the fourth stage is testing the query command. Each query command will be executed ten times to see the average execution time in this fourth stage.

In comparison, the fifth stage is recording the results of the execution trials. The fourth and fifth stages will run on two different hardware specifications. These stages can be illustrated in Figure 1, and the description of the illustration will be presented in more detail in the sub-chapters 2.3.1 and 2.3.2.

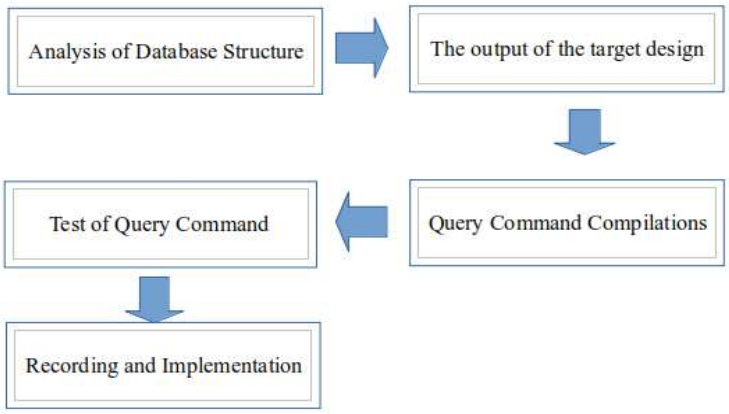

Figure 1 Experiment Stage

2.3.1 Database Structure

Database structure analysis is a structural analysis carried out on raw data arranged into several tables in the database structure. The tables needed are shown in Figure 2.

The database structure shown in Figure 2 is involved in the query optimization process. Each table is, on average, one-to-one related. Only the alumni table is related one-to-many.

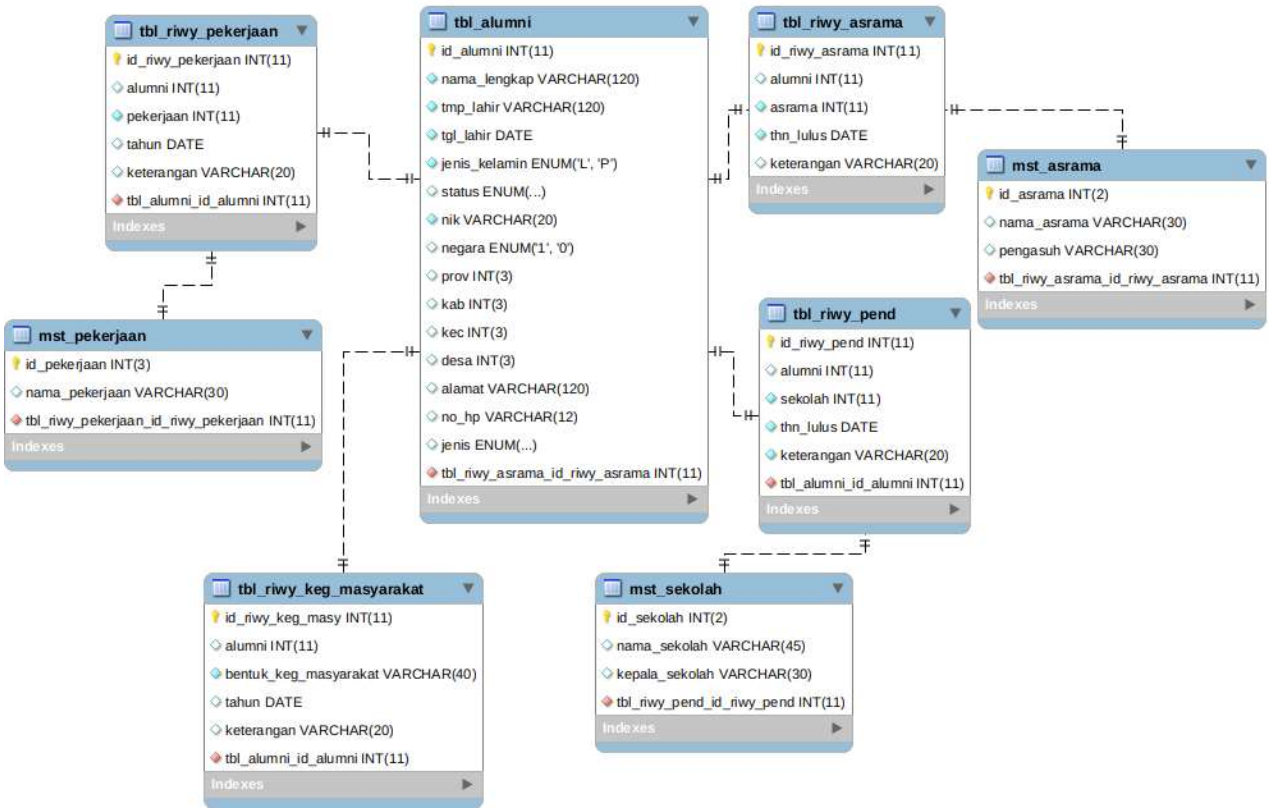

Figure 2 Database Structure

2.3.2 Design of Output and Query Command

The output design stage is made to formulate the desired output. At this stage, 5 data output targets are arranged to perform query optimization experiments. They are the expected targets, namely: 1. Graduate data based on educational history, 2. Graduate data based on dormitory history, 3. Graduate data based on work history, 4. Graduate data based on community activity history, 5. Graduate data with education, work, dormitory, and school history.

The command used to generate the output target will be helpful in the following few paragraphs. To get the first output target, graduate data based on education history with the statement "where," you can use the following command: Select * from tbl_alumni join tbl_riwy_pend where tbl_riwy_pend.alumni="23".

While to get the second output target, namely graduate data based on the history of the dormitory, by applying the statement "where," you can use the following query command: SELECT * from tbl_alumni, join tbl_riwy_asrama where tbl_riwy_asrama.alumni="25".

When we want to get the third output target, graduate data based on employment history, by applying the statement "where," you can use the following query command: SELECT * from tbl_alumni join tbl_riwy_jobs where tbl_riwy_jobs.alumni="27".

The following output target is graduate data, based on the history of activities in the community. To get this data, we can use the query command like this: select * from tbl_alumni join tbl_riwy_keg_community where tbl_riwy_keg_community.alumni="30."

The next output target is to do a joint between several tables to get graduate data based on education history, dormitory history, work history, and activity history in the community. To get this data, we can use the query command like

this : select * from tbl_alumni join tbl_riwy_pend on tbl_riwy_pend.alumni = tbl_alumni.id_alumni join tbl_riwy_asrama on tbl_riwy_asrama.alumni = tbl_alumni.id_alumni join tbl_riwy_pekerjaan on tbl_riwy_pekerjaan.alumni = tbl_alumni.id_alumni join tbl_riwy_keg_masyarakat on tbl_riwy_keg_masyarakat .alumni = tbl_alumni.id_alumni.

One way of optimizing the query is by using a subquery. To get the output data like the previous query by applying a subquery, we can use a query command like this.

SELECT id_alumni, nama_lengkap, subquery1.nama_sekolah, subquery1.thn_lulus AS tahun_lulus_pend, subquery2.nama_asrama, subquery2.thn_lulus AS tahun_lulus_asrama, subquery3.nama_pekerjaan, subquery3.tahun AS tahun_kerja, subquery4.bentuk_keg_masyarakat, subquery4.tahun AS tahun_keg FROM tbl_alumni (SELECT alumni, nama_sekolah, thn_lulus FROM tbl_riwy_pend JOIN tbl_alumni ON tbl_riwy_pend.alumni = tbl_alumni.id_alumni JOIN mst_sekolah ON tbl_riwy_pend.sekolah = mst_sekolah.id_sekolah) AS subquery1, (SELECT alumni, nama_asrama, thn_lulus FROM tbl_riwy_asrama JOIN tbl_alumni ON tbl_riwy_asrama.alumni = tbl_alumni.id_alumni JOIN mst_asrama ON tbl_riwy_asrama.asrama = mst_asrama.id_asrama) AS subquery2, (SELECT alumni, nama_pekerjaan, tahun FROM tbl_riwy_pekerjaan JOIN tbl_alumni ON tbl_riwy_pekerjaan.alumni = tbl_alumni.id_alumni JOIN mst_pekerjaan ON tbl_riwy_pekerjaan.pekerjaan = mst_pekerjaan.id_pekerjaan) AS subquery3, (SELECT alumni, bentuk_keg_masyarakat, tahun FROM tbl_riwy_keg_masyarakat JOIN tbl_alumni ON tbl_riwy_keg_masyarakat.alumni = tbl_alumni.id_alumni) AS subquery4 WHERE tbl_alumni.id_alumni = subquery1.alumni AND tbl_alumni.id_alumni = subquery2.alumni AND tbl_alumni.id_alumni = subquery3.alumni AND tbl_alumni.id_alumni = subquery4.alumni.

The hardware specifications used to test the query command above can be described in Table 1.

Table 1 The hardware Specifications

No.	Description	Value
1	Operating System	Linux Mint 19.3 Tricia
2	Processor	Intel(R) Core (TM) i5-2520M CPU @ 2.50GHz
3	RAM	DDR3 4 GB
4	Operating System	Windows 7 Starter SP 1
5	Processor	Intel(R) Core (TM) i5-2520M CPU @ 2.50GHz
6	RAM	DDR3 1,5 GB

3 Result and Discussion

Based on the query testing process that has been carried out, it can be described in the following discussion. The first query was tested ten times to get the average execution time. This query command was run on two different hardware platforms to see the difference in execution time. The execution time can be presented in Figure 3.

In the first query command test, the index implementation was run on two different hardware platforms. It has a difference in execution time of 0.0781 seconds. At the same time, the query that does not use the index has a difference in execution time of 0.011 seconds.

In the second query test, the index implementation between the first and second hardware has an average execution time difference of 0.0889 seconds. As for queries on tables that do not apply indexes, the average execution time difference is 0.0525 seconds. The detailed results can be seen in Figure 4.

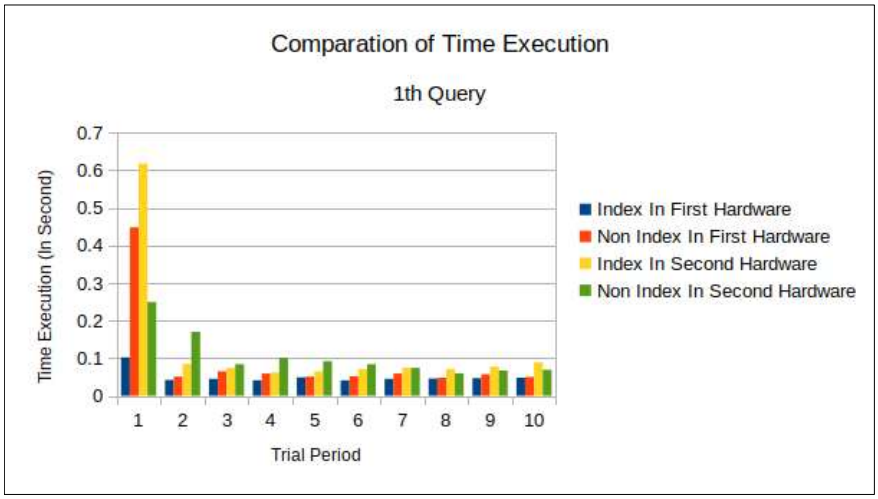

Figure 3 First Query Process Outcome

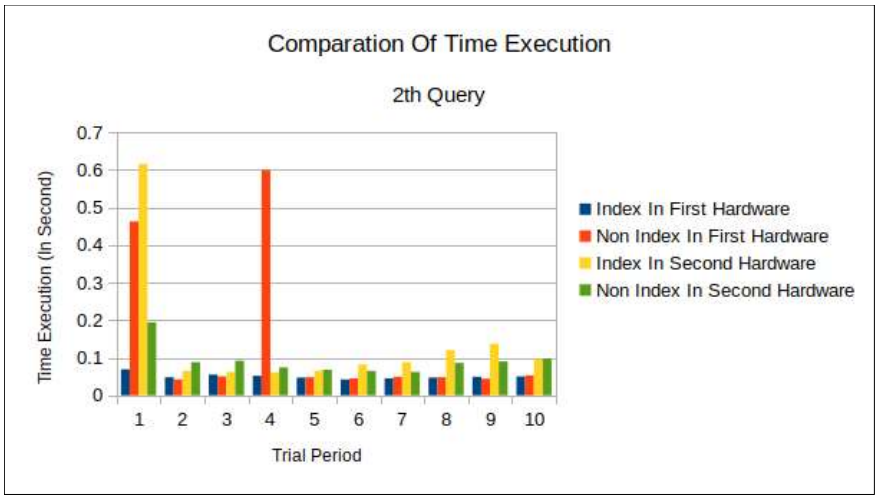

Figure 4 Second Query Process Outcome

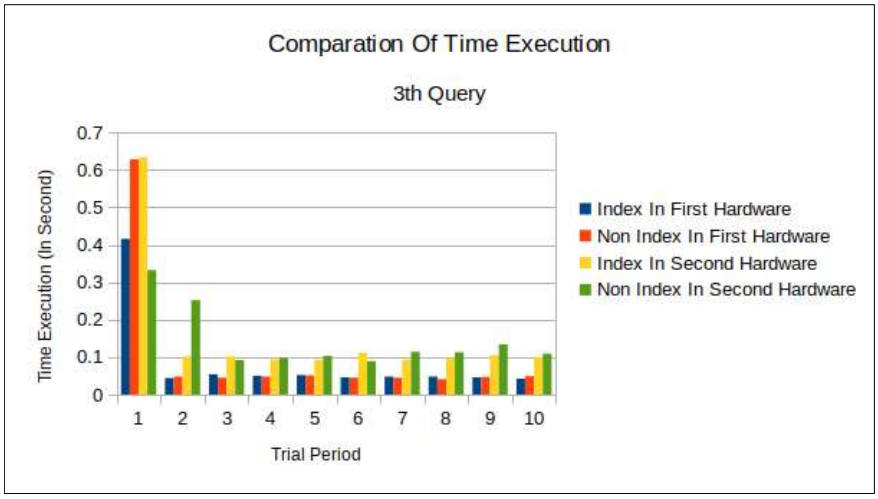

Figure 5 Third Query Process Outcome

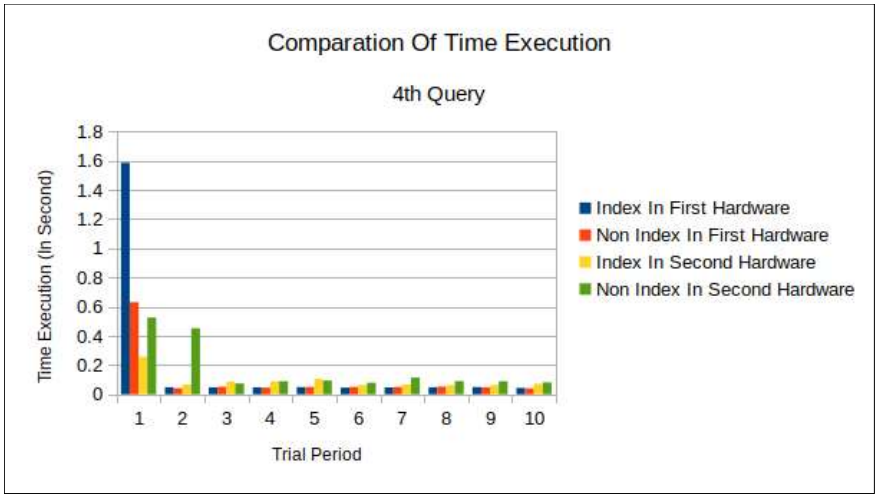

Figure 6 Fourth Query Process Outcome

The third query test has little difference from the previous test. It was only different in the join table. The average difference between the first and second hardware index implementations was 0.0685 seconds. In comparison, those who do not use the index have a difference in execution time of 0.0389 seconds. Meanwhile, the average difference between queries using the index on the fourth query command test was 0.1082 seconds. The execution time of each test is presented in Figure 5 and Figure 6.

In testing the fifth query command, we test a five-tables join. If we pay attention, the execution time between the first and tenth tests has a longer execution time than the previous. However, the resulting pattern is still the same: the index application can reduce execution time, although it may not be very significant. If we pay attention, the factors that affect the execution time of the query command are not limited to the application of the index but also the hardware specifications, especially hardware related to data processing. The results of testing the fifth query command are presented in Figure 7.

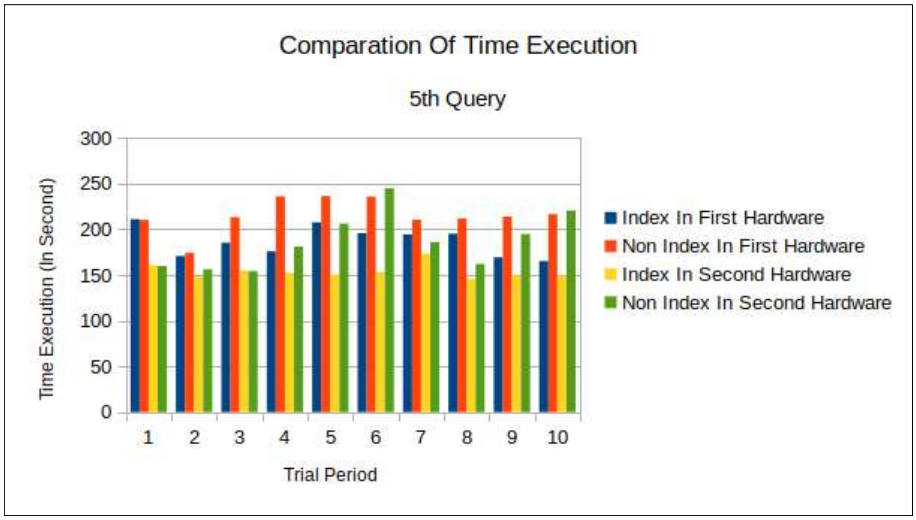

Figure 7 Fifth Query Process Outcome

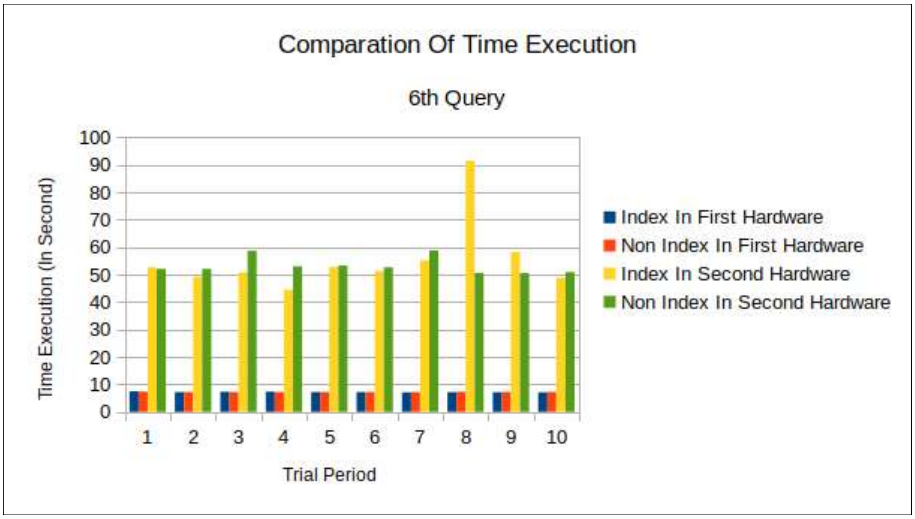

Figure 8 Sixth Query Process Outcome

In Figure 7, the execution time is above 100 seconds because the fifth query command does not apply data conditions using a 'where' clause. An application of the subquery in the last test combined with the index can be made. However, execution time is greatly influenced by the RAM capacity because the subquery requires a temporary storage medium that is large enough to accommodate the results of the subquery before being executed on the central query command. The details of execution time can be seen in Figure 8.

4 Conclusions

Based on the discussion above, some conclusions can be drawn as follows.

1. The structure of the query command and the table that is read affect the execution process of the query command.
2. The database normalization stage must be carried out so that the preparation of query commands can be carried out correctly and efficiently.
3. Application of index as a stage in database optimization can be done but must be accompanied by determining the right field.
4. The optimization factor is not limited to the application of indexes and subqueries but is also influenced by hardware specifications.

Bibliography

[1] M. M. Dewi and N. Rezeki, "Analisis Perbandingan Optimasi Query Nasted Join dan Hash Join pada MySQL Server," CSRID (Computer Sci. Res. Its Dev. Journal), vol. 9, no. 1, p. 31, 2017, doi: 10.22303/csrid.9.1.2017.31-42.

[2] N. Hartono, E. Utami, and A. Amborowati, "Migrasi dan Optimalisasi Database Sistem Informasi Manajemen Universitas Cokroaminoto Palopo," J. Buana Inform., vol. 7, no. 4, pp. 255–264, 2016, doi: 10.24002/jbi.v7i4.766.

[3] T. Hastono, "Optimasi Query Sistem Informasi Menggunakan Stored Procedure," J. Din. Inform., vol. 8, no. 2, pp. 79–89, 2019, [Online]. Available: https://jdi.upy.ac.id/index.php/jdi/article/view/33/29.

[4] F. Inastiana, A. Triayudi, and E. T. E. Handayani, "Implementation of the Waterfall Method for Designing Sisar (Archive Information System) at the National University," J. Mantik, vol. 3, no. January, pp. 31–38, 2020, [Online]. Available: https://iocscience.org/ejournal/index.php/mantik/article/view/832.

[5] F. S. Nugraha, F. H. Purwanto, and F. E. N. Saputro, “Optimasi Query pada Laporan Transaksi Penjualan menggunakan Materialized View (Studi Kasus : Moonly café),” CCIT J., vol. 11, no. 1, pp. 1–14, 2018, doi: 10.33050/ccit.v11i1.552.

[6] P. K. Safitri, W. W. Winarno, and E. Pramono, “Optimasi Query untuk Sistem Informasi Penjadwalan Mata Pelajaran Sekolah Menggunakan View (Studi Kasus: SMK VIP Purworejo),” Respati, vol. XIII, pp. 46–51, 2018, [Online]. Available: http://jti.respati.ac.id/index.php/jurnaljti/article/view/259.

[7] R. Pamungkas, “Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index,” Res. Comput. Inf. Syst. Technol. Manag., vol. 1, no. 1, p. 27, 2018, doi: 10.25273/research.v1i1.2453.

[8] Y. A. Bismo Raharjo, D. Mantriwira, and F. S. Sumanto, “Analisis Kinerja Optimasi Query Cross Join, Natural Join Dan Full Outer Join,” CSRID (Computer Sci. Res. Its Dev. Journal), vol. 9, no. 2, p. 106, 2018, doi: 10.22303/csrid.9.2.2017.106-115.

[9] Ermatita, “Analisis Optimasi Query Pada Data Mining,” J. Sist. Inf., vol. 1, no. 1, pp. 47–54, 2009.