Database optimization for improved system performance and response time of hospital management information system

Bayu Rahayudi [1,*], Nurizal Dwi Priandani [2], Buce Trias Hanggara [3],Wayan Firdaus Mahmudy [4]

Universitas Brawijaya, Malang, Indonesia
[1] ubay1@ub.ac.id*; [2] priandani@ub.ac.id; [3] buce_trias@ub.ac.id; [4] wayanfm@ub.ac.id
* corresponding author

ARTICLE INFO

Article history
Received August 5, 2021
Revised August 15, 2021
Accepted August 30, 2021

Keywords
SIMRS Analysis
System Profiling
Actual Execution Plan
Analysis
Response Time Measurement

ABSTRACT

A Regional Hospital in East Java has implemented a Hospital Management Information System, namely SIMRS, in their data management system but has experienced problems in the form of slow system response when accessed by many users, was experienced in the last years when the system had been running for four years since 2016. The system's slow response causes hospital services to be disrupted and also the quality of service to decline. So, an analysis to the existing database system is carried out, which includes an analysis of the system's database performance. Since many SIMRS use database servers on their data processing, then their applications will be based on executing queries and stored procedures (most of the queries are stored in stored procedures). So that, analysis of those queries will be carried out. The optimization process will include analyzing and mapping the database's queries, profiling, and analyzing the Actual Execution Plan. By doing so, it is known which parts of the query are causing a decrease in performance and time system response. Based on the analysis results, recommendations are given for improving and rewriting several stored procedures and query statements, and the system response time is getting better.

1. Introduction

The Hospital Management Information System (*Sistem Informasi Manajemen Rumah Sakit /SIMRS*) is an arrangement that deals with data collection, data management, information presentation, analysis and conclusion of information, and delivery of information needed by stakeholders regarding hospital activities [1]. In general, SIMRS includes clinical information systems, administrative information systems, and management information systems. SIMRS is an integrated information system prepared to handle the entire hospital management process, from services, diagnosis, and action for patients, medical records, pharmacies, pharmacy warehouses, billing, personnel database, employee payroll, accounting processes to control by management [2][3].

The Hospital Management Information System can be utilized by all general hospitals, both managed by government and privately as regulated in the Law of the Republic of Indonesia Number 44 of 2009 concerning Hospitals, and also in Regulation of the Minister of Health No. 82 of 2013 concerning Hospital Management Information Systems [4][5][6]. In the latest, article 1 paragraph 6 states that the function of SIMRS is to increase efficiency, effectiveness, professionalism, performance, and access to hospital services.

Information systems play an important role in the production, sharing, storage and transmission of information in various fields, especially in Hospital Management Information Systems [7]–[11]. A good SIMRS system, information, and service quality satisfy its users [12]–[17]. The system quality included a system that can function properly and respond quickly to its users [18][19]. The fast

https:doi.org/10.31763/businta.v5i2.491

response will depend on the amount of data stored in the database and commands for processing and retrieving data [19]. As time goes by, where there is additional data on SIMRS, it will affect the system response time. SIMRS, which have and store large amounts of data, have the potential for slower response times as more data is stored. The implementation of a good design and database system will affect the system's performance [20]. Asyary [21] also conducted research that aims to determine the use of SIMRS from the user's point of view with the Technology Acceptance Model (TAM) measurement method in a case study at a maternity hospital in Lampung, Indonesia. These studies provide an overview of the use of information system features and user workflows in the currently implemented information systems considered not optimal.

Many papers showed research about database optimizations and said that database optimization is an important process that should be done when building information systems [22]–[25]. Zhang [25] said that Optimizing database systems plays a vital role even though it is a very complex task. Satoto [22] discusses how to optimize the database system so that when the data is accessed does not affect the performance of server systems. Process optimization is done on the design of the database system. He said that database design plays a vital role in determining system performance. Wankhade [23] in his paper discusses about the importance, objectives and different approaches to query optimization. He tried to summarize some of the helpful query optimization techniques focused on common query constructs. Dorottya et.al [26] discusses the aspects of optimization performance related to data access in a database containing soy and corn-based products to ensure a quick search in the database. The result was database optimization techniques had been derived to address the issues that may limit the performance of a database to an extent of vulnerability.

One of the Regional Hospital (RSUD in the Indonesian Language) in East Java has also used SIMRS to manage his hospital and experienced a similar problem: slow response time to SIMRS users. SIMRS slow response causes hospital services to decline. For this reason, an analysis of the existing SIMRS is carried out to find out the problems so that the system can be optimized and system response time can be increased.

2. Research Method

The research methodology carried out in this research is following the concept which depicted in Fig. 1. The activity begins with a background description, objectives determination and purposes of the research. After that, data collection is carried out, and followed by analyzing data that has been collected. Based on the analysis, the recommendation is proposed to make system improvements and better system response time.

2.1. Method of collecting data

Data collection activities is carried out by collecting data through primary data surveys and secondary data observation. Primary data collection was carried out by making direct observations at the RSUD, by looking at the location of the server and client using SIMRS, and observing the process of using SIMRS by users. Secondary data collection is carried out through literature or literature studies and related information to systems analysis activities. At this stage, data collection is carried out through manual book, application guides, and reports on SIMRS.

2.2. System Analysis

The system analysis stage is carried out by analyzing the existing system, by collecting data on a running system, including system functions, system input-output, processes, databases used [27]. An analysis also includes a performance tuning process that consists of identifying performance bottlenecks, prioritizing the issues, troubleshooting their causes, applying different resolutions, quantifying performance improvements, and then repeating the whole process again and again [28]. In addition, data collection is also carried out on several hardware devices such as a set of computers, network equipment used, and analysis of the software used in SIMRS.

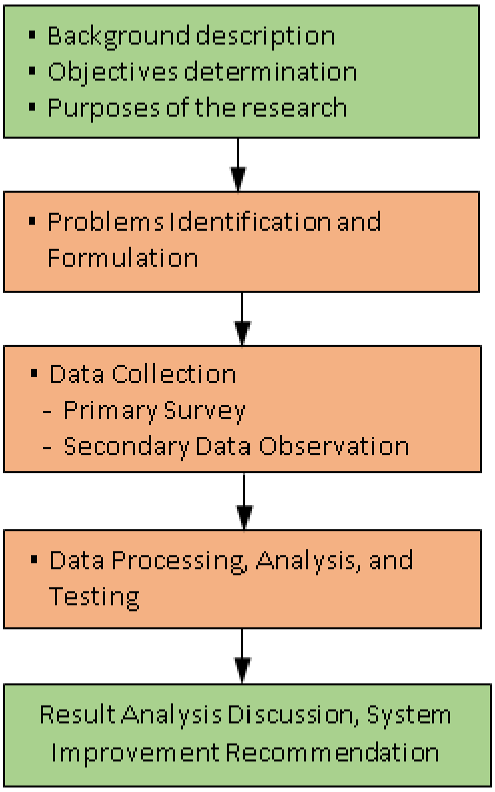

▪ Background description
▪ Objectives determination
▪ Purposes of the research
▪ Problems Identification and Formulation
▪ Data Collection
- Primary Survey
- Secondary Data Observation
▪ Data Processing, Analysis, and Testing
Result Analysis Discussion, System Improvement Recommendation

Fig. 1. Research Methodology used in this research

2.3. Optimization and System Improvement

From the results of the analysis, recommendations and system improvement are carried out into computer code. Improvement is carried out by optimizing the database design, including improving the relations between tables in the database and improving query and stored procedure codes.

2.4. System Testing

System testing is carried out to ensure that the system is running according to the results of the analysis and optimization. At this stage, functional testing and performance measurement of each process is carried out to ensure that all functions are running, all input entered has been validated and the output displayed has met the needs.

3. General Description of RSUD X SIMRS

SIMRS on one of RSUD in East Java was implemented for the first time in January 2016. The SIMRS is an application specially designed using the ERP (*Enterprise Resource Planning*) concept by integrating all units/sections/installations in the hospital, both medical and non-medical services. This application consists of 36 modules and was developed with the Visual Basic Programming Language and used a database implemented using the Microsoft SQL Server DBMS. The database developed consists of several objects, including 1210 tables, 1106 stored procedures, 2936 views and 212 functions.

Based on the initial analysis carried out on the SIMRS, it is known that two SIMRS application modules resulted in poor performance and slow response time. Both modules are included in the main module of SIMRS. These modules are the Pharmacy Logistics Module and the Medical Records Module. These two modules give a large load to the system so that when the two modules are run, the system's performance will decline significantly.

4. System Testing and Analysis

For system testing and verification, analysis will be carried out on two SIMRS modules already mentioned. Process mapping, profiling techniques, and execution plan analysis will be carried out on the query procedures and functions contained in the modules.

4.1. Pharmacy Logistics Module Analysis

This pharmacy logistics module is a SIMRS module which functions is to process data in pharmacies and drug storage. This module's mostly process is checking (data loading), updating, and saving data for prescriptions and medicines.

The initial step of the module analysis is to map the processes that occur in the module. One of the results of the process mapping can be seen in Fig. 2. It is present the process in the service of medicine prescription, whereas the process involves several queries and stored procedures. After mapping the process, they are then followed by analyzing the process and measuring processing time of the query using the Profiler. And then continuing the analysis using the Actual Execution Plan. The process of query monitoring is done using MS SQL Profiler, which is carried out in the frmUseObatAlkes Object. This object accessed the drug storage database and triggered several queries and stored procedure. The monitoring results can be seen in Fig. 3, and it is found that the execution of (SP)Add_PemakaiObatAlkesResepNew (stored procedure that checking medicine stocks) and also (SP) Add_Generate_ResepObat (stored procedure that updates medicine stocks) are the processes that consume the largest resources.

The next step is analyzing Actual Execution Plan. The analysis targeted (SP) Add_PemakaiObatAlkesResepNew and (SP)Add_GenerateRe sepObat. The analysis results of the Actual Execution Plan show that there are queries with the high cost. The first one had cost of 100%, i.e (SP)Add_GenerateResepObat, and the second one had a query load of 90% on (SP)Add_PemakaiObatAlkesResepNew.

The result of the analysis shows that previously problems described are caused by computation time and cost that is carried out in saving and loading data. If this process can be refined and optimized, then the problems can be resolved. Process refining and optimization will reduce computation time and cost and improve the response time of the system. The process of improvement and recommendations for this module are summarized and explained in the recommendation section below.

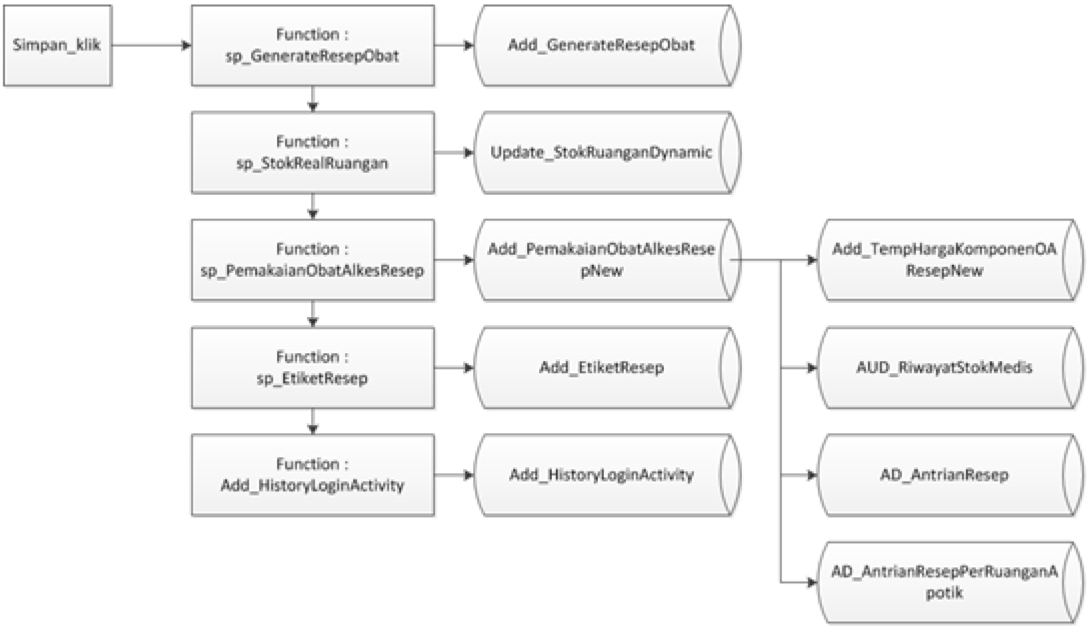
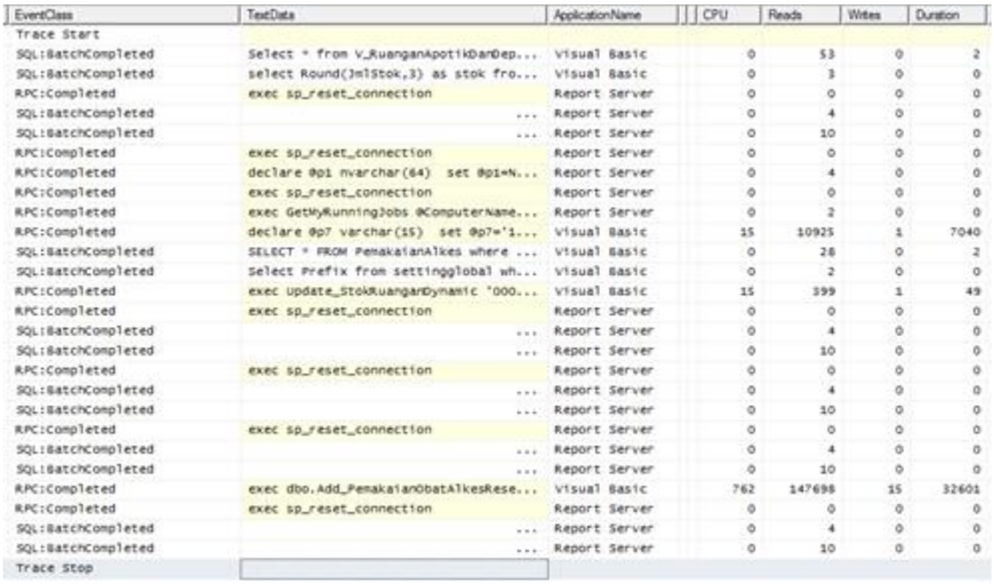

Simpan_klik
Function : sp_GenerateResepObat
Add_GenerateResepObat
Function : sp_StokRealRuangan
Update_StokRuanganDynamic
Function : sp_PemakaianObatAlkesResep
Add_PemakaianObatAlkesResepNew
Add_TempHargaKomponenOAResepNew
Function : sp_EtiketResep
Add_EtiketResep
AUD_RiwayatStokMedis
Function : Add_HistoryLoginActivity
Add_HistoryLoginActivity
AD_AntrianResep
AD_AntrianResepPerRuanganApotik

Fig. 2. Results of Pharmacy Logistics Module Process Mapping



EventClass	TextData	ApplicationName	CPU	Reads	Writes	Duration
Trace Start						
SQL:BatchCompleted	Select * from V_RuanganApotikDanDep...	Visual Basic	0	53	0	2
SQL:BatchCompleted	select Round(JmlStok,3) as stok fro...	Visual Basic	0	3	0	0
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
SQL:BatchCompleted	...	Report Server	0	4	0	0
SQL:BatchCompleted	...	Report Server	0	10	0	0
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
RPC:Completed	declare @p1 nvarchar(64) set @p1=N...	Report Server	0	4	0	0
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
RPC:Completed	exec GetMyRunningJobs @ComputerName...	Report Server	0	2	0	0
RPC:Completed	declare @p7 varchar(15) set @p7='1...	Visual Basic	15	10925	1	7040
SQL:BatchCompleted	SELECT * FROM PemakaianAlkes where ...	Visual Basic	0	28	0	2
SQL:BatchCompleted	Select Prefix from settingglobal wh...	Visual Basic	0	2	0	0
RPC:Completed	exec Update_StokRuanganDynamic '000...	Visual Basic	15	399	1	49
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
SQL:BatchCompleted	...	Report Server	0	4	0	0
SQL:BatchCompleted	...	Report Server	0	10	0	0
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
SQL:BatchCompleted	...	Report Server	0	4	0	0
SQL:BatchCompleted	...	Report Server	0	10	0	0
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
SQL:BatchCompleted	...	Report Server	0	4	0	0
SQL:BatchCompleted	...	Report Server	0	10	0	0
RPC:Completed	exec dbo.Add_PemakaianObatAlkesRese...	Visual Basic	762	147698	15	32601
RPC:Completed	exec sp_reset_connection	Report Server	0	0	0	0
SQL:BatchCompleted	...	Report Server	0	4	0	0
SQL:BatchCompleted	...	Report Server	0	10	0	0
Trace Stop						

Fig. 3. Profiler results for the Pharmacy Logistics Module

4.2. Medical Records Module Analysis

Similar to the process carried out in the Pharmacy Logistics module, it is also carried out to the Medical Record module. This medical record module will record (update and save) the patient's medical history and treatment data through related queries and stored procedures. The results of mapping process on tables, stored procedures and functions of the module are shown in Fig. 4, while the results of the profiling process can be seen in Fig. 5.

It is shown in Fig. 5 that the execution process of (SP)Add_BiayaPela yananOtomatisNew (stored procedure that calculate medical treatment cost), Add_RegistrasiPasienMRS (stored procedure that add/update patient's record data) and Add_PasienMasukKamar (query that add/update data for impatient data) are processes that consumes the most resources. Based on analyzing Actual Execution Plan in the Medical Record module, it is shown that query with the highest cost is on the 6th, 7th and 8th queries with a query load of 16%, 27% and 58% respectively on (SP)Add_BiayaPelayanan_Automatic_New.

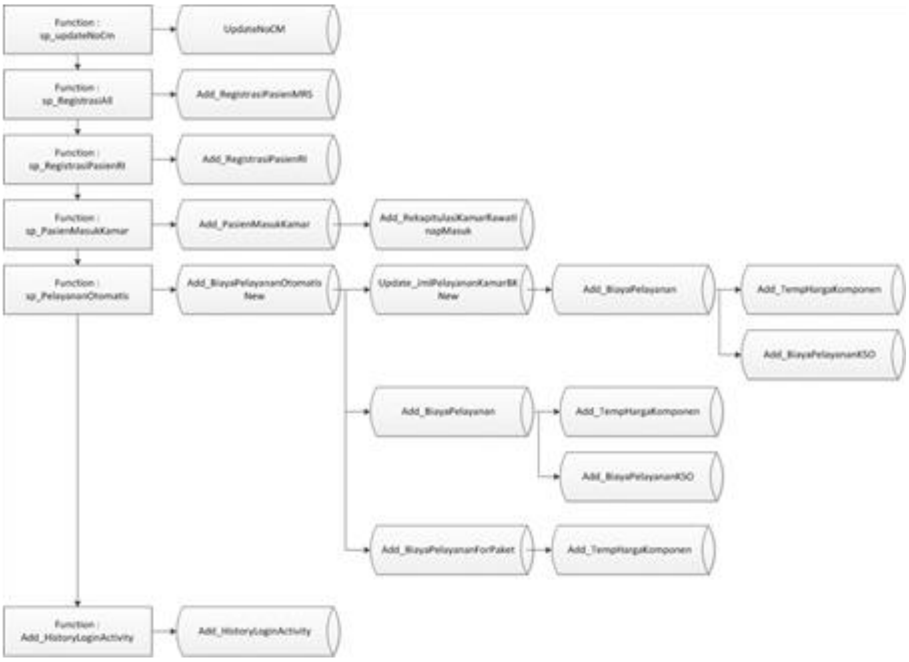
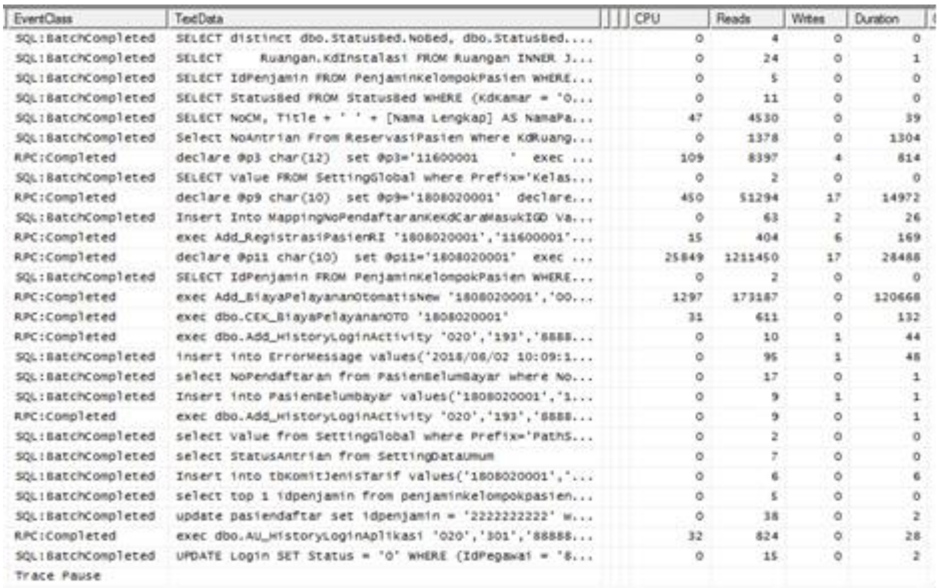

Function : sp_updateNoCm
UpdateNoCM
Function : sp_RegistrasiAll
Add_RegistrasiPasienMRS
Function : sp_RegistrasiPasienRI
Add_RegistrasiPasienRI
Function : sp_PasienMasukKamar
Add_PasienMasukKamar
Add_RekapitulasiKamarRawatInapMasuk
Function : sp_PelayananOtomatis
Add_BiayaPelayananOtomatisNew
Update_JmlPelayananKamarBKNew
Add_BiayaPelayanan
Add_TempHargaKomponen
Add_BiayaPelayananKSO
Add_BiayaPelayanan
Add_TempHargaKomponen
Add_BiayaPelayananKSO
Add_BiayaPelayananForPaket
Add_TempHargaKomponen
Function : Add_HistoryLoginActivity
Add_HistoryLoginActivity

Fig. 4. Result of Mapping Process on Medical Record module



EventClass	TextData	CPU	Reads	Writes	Duration
SQL:BatchCompleted	SELECT distinct dbo.StatusBed.NoBed, dbo.StatusBed....	0	4	0	0
SQL:BatchCompleted	SELECT Ruangan.KdInstalasi FROM Ruangan INNER J...	0	24	0	1
SQL:BatchCompleted	SELECT IdPenjamin FROM PenjaminKelompokPasien WHERE...	0	5	0	0
SQL:BatchCompleted	SELECT StatusBed FROM StatusBed WHERE (KdKamar = 'O...	0	11	0	0
SQL:BatchCompleted	SELECT NoCM, Title + ' ' + [Nama Lengkap] AS NamaPa...	47	4530	0	39
SQL:BatchCompleted	Select NoAntrian From ReservasiPasien where KdRuang...	0	1378	0	1304
RPC:Completed	declare @p3 char(12) set @p3='11600001 ' exec ...	109	8397	4	814
SQL:BatchCompleted	SELECT Value FROM SettingGlobal where Prefix='Kelas...	0	2	0	0
RPC:Completed	declare @p9 char(10) set @p9='1808020001' declare...	450	51294	17	14972
SQL:BatchCompleted	Insert Into MappingNoPendaftaranKeKdCaraMasukIGD Va...	0	63	2	26
RPC:Completed	exec Add_RegistrasiPasienRI '1808020001','11600001'...	15	404	6	169
RPC:Completed	declare @p11 char(10) set @p11='1808020001' exec ...	25849	1211450	17	28488
SQL:BatchCompleted	SELECT IdPenjamin FROM PenjaminKelompokPasien WHERE...	0	2	0	0
RPC:Completed	exec Add_BiayaPelayananOtomatisNew '1808020001','00...	1297	173187	0	120668
RPC:Completed	exec dbo.CEK_BiayaPelayananOTO '1808020001'	31	611	0	132
RPC:Completed	exec dbo.Add_HistoryLoginActivity '020','193','8888...	0	10	1	44
SQL:BatchCompleted	insert into ErrorMessage values('2018/08/02 10:09:1...	0	95	1	48
SQL:BatchCompleted	select NoPendaftaran from PasienBelumBayar where No...	0	17	0	1
SQL:BatchCompleted	Insert into PasienBelumbayar values('1808020001','1...	0	9	1	1
RPC:Completed	exec dbo.Add_HistoryLoginActivity '020','193','8888...	0	9	0	1
SQL:BatchCompleted	select value from SettingGlobal where Prefix='PathS...	0	2	0	0
SQL:BatchCompleted	select StatusAntrian from SettingDataUmum	0	7	0	0
SQL:BatchCompleted	Insert into tbKomitJenisTarif values('1808020001','...	0	6	0	6
SQL:BatchCompleted	select top 1 idpenjamin from penjaminkelompokpasien...	0	5	0	0
SQL:BatchCompleted	update pasiendaftar set idpenjamin = '2222222222' w...	0	38	0	2
RPC:Completed	exec dbo.AU_HistoryLoginAplikasi '020','301','88888...	32	824	0	28
SQL:BatchCompleted	UPDATE Login SET Status = '0' WHERE (IdPegawai = '8...	0	15	0	2
Trace Pause					

Fig. 5. Profiling results of the Medical Record module

4.3. Discussion and Recommendations on Applications and Databases

Based on the analysis of the Pharmacy Logistics and Medical Records module, a comprehensive and global recommendation can be obtained, which can also be implemented for all modules on SIMRS of the RSUD to improve existing information systems' performance and response time.

For optimization of unnecessary queries, the query algorithm tracing is carried out in the analysis phase to find out what steps are carried out in the stored procedure. In this phase, several queries were found that hindered the data exchange process, for example:

- A SELECT table process is inserted into the variables, but the variables used are not processed in any process. This results in wasted CPU and memory usage. The recommendation given is to delete this SELECT query.
- There is an update query where the condition of the selection process does not return a value, so no rows are updated, but this query resulted in a long process. The recommendation is to add the If-Else conditional command to the query, which will not be updated if it does not return a value.
- Use the SELECT and WHERE commands to get data that only has one value. The recommendation given is to replace the command with the SELECT TOP 1 command which will return one value on hundreds or millions of rows of data.

One of the problems that arise and slowdown the processes is an inefficient looping process. It is found a process contained in the Medical Record module in (SP) ADD_rekapitulasiKamarRawatInap (stored procedure that calculates and count for room that is occupied and unoccupied) that was carried out about 700 times through looping, which was obtained from the select count query process in view V_InformasiKamarRawatInap (view/query that process impatient room records). After further analysis, it was found that the 700 data results were duplicated, that the really needed data was only about 100 data. Thus,it is recommended to use the SELECT DISTINCT command when retrieving data to the V_InformasiKamarRawatInap view.

Performance problems also occur using aggregate functions, namely the MAX and COUNT commands, which are used to retrieve data that only produces one value. These problems can be solved by applying the concept of Lightly Summarized Data, a simple concept of a data warehouse. In the Pharmacy Logistics module, a process is needed to collect the maximum queue number per day. To ease the read process, both in terms of memory and CPU, a summary table is created called CounterAntrianResepSeq (table that record patient's queue number) which contains only 2 attributes, namely date and NoMaxAntriResep (maximum patient's number that can be hadled). The data stored in the table is the maximum queue no data per day which will continue to be updated when a new queue arrives. Unlike the previous query that requires the MAX function to calculate the maximum value of several rows of data, this process only execute a SELECT statement. There is a very significant difference in the number of process, which if in the previous table the process will access more than 100,000 rows, then in the lightly summarized table the process will only access 100 rows.

5. Performance Evaluation after Implementing Recommendations

After implementing the recommendations, by rewriting some queries and repairing programs in Logistics Pharmacy module and Medical Record module, The SIMRS performance increase significantly. After recommendations for Logistics Pharmacy module are implemented, the total cost that can be reduced is around 95% for the entire process in the. And after recommendations for Medical Record module are implemented, the total cost that can be reduced is around 93% for the entire process. So the overall performance of the systems after recommendation implementation increase by 94%. This improvement has a significant effect on the SIMRS service to customers, which can reduce queue time and affect hospital employee response, that they can respond better (quicker) to customers.

6. Conclusions

This study analyzes the query process that occurs in several SIMRS modules at one of RSUD in East Java, which has decreased performance and response time. After analyzing the query, by mapping the process, profiling, and analyzing the Actual Execution Plan, it is known which parts of the query cause a decrease in system performance and response time, so that recommendations can be made to improve queries improve performance and response time. The recommendations include optimizing and eliminating unnecessary queries, looping process refinement that increased efficiency and reduced redundancy in query data resulting by system, and lastly, using variables and light summarized data in queries and stored procedures.

Acknowledgment

We would like to thank the IT team department of RSUD X in East Java for their help and cooperation when conducting this research.

References

[1] V. N. Helia, V. I. Asri, E. Kusrini, and S. Miranda, “Modified technology acceptance model for hospital information system evaluation – a case study,” *MATEC Web Conf.*, vol. 154, p. 01101, Feb. 2018, doi: 10.1051/matecconf/201815401101.

[2] P. R. Vegoda, “Introduction to hospital information systems,” *Int. J. Clin. Monit. Comput.*, vol. 4, no. 2, pp. 105–109, 1987, doi: 10.1007/BF02915853.

[3] N. I. Ismail, N. H. Abdullah, and A. Shamsuddin, “Adoption of Hospital Information System (HIS) in Malaysian Public Hospitals,” *Procedia - Soc. Behav. Sci.*, vol. 172, pp. 336–343, 2015, doi: 10.1016/j.sbspro.2015.01.373.

[4] Menteri Kesehatan Republik Indonesia, “Peraturan Menteri Kesehatan RI Nomor 82 tentang Sistem Informasi Manajemen Rumah Sakit,” *Peratur. Menteri Kesehat.*, no. 87, pp. 1–36, 2013.

[5] P. R. Indonesia, “Undang-undang Republik Indonesia Nomor 44 Tahun 2009,” *Undang. Republik Indones.*, vol. 44, no. 10, p. 41, 2009.

[6] P. W. Handayani, A. N. Hidayanto, D. Ayuningtyas, and I. Budi, “Hospital information system institutionalization processes in indonesian public, government-owned and privately owned hospitals,” *Int. J. Med. Inform.*, vol. 95, pp. 17–34, 2016, doi: 10.1016/j.ijmedinf.2016.08.005.

[7] D. Demirel, “Hospital Management Information Systems in Health Sector and Development in Turkey Journal of Current Researches on Health Sector Hospital Management Information Systems in Health Sector and Development in Turkey,” *J. Curr. Res. Heal. Sect. (J o C R e H e S)*, 2018.

[8] G. H. C. Sibuea, T. A. Napitupulu, and A. R. Condrobimo, “An evaluation of information system using HOT-FIT model: A case study of a hospital information system,” *Proc. 2017 Int. Conf. Inf. Manag.*

Technol. ICIMTech 2017, vol. 2018-January, no. November, pp. 106–111, 2018, doi: 10.1109/ICIMTech.2017.8273520.

[9] S. L. Hsieh *et al.*, “An integrated healthcare enterprise information portal and healthcare information system framework,” *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, no. December 2003, pp. 4731–4734, 2006, doi: 10.1109/IEMBS.2006.260715.

[10] T. Takemura, N. Kume, K. Okamoto, T. Kuroda, and H. Yoshihara, “Development of an environment for information reuse on a hospital information system,” *6th Int. Conf. Soft Comput. Intell. Syst. 13th Int. Symp. Adv. Intell. Syst. SCIS/ISIS 2012*, pp. 1384–1387, 2012, doi: 10.1109/SCIS-ISIS.2012.6505339.

[11] P. Balaraman and K. Kosalram, “E –Hospital Management & Hospital Information Systems – Changing Trends,” *Int. J. Inf. Eng. Electron. Bus.*, vol. 5, no. 1, pp. 50–58, 2013, doi: 10.5815/ijieeb.2013.01.06.

[12] N. Fadilla and W. Setyonugroho, “Sistem Informasi Manajemen Rumah Sakit Dalam Meningkatkan Efisiensi: Mini Literature Review,” *Tek. Inform. dan Sist. Inf.*, vol. 8, no. 1, 2021.

[13] P. W. Handayani, A. N. Hidayanto, A. A. Pinem, I. C. Hapsari, P. I. Sandhyaduhita, and I. Budi, “Acceptance model of a Hospital Information System,” *Int. J. Med. Inform.*, vol. 99, no. 1171, pp. 11–28, 2017, doi: 10.1016/j.ijmedinf.2016.12.004.

[14] M. Khalifa and O. Alswailem, “Hospital information systems (HIS) acceptance and satisfaction: A case study of a Tertiary Care Hospital,” *Procedia Comput. Sci.*, vol. 63, no. Icth, pp. 198–204, 2015, doi: 10.1016/j.procs.2015.08.334.

[15] F. R. Rumambi, A. J. Santoso, and D. B. Setyohadi, “Identification of factors influencing the Success of Hospital Information System (SIRS) by Hot-Fit model 2006: A case study of RSUD Dr Samratulangi Tondano, Minahasa Regency, North Sulawesi,” *Proc. - 2017 Int. Conf. Soft Comput. Intell. Syst. Inf. Technol. Build. Intell. Through IOT Big Data, ICSIIT 2017*, vol. 2018-January, pp. 202–207, 2017, doi: 10.1109/ICSIIT.2017.38.

[16] H. A. Anema, J. Kievit, C. Fischer, E. W. Steyerberg, and N. S. Klazinga, “Influences of hospital information systems, indicator data collection and computation on reported Dutch hospital performance indicator scores,” *BMC Health Serv. Res.*, vol. 13, no. 1, p. 1, 2013, doi: 10.1186/1472-6963-13-212.

[17] H. Yu, J. Li, H. Chen, X. Zhang, Y. Tian, and Y. Yang, “Performance evaluation of post-relational database in Hospital Information Systems,” *2nd Int. Work. Educ. Technol. Comput. Sci. ETCS 2010*, vol. 2, pp. 247–251, 2010, doi: 10.1109/ETCS.2010.134.

[18] J. Choi *et al.*, “Implementation of consolidated HIS: Improving quality and efficiency of healthcare,” *Healthc. Inform. Res.*, vol. 16, no. 4, pp. 299–304, 2010, doi: 10.4258/hir.2010.16.4.299.

[19] K. M. Kuo, C. F. Liu, P. C. Talley, and S. Y. Pan, “Strategic improvement for quality and satisfaction of hospital information systems,” *J. Healthc. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/3689618.

[20] B. Rahayudi, A. W. Widodo, C. Dewi, M. A. Rahman, and W. F. Mahmudy, “Design Spatial-based Information System using Flexible Field-Value Database∗,” *Proc. 2019 4th Int. Conf. Sustain. Inf. Eng. Technol. SIET 2019*, pp. 100–103, 2019, doi: 10.1109/SIET48054.2019.8986009.

[21] A. Asyary, A. K. Nur Prasetyo, T. Eryando, and S. Gerke, “Users’ perception of the hospital information system in a Maternity Hospital in Lampung, Indonesia,” *Kesmas*, vol. 14, no. 2, pp. 76–81, 2019, doi: 10.21109/kesmas.v14i2.2574.

[22] K. I. Satoto, R. R. Isnanto, R. Kridalukmana, and K. T. Martono, “Optimizing MySQL database system on information systems research, publications and community service,” *Proc. - 2016 3rd Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2016*, pp. 1–5, 2017, doi: 10.1109/ICITACEE.2016.7892476.

[23] P. Wankhade and V. Deshmukh, “An Overview of Query Optimization Techniques in Database Systems,” *Int. Res. J. Eng. Technol.*, pp. 1116–1119, 2008.

[24] G. C. A. L. Aponso, T. M. T. I. Tennakon, A. M. C. B. Arampath, S. Kandeepan, H. P. K. K. S. Amaratunga, and S. Lanka, “Database Optimization Using Genetic Algorithms for Distributed Databases,” *Int. J. Comput.*, vol. 24, no. 1, pp. 23–27, 2017.

[25] J. Zhang, “Research on Database Application Performance Optimization Method,” no. Mmebc, pp. 2236–2239, 2016, doi: 10.2991/mmebc-16.2016.448.

[26] D. C. Pamfil and R. Academy, “Database optimization techniques applied to a database contain soy and corn based products to ensure a quick search in this database.,” *Bull. Univ. Agric. Sci. Vet. Med. Cluj-Napoca. Hortic.*, vol. 69, no. 2, 2012, doi: 10.15835/buasvmcn-hort:8823.

[27] T. Sutabri, *Analisis Sistem Informasi*. Penerbit Andi.

[28] G. Fritchey, *SQL Server 2012 Query Performance Tuning*. Berkeley, CA: Apress, 2012.