

Optimization of Queries on Databases by Using the Ingres Algorithm

Nur Ihsan Nudin^{1*}, Agus Sifaunajah², Muhyiddin Zainul Arifin³

¹ Informatics, Universitas KH. A. Wahab Hasbullah

^{2,3} Information System, Universitas KH. A. Wahab Hasbullah

*Email: ihsann505@gmail.com

ABSTRACT

Bahrul Ulum Islamic Boarding School is the oldest cottage in Jombang district which was founded in 1830 AD until now. which of course has a very large number of students, both students and alumni status. The author wishes to research how to manage data in information systems with hundreds or millions of data quickly and easily. Proper query design is very influential on information system applications, if you only rely on conventional query techniques, the results needed to process queries are relatively slow, and the time required is also longer. The purpose of this research is to optimize the query using the ingres algorithm which will later be tested and compared the results with queries that have not been optimized or conventional to determine the extent to which the query process can be accelerated. The experimental results show that the execution time is more effective and faster using an optimized command using the Ingres algorithm with an average processing time of 0.13226 seconds, and an average request that has not been optimized 0.2079 seconds, in other words the result is 65% faster than unoptimized queries.

Keywords: *Ingres Algorithm; Query Optimization; Conventional; Relative; Experiment.*

INTRODUCTION

Along with the development of the times where humans live in the current technological era, many things can be facilitated by the existence of computerized technology. Data processing can be done more easily and quickly to produce the desired information, especially in obtaining data with hundreds or thousands of millions of data to process information. However, in an application, the larger the data stored in the database, the heavier the processing. So it is very necessary to have a speed of access that facilitates to process the information in it.

In the processing of data that the application suspends, MsqL can be used to retrieve data from the database. To produce the results we want, of course, it is very necessary to write an effective and efficient query or syntax command in order to get easy and fast results according to what we want. With this problem, the author wants to try to experiment with the application of the ingres algorithm for data optimization in the alumni data information system at the Bahrul Ulum Islamic Boarding School, Jombang tambakberas Jombang, the Ingres algorithm is one of the methods to optimize queries on a centralized database system, but in addition it can also be applied to a distributed database system.

The Ingres algorithm by using a dynamic query optimization method that recursively parses the query to get or break down smaller queries and then execute it sequentially (Özsu, Valduriez, 2011 in Wibowo et al., 2018).

Query Optimization is a process for analyzing queries to determine what sources are used by the query and whether the use of those sources can be reduced without changing the output. Or it can also be said that query optimization is a procedure to improve the evaluation strategy of a query to make the evaluation more effective. Query optimization includes several techniques such as transforming queries into the same form of logic, choosing the optimal access path and optimizing data storage (Siallagan et al., 2008).

Query optimization is a procedure to improve the evaluation strategy of a query to make the evaluation more effective. Query optimization includes several techniques such as transforming queries into the same form of logic, having the fastest access path and optimizing data storage (Dewi & Rezeki, 2017).

Queries have the ability to display data from the database for further processing which is usually taken from tables in the database and also queries can be defined as commands to access data on the database (Conoras & Kurnawan, 2017).

METHOD

The research method used is experimental research, the author uses Query Optimization in the Database using the Ingres Algorithm, the method used is the System Development Life Cycle (SDLC) methodology.

- **Research methods**

Waterfall model is "Linear Sequential Model". This model is sometimes referred to as the "classic life cycle". It is called waterfall because the step by step passed must wait for the completion of the previous stage and run sequentially. Waterfall demonstrates a systematic approach to software development (Prihatin et al., 2016). This model starts from the planning, analysis, design, coding, development, and deployment stages.

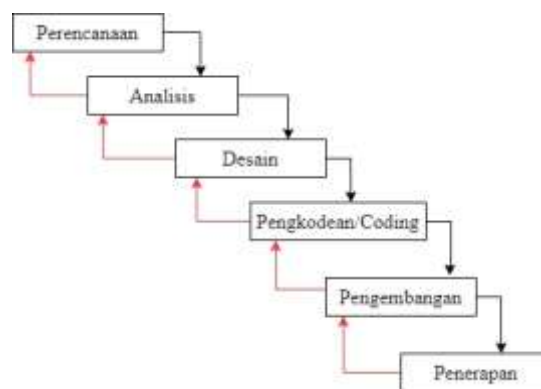


Figure 1. Waterfall Method The

- **Experimental Stages**

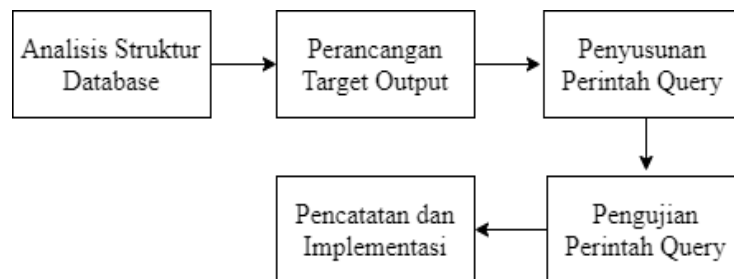


Figure 2. Experimental Stages

In the illustration picture above, it can be described as follows:

- **Database structure analysis**
Database structure analysis is a table structure that is carried out on raw data arranged into several tables in the database structure.
- **Designing target output**
The design of the target output is the result of a process that has been processed aimed at transforming data into information that can be seen more easily by users.
- **Command building queries**
At this stage, several query command models will be compiled based on the desired target output as described in the target output stage.
The explanation of the design of the output target that has been described above, I will create a query command script based on the output target which we will see the speed of time using query commands that have not been optimized and we compare with queries that have been optimized using the ingres algorithm ;
- **Query commands that have not been optimized.**

Table 1. Query time results that have not been optimized.

Query command script
<pre> SELECT nama_lengkap, alamat, mst_sekolah.nama_sekolah, tbl_riwy_pend.thn_lulus, mst_pekerjaan.nama_pekerjaan, tbl_riwy_pekerjaan.tahun, tbl_riwy_keg_masyarakat.bentuk_keg_masyarakat FROM tbl_alumni JOIN tbl_riwy_pend ON tbl_alumni.id_alumni = tbl_riwy_pend.id_alumni JOIN tbl_riwy_pekerjaan ON tbl_alumni.id_alumni = tbl_riwy_pekerjaan.id_alumni JOIN tbl_riwy_keg_masyarakat ON tbl_alumni.id_alumni = tbl_riwy_keg_masyarakat.id_alumni JOIN mst_sekolah ON tbl_riwy_pend.id_pendidikan = mst_sekolah.id_sekolah JOIN mst_pekerjaan ON tbl_riwy_pekerjaan.id_pekerjaan = mst_pekerjaan.id_pekerjaan WHERE tbl_riwy_pend.id_pendidikan = 7 OR tbl_riwy_pekerjaan.id_pekerjaan = 4 OR tbl_riwy_keg_masyarakat.bentuk_keg_masyarakat = 'mwc nu' </pre>

- The query command that has been optimized uses the ingres algorithm
 To get the optimized query command I need several stages ;

Table 2. Stages of Query Command Script

Query command script		
First Stage	The second stage	The third stage
<pre> SELECT tbl_riwy_pend.id_alumni, nama_lengkap,alamat, mst_sekolah.nama_sekolah, tbl_riwy_pend.thn_lulus FROM tbl_alumni JOIN tbl_riwy_pend ON tbl_alumni.id_alumni = tbl_riwy_pend.id_alumni JOIN mst_sekolah ON tbl_riwy_pend.id_pendidika n = mst_sekolah.id_sekolah </pre>	<pre> SELECT tbl_riwy_pekerjaan.id_alumni, nama_lengkap, alamat, mst_pekerjaan.nama_pekerjaan , tbl_riwy_pekerjaan.tahun FROM tbl_alumni JOIN tbl_riwy_pekerjaan ON tbl_alumni.id_alumni = tbl_riwy_pekerjaan.id_alumni JOIN mst_pekerjaan ON tbl_riwy_pekerjaan.id_pekerjaa n =mst_pekerjaan.id_pekerjaan </pre>	<pre> SELECT tbl_riwy_keg_masyarakat.id_alumni, nama_lengkap, alamat, tbl_riwy_keg_masyarakat.bentuk_keg_masyara kat, tbl_riwy_keg_masyarakat.tahun FROM tbl_alumni JOIN tbl_riwy_keg_masyarakat ON tbl_alumni.id_alumni = tbl_riwy_keg_masyarakat.id_alumni </pre>

After doing the three stages above, we get the results of the ingres algorithm, as for the script command of the ingres algorithm as follows;

- Queries that have been optimized using the ingres algorithm.

Table 3. Optimized queries

Query command script
<pre> SELECT tbl_alumni.id_alumni AS id, tbl_alumni.nama_lengkap, tbl_alumni.alamat,nama_sekolah, tahun_lulus,nama_pekerjaan, tmp_keg_masyarakat.tahun, bentuk_keg_masyarakat, tmp_keg_masyarakat.tahun FROM tbl_alumni JOIN tmp_pendidikan ON tbl_alumni.id_alumni = tmp_pendidikan.id_alumni JOIN tmp_pekerjaan ON tbl_alumni.id_alumni = tmp_pekerjaan.id_alumni JOIN tmp_keg_masyarakat ON tbl_alumni.id_alumni = tmp_pekerjaan.id_alumni WHERE tmp_pendidikan.nama_sekolah= 'bahrul ulum' OR tmp_pekerjaan.nama_pekerjaan = 'guru' OR tmp_keg_masyarakat.bentuk_keg_masyarakat = 'mwc nu' </pre>

- Testing query commands
 At this stage, query command testing will be carried out using DBMS software. The software used is MySQL, phpMyAdmin and Navicat.
- Logging and implementation
 Recording and implementation is the process of documenting the results of observations when testing query commands. After that, the query that has been recorded results, will be compared with the processing speed with the optimized query.

RESULT AND DISCUSSION

Each description of the discussion will be tested and recorded the results of queries that have not been optimized with queries that have been optimized using the Ingres Algorithm so that we can find out the most effective execution time in processing a very large amount of data (Suhartati & Atma, 2017).

Accessing the database and testing query commands using phpMyAdmin following table relationships for the query trial process.



Figure 3. Relationships Table

Result

The following are the results of the execution of query commands that have not been optimized and the results of the execution of query commands that have been optimized using the Ingres algorithm ;

- Execution Results on query commands that have not been optimized

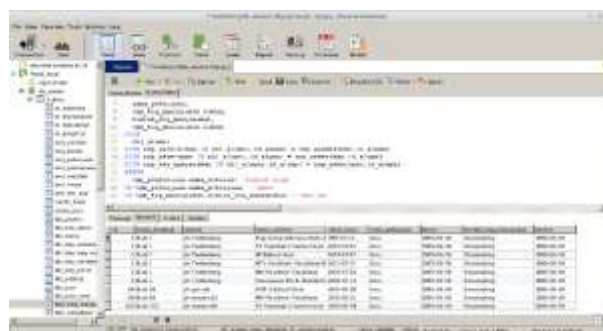


Figure 4. Unoptimized queries

- Optimized query command using the Ingres algorithm

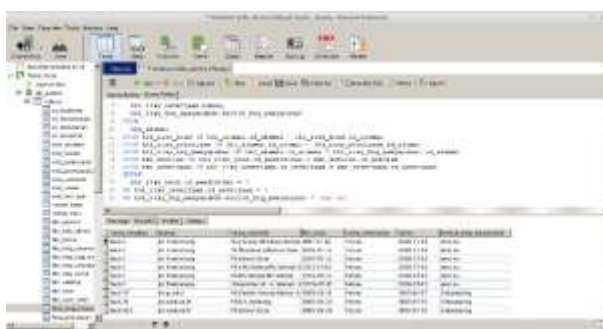


Figure 5. Optimized queries

Discussion

In this discussion, the results of the execution time from the study will be divided into 2 parts, namely the results of execution on queries that have not been optimized and execution results on queries that have been optimized using Algorimta Ingres. The following is a table of query execution results that have been optimized with queries that have not been optimized.

- Unoptimized query time results

Table 4. Query Time Results That Have Not Been Optimized

Uji coba Query	Objek Query			
	1	2	3	4
1	0.0636	0.0831	0.0750	2.467
2	0.0956	0.0930	0.0797	0.0791
3	0.0835	0.0861	0.0774	0.0723
4	0.0847	0.0794	0.0801	0.0825
5	0.0854	0.0773	0.0951	0.0759
Hasil	0.08256	0.08378	0.08146	0.55536
Rata-rata	0.20079			

- Optimized execution results using the ingres algorithm

Table 5. Query Time Results That Have Not Been Optimized

Uji Coba Query	Objek Query			
	1	2	3	4
1	0.0074	0.0203	0.0065	1.895
2	0.0062	0.0200	0.0046	0.0080
3	0.0059	0.0198	0.0045	0.0095
4	0.0054	0.0220	0.0057	0.0077
5	0.0056	0.0223	0.0051	0.0068
Hasil	0.0061	0.02088	0.00528	0.3854
Rata -rata	0.13226			

- Comparison of optimized query speed with queries that have not been optimized.

Table 6. Comparison Of Query Speeds

Q	Sebelum Dioptimalisasi	Sesudah dioptimalisasi	Kesimpulan	
1	0.08256	0.0061	73%	Lebih cepat
2	0.08378	0.02088	24%	Lebih cepat
3	0.08146	0.00528	64%	Lebih cepat
4	0,55536	0.3854	69%	Lebih cepat

CONCLUSION

Based on experiments conducted using queries for Query Optimization in Databases with the Ingres Algorithm. for use in the Bahrul Ulum Tambakberas Jombang Islamic boarding school. The result of a more effective and fast execution time is to use query commands that have been optimized using the ingres algorithm with an average processing of 0.13226 seconds, and the average processing of queries that have not been optimized 0.20079 seconds, in other words, the result is 65% faster than queries that have not been optimized.

REFERENCES

Conoras, M. E. B., & Kurnawan, A. D. (2017). Optimasi Query Untuk Pencarian Data Menggunakan Penguraian Kalimat dan Algoritme levenshtein Distance. *Seminar Nasional APTIKOM*, 77–81.

Dewi, M. M., & Rezeki, N. (2017). Analisis Perbandingan Optimasi Query Nasted Join dan Hash Join pada MySQL Server. *CSRID Journal*, 9(1), 31–41.

- Hastono, T. (2019). Desain Stored Procedures Untuk Optimalisasi Query pada Sistem Informasi Keuangan SD Muhammadiyah Sidoarum Berbasis Client-Server. *Seminar Nasional*, 35–39. <http://repository.upy.ac.id/id/eprint/2254>
- Prihatin, N., Sakethi, D., & Febriansyah, F. I. (2016). Optimasi Query Pada Sistem Informasi Pencatatan Aktifitas Perubahan Data Nilai Mahasiswa. *Jurnal Komputer*, 4(2), 67–75.
- Safitri, P. K., Winarno, W. W., & Pramono, E. (2018). Optimasi Query untuk Sistem Informasi Penjadwalan Mata Pelajaran Sekolah Menggunakan View (Studi Kasus : SMK VIP Purworejo). *Jurnal Teknologi Informasi*, XIII, 46–51.
- Siallagan, M., Sabariah, M. K., & Sontya, M. (2008). Optimasi Query Database Menggunakan Algoritma Genetik. *Seminar Nasional Aplikasi Teknologi Informasi*, 53–57. <http://jurnal.stmikbpn.ac.id/index.php/metik1/article/view/5>
- Suhartati, S., & Atma, Y. D. (2017). Optimasi Query Sederhana Guna Kecepatan Query pada Database Server. *METIK Jurnal*, 1(1). <http://jurnal.stmikbpn.ac.id/index.php/metik1/article/view/5>
- Watung, I. A., Sinsuw, A. A. E., Paturusi, S. D. E., & Najoran, X. B. N. (2014). Perancangan Sistem Informasi Data Alumni Fakultas Teknik Unsrat Berbasis Web. *E-Journal Teknik Elektro Dan Komputer*, 1–9.
- Wibowo, G. S., Susanto, A., & Utomo W.M., I. (2018). Optimasi Query Menggunakan Algoritma Ingres Pada Portal Lumbung Data Pendidikan Jawa Tengah. *Jurnal Maklumatika*, 5(1), 1–12. <http://ojs.unwaha.ac.id/sdsa/sdt.index.php>