

Analisis Perbandingan Cartesian Product, Cross Join, Inner Join dan Outer Join dalam Si Akad

Comparative Analysis of Cartesian Product, Cross Join, Inner Join and Outer Join in Si Akad

Fatkhurrochman¹, Ahmad Wildan L², Juwari³
^{1,2,3}Magister Teknik Informatika, Universitas Amikom Yogyakarta

Informasi Makalah

Dikirim, 7 Januari , 2018
Diterima, 30 April , 2018

Kata Kunci:

sistem informasi akademik
cartesian product
cross join
inner join
outer join

Keyword:

academic information system
cartesian product
cross join
inner join
outer join

INTISARI

Sistem akademik merupakan salah satu sistem yang dikembangkan dalam lembaga pendidikan. Manajemen sistem informasi akademik yang kurang baik akan mempengaruhi performa dari sistem informasi akademik. Pemilihan query yang kurang tepat akan berpengaruh pada kecepatan akses data dan sumber daya terkait. Dalam penelitian ini akan dilakukan analisis perbandingan query dalam sistem informasi akademik. Penelitian ini bertujuan untuk menentukan query yang tepat dalam menampilkan data nilai mahasiswa. Sehingga kecepatan data lebih optimal dan penggunaan sumber daya terkait lebih efektif. Query yang dianalisis adalah cartesian product, cross join, inner join dan outer join. dalam penelitian ini menggunakan metode eksperimen yaitu dengan try and error. Hasil dari penelitian ini adalah analisis perbandingan dengan cross join lebih efektif dibandingkan dengan cartesian product, inner join dan outer join. Dengan nilai waktu cross join 0.0039 detik, inner join 0,0045 detik, outer join 0,0077 detik dan cartesian product 0,0104 detik.

ABSTRACT

Academic system is one system developed in educational institutions. Improper management of academic information systems will affect the performance of academic information systems. Selection of a less precise query will affect the speed of data access and related resources. In this research will be done comparative analysis of query in academic information system. This study aims to determine the right query in displaying student value data. So the data rate is more optimized and the use of related resources is more effective. The queries analyzed were cartesian product, cross join, inner join and outer join. in this research using experimental method that is with try and error. The result of this research is comparison analysis with cross join more effective than cartesian product, inner join and outer join. With cross join time value 0.0039 seconds, inner join 0.0045 seconds, outer join 0.0077 seconds and cartesian product 0.0104 seconds.

Korespondensi Penulis:

Penulis Ke-1
Magister Teknik Informatika
Universitas Amikom Yogyakarta
Jalan Ring Road Utara Condong CaturDepok-Sleman
Email: fatkhurrochman69@gmail.com

1. PENDAHULUAN

Sistem informasi akademik (SIKAD) merupakan sebuah sistem yang dikembangkan sebagai salah satu fasilitas pendukung dalam lembaga pendidikan[1]. Salah satu fungsi SiAkad adalah menampilkan data nilai mahasiswa. Semakin banyak data yang akan ditampilkan maka kebutuhan sumber daya yang dibutuhkan juga semakin besar[2]. Manajemen sistem informasi akademik yang tidak dilelola dengan baik akan berpengaruh terhadap performa dari sistem informasi akademik tersebut[3].

Salah satu solusi untuk menangani masalah tersebut adalah dengan menerapkan query yang tepat. Penggunaan query yang krang tepat akan berpengaruh terhadap kecepatan akses data dan penggunaan sumber daya terkait. Dengan menggunakan metode cartesian product, cross join, inner join atau outer join akan meningkatkan performa sistem informasi. Pemilihan query yang tepat dapat meningkatkan kecepatan akses data, sehingga penggunaan sumber daya terkait dapat lebih efektif[4].

Dalam penelitian oleh Eko Darmanto, peneliti melakukan optimalisasi bahasa SQL berdasarkan Relational Algebra pada kasus rekapitulasi mahasiswa layak wisuda. Hasil dari penelitian ini menunjukkan bahwa pemuatan data ke dalam komputer lebih optimal jika menggunakan join dibandingkan dengan cartesian product. Ditinjau dari kecepatan komputasi dengan kondisi tertentu, cartesian product lebih cepat dibandingkan dengan join [5].

Pada penelitian sebelumnya oleh Mellany Mustika Dewi dan Novandi Rezeki, melakukan penelitian tentang analisis perbandingan optimalisasi Query Nested Join dan Hast Join pada MySQL Server. Peneliti membandingkan kecepatan pengolahan data pada query. Penelitian ini menghasilkan grafik hasil optimasi query dengan hasil terbaik adalah nested join[6].

Dalam penelitian oleh Suhartati dan Yeyen Dwi Atma, melakukan penelitian dengan optimalisasi Query sederhana dalam meningkatkan kecepatan query dalam database server. Dalam penelitian ini menghasilkan optional query yang telah dioptimalkan[7].

Dalam penelitian ini, peneliti melakukan analisis perbandingan optimasi query pada sistem informasi akademik menggunakan cartesian product, cross join, inner join dan outer join. Penelitian ini bertujuan untuk menentukan query yang tepat dalam sistem informasi akademik. Sehingga akan menghasilkan perbandingan kecepatan proses data antar metode yang digunakan dalam kasus yang sama.

2. METODOLOGI PENELITIAN

Optimalisasi Query

Optimalisasi query merupakan sebuah prosedur untuk meningkatkan strategi evaluasi proses akses data menjadi lebih efektif[4]. Optimasi query mencakup beberapa teknik seperti transformasi query ke dalam bentuk logika yang sama, memilih akses data yang tercepat dan mengoptimalkan penyimpanan data[5].

Adapun langkah-langkah yang dilakukan dalam penelitian ini meliputi :

1. Analisa objek, objek yang digunakan adalah sebuah perguruan tinggi dengan kriteria memiliki sejumlah mahasiswa, sejumlah mata kuliah, sejumlah kelas dan sejumlah dosen yang dijadikan sebagai data informasi terkait nilai mahasiswa. Objek memiliki aturan yang harus terpenuhi untuk menampilkan informasi nilai mahasiswa.
2. Penggalan data, data-data asli dianalisis dan dilakukan analisa awal tentang kecenderungan adanya relasi antar data. Kecenderungan relasi ini harus bisa dimodelkan menggunakan alat pemodelan data berupa ER-Diagram. Pada saat implementasi menggunakan data simulasi, karena data-data dari sebuah perguruan tinggi bersifat privasi.
3. Penemuan akar masalah, berdasarkan analisa data dan objek terkait, akar permasalahan harus dapat diidentifikasi dengan baik. Misalnya masalah data telah terintegrasi dengan baik tetapi belum ada langkah optimal dalam prosesnya. Sehingga akan berdampak pada lamanya proses query.
4. Menentukan solusi sementara, solusi sementara ini akan memberikan deskripsi terhadap langkah awal penggunaan data yang telah dilakukan analisa optimalisasi dengan metode tertentu. Maka akan menghemat biaya proses dan memperlebar pita konektivitas akses data.
5. Perancangan basis data, dalam tahap ini merupakan tahap paling penting dalam perancangan database relational. Media yang digunakan adalah ER-Diagram dan dipetakan dalam bentuk skema relasional basis data.
6. Implementasi basis data, untuk menganalisa kecepatan proses rancangan basis data maka hasil rancangan diimplementasikan dalam sebuah DBMS. DBMS yang digunakan adalah MySQL karena bersifat bebas pakai non komersial dengan lisensi publik.
7. Analisa menggunakan bahasa SQL, untuk mendefinisikan data, memanipulasi dan mengontrol akses, diperlukan DBMS yang memiliki bahasa SQL yang kuat. Penggunaan bahasa SQL menggunakan analisa model eksperimen dengan *trial and error*.

8. Analisa optimalisasi query, setiap hasil dari pemrosesan query dianalisa lama prosesnya. Semakin lama prosesnya diasumsikan bahwa akan memakan banyak biaya sumber daya komputasi yang ada. Sumber daya komputasi yang dipakai adalah kapasitas memori penyimpanan sementara RAM, kecepatan prosesor serta lebar pita transmisi data dan medianya.

Proses Pengujian

Proses pengujian optimalisasi query dilakukan dengan menggunakan laptop dengan spesifikasi sebagai berikut :

Laptop : HP
 Sistem Operasi : Windows 10 Pro
 Prosesor : AMD A4-3330MX APU
 RAM : 2,00 GB
 Tipe Sistem : 64 bit-OS

Aplikasi yang digunakan adalah SQLyog sebagai MySQL user interface dan XAMPP. Setelah XAMPP dijalankan, maka query dengan jumlah record yang telah dimasukkan dapat diproses dan dianalisa terkait waktu yang diperlukan dalam setiap eksekusi.

3. HASIL DAN PEMBAHASAN

Perancangan Basis data

Skema relasional basis data sistem informasi akademik adalah sebagai berikut :

Mahasiswa = (NIM, nama, jekel, alamat, tahun_akad).

Nilai = (NIM, id_kelas, nilai).

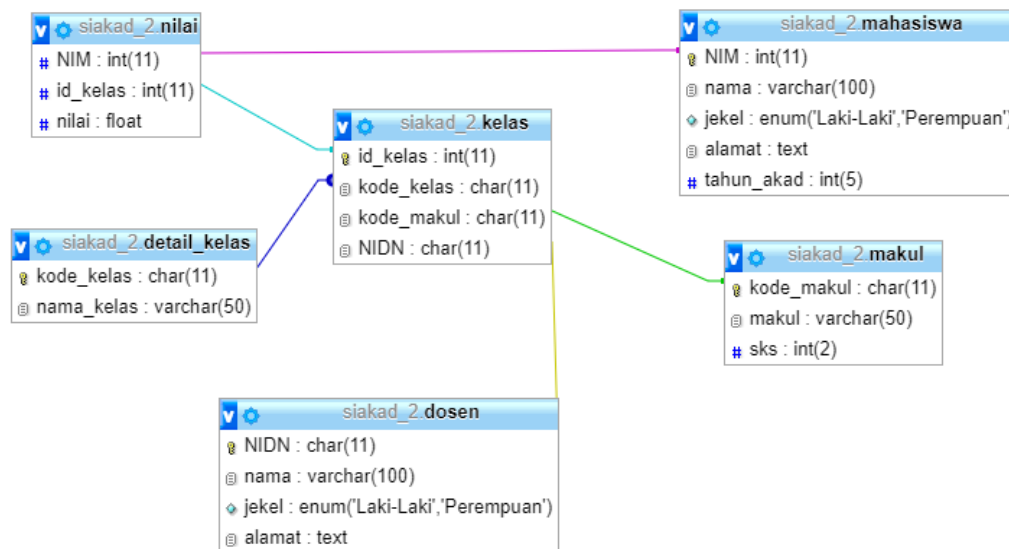
Kelas = (id_kelas, kode_kelas, kode_makul, NIDN).

Detail_kelas = (kode_kelas, nama_kelas).

Makul = (kode_makul, makul, sks).

Dosen = (NIDN, nama, jekel, alamat).

Berdasarkan sistem relasi yang terbantu, maka disusun sebuah perancangan database sebagai berikut :



Gambar 1. Perancangan database

Berdasarkan perancangan database pada Gambar 1, diasumsikan bahwa tabel telah berada dalam sebuah DBMS dengan data simulasi mendekati kenyataan. Simulasi dari data perguruan tinggi dengan jumlah entitas data yaitu pada Tabel 1 berikut :

Tabel 1. Simulasi jumlah data

No	Nama tabel	Jumlah kolom	Jumlah data
1	Mahasiswa	5	150
2	Nilai	3	1200
3	Kelas	4	96
4	Detail_kelas	2	12
5	Makul	3	20
6	Dosen	4	14

Analisis Optimasi Perbandingan

Berdasarkan Tabel 1, simulasi jumlah data dapat diperoleh bahwa setiap mahasiswa menempuh 8 mata kuliah dalam satu kelas dengan jumlah nilai 1200 dari 150 mahasiswa dalam 96 kelas perkuliahan. Dalam satu query dilakukan 12 pengujian dengan limit data 2n dan perlimit dilakukan eksekusi sebanyak 5 kali. Hasil pengujian dianalisis dan didapatkan rata-rata dari setiap query tersebut.

Dalam pengujian tersebut, database yang telah disiapkan dilakukan pengujian dengan 5 buah query, yaitu query 1 relasi, query 2 relasi, query 3 relasi, query 4 relasi dan query 5 relasi. Dimana setiap query relasi dilakukan pengujian dengan 4 teknik yang berbeda, yaitu cartesian product, cross join, inner join dan outer join.

Berikut 5 buah query yang akan dieksekusi :

1. Query menggunakan cartesian product

Query satu relasi

```
select mahasiswa.NIM, mahasiswa.nama, nilai.nilai from mahasiswa, nilai where
mahasiswa.NIM=nilai.NIM limit n;
```

Query dua relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, nilai.nilai from mahasiswa, nilai, kelas where
mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas limit n;
```

Query tiga relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas,detail_kelas.nama_kelas, makul.makul,
makul.sks, nilai.nilai from mahasiswa, nilai, kelas, makul, detail_kelas where
mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas and kelas.kode_makul =
makul.kode_makul and kelas.kode_kelas = detail_kelas.kode_kelas limit n;
```

Query empat relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas,detail_kelas.nama_kelas, makul.makul,
makul.sks, nilai.nilai from mahasiswa, nilai, kelas, makul, detail_kelas where
mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas and kelas.kode_makul =
makul.kode_makul and kelas.kode_kelas = detail_kelas.kode_kelas limit n;
```

Query lima relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, detail_kelas.nama_kelas, makul.makul,
makul.sks, dosen.nama, nilai.nilai from mahasiswa, nilai, kelas, makul, detail_kelas, dosen where
mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas and kelas.kode_makul = makul.kode_makul
and kelas.kode_kelas = detail_kelas.kode_kelas kelas.NIDN = dosen.NIDN limit n;
```

Berdasarkan hasil eksekusi query diatas, dimana n pada limit adalah jumlah data maka didapatkan data kecepatan proses akses database pada Tabel 2 berikut :

Tabel 2. Kecepatan akses data Cartesian Product

Jumlah data	Jumlah relasi				
	1	2	3	4	5
1	0,0013	0,0014	0,0016	0,0015	0,0016
2	0,0013	0,0015	0,0016	0,0015	0,0017
4	0,0013	0,0016	0,0018	0,0014	0,0015
8	0,001				
	4	0,0016	0,0020	0,0015	0,0015

16	0,0014	0,0036	0,0019	0,0019	0,0019
32	0,001				
	4	0,0032	0,0023	0,0020	0,0025
64	0,0017	0,0031	0,0027	0,0026	0,0028
128	0,0053	0,0028	0,0031	0,0076	0,0086
512	0,0065	0,0044	0,0057	0,0067	0,0086
1024	0,0106	0,0181	0,0088	0,0277	0,0328
1200	0,0119	0,0251	0,0382	0,0444	0,0327

2. Query menggunakan cross join

Query satu relasi

select mahasiswa.NIM, mahasiswa.nama, nilai.nilai from mahasiswa cross join nilai where mahasiswa.NIM=nilai.NIM limit n ;

Query dua relasi

select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, nilai.nilai from mahasiswa cross join nilai, kelas where mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas limit n ;

Query tiga relasi

select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, makul.makul, makul.sks, nilai.nilai from mahasiswa cross join nilai, kelas, makul where mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas and kelas.kode_makul = makul.kode_makul limit n ;

Query empat relasi

select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, detail_kelas.nama_kelas, makul.makul, makul.sks, nilai.nilai from mahasiswa cross join nilai, kelas, makul, detail_kelas where mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas and kelas.kode_makul = makul.kode_makul and kelas.kode_kelas = detail_kelas.kode_kelas limit n ;

Query lima relasi

Select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, detail_kelas.nama_kelas, makul.makul, makul.sks, dosen.nama, nilai.nilai from mahasiswa cross join nilai, kelas, makul, detail_kelas, dosen where mahasiswa.NIM=nilai.NIM and kelas.id_kelas = nilai.id_kelas and kelas.kode_makul = makul.kode_makul and kelas.kode_kelas = detail_kelas.kode_kelas and kelas.NIDN = dosen.NIDN limit n ;

Berdasarkan hasil eksekusi query diatas, dimana n pada limit adalah jumlah data maka didapatkan data kecepatan proses akses database pada Tabel 3 berikut :

Tabel 3 Kecepatan akses data Cross Join

Jumlah data	Waktu				
	1	2	3	4	5
1	0,0007	0,0009	0,0012	0,0015	0,0016
2	0,0011	0,0011	0,0013	0,0016	0,0018
4	0,0012	0,0010	0,0013	0,0017	0,0018
8	0,0010	0,0012	0,0013	0,0016	0,0019
16	0,0012	0,0013	0,0014	0,0017	0,0020
32	0,0012	0,0016	0,0016	0,0019	0,0022
64	0,0015	0,0017	0,0019	0,0022	0,0024
128	0,0019	0,0023	0,0026	0,0034	0,0037
512	0,0025	0,0028	0,0035	0,0050	0,0049
1024	0,0036	0,0042	0,0056	0,0063	0,0074
1200	0,0061	0,0072	0,0122	0,0109	0,0113

3. Query menggunakan inner join

Query satu relasi

select mahasiswa.NIM, mahasiswa.nama, nilai.nilai from mahasiswa inner join nilai on mahasiswa.NIM=nilai.NIM limit n ;

Query dua relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, nilai.nilai from mahasiswa inner join nilai on
mahasiswa.NIM=nilai.NIM inner join kelas on kelas.id_kelas = nilai.id_kelas limit n;
```

Query tiga relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, makul.makul, makul.sks, nilai.nilai from
mahasiswa inner join nilai on mahasiswa.NIM=nilai.NIM inner join kelas on kelas.id_kelas = nilai.id_kelas
inner join makul on kelas.kode_makul = makul.kode_makul limit n;
```

Query empat relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas,detail_kelas.nama_kelas, makul.makul,
makul.sks, nilai.nilai from mahasiswa inner join nilai on mahasiswa.NIM=nilai.NIM inner join
kelas on kelas.id_kelas = nilai.id_kelas inner join makul on kelas.kode_makul = makul.kode_makul
inner join detail_kelas on kelas.kode_kelas = detail_kelas.kode_kelas limit n;
```

Query lima relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, detail_kelas.nama_kelas, makul.makul,
makul.sks, dosen.nama, nilai.nilai from mahasiswa inner join nilai on mahasiswa.NIM=nilai.NIM inner
join kelas on kelas.id_kelas = nilai.id_kelas inner join makul on kelas.kode_makul = makul.kode_makul
inner join detail_kelas on kelas.kode_kelas = detail_kelas.kode_kelas inner join dosen on
kelas.NIDN = dosen.NIDN limit n;
```

Berdasarkan hasil eksekusi query diatas, dimana n pada limit adalah jumlah data maka didapatkan data kecepatan proses akses database pada Tabel 4 berikut :

Jumlah data	Waktu				
	1	2	3	4	5
1	0,0013	0,0013	0,0014	0,0016	0,0017
2	0,0014	0,0013	0,0015	0,0016	0,0018
4	0,0015	0,0014	0,0015	0,0019	0,0021
8	0,0016	0,0015	0,0016	0,0016	0,0023
16	0,0014	0,0017	0,0017	0,0018	0,0023
32	0,0016	0,0017	0,0019	0,0022	0,0021
64	0,0017	0,0020	0,0024	0,0036	0,0048
128	0,0023	0,0026	0,0028	0,0029	0,0030
512	0,0029	0,0033	0,0045	0,0043	0,0041
1024	0,0044	0,0050	0,0069	0,0068	0,0079
1200	0,0061	0,0089	0,0107	0,0199	0,0188

4. Query menggunakan outer join

Query satu relasi

```
select mahasiswa.NIM, mahasiswa.nama, nilai.nilai from mahasiswa left join nilai on
mahasiswa.NIM=nilai.NIM limit n;
```

Query dua relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, nilai.nilai from mahasiswa left join nilai on
mahasiswa.NIM=nilai.NIM left join kelas on kelas.id_kelas = nilai.id_kelas limit n;
```

Query tiga relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, makul.makul, makul.sks, nilai.nilai from
mahasiswa left join nilai on mahasiswa.NIM=nilai.NIM left join kelas on kelas.id_kelas = nilai.id_kelas
left join makul on kelas.kode_makul = makul.kode_makul limit n;
```

Query empat relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas,detail_kelas.nama_kelas, makul.makul,
makul.sks, nilai.nilai from mahasiswa left join nilai on mahasiswa.NIM=nilai.NIM left join kelas on
kelas.id_kelas = nilai.id_kelas left join makul on kelas.kode_makul = makul.kode_makul left join
detail_kelas on kelas.kode_kelas = detail_kelas.kode_kelas limit n;
```

Query lima relasi

```
select mahasiswa.NIM, mahasiswa.nama, kelas.kode_kelas, detail_kelas.nama_kelas, makul.makul,
makul.sks, dosen.nama, nilai.nilai from mahasiswa left join nilai on mahasiswa.NIM=nilai.NIM left
join kelas on kelas.id_kelas = nilai.id_kelas left join makul on kelas.kode_makul = makul.kode_makul left
join detail_kelas on kelas.kode_kelas = detail_kelas.kode_kelas left join dosen on kelas.NIDN = dosen.NIDN
limit n;
```

Berdasarkan hasil eksekusi query diatas, dimana n pada limit adalah jumlah data maka didapatkan data kecepatan proses akses database pada Tabel 5 berikut :

Tabel 5. Kecepatan akses data outer Join

Jumlah data	Waktu				
	1	2	3	4	5
1	0,0011	0,0013	0,0012	0,0011	0,0016
2	0,0012	0,0014	0,0009	0,0012	0,0012
4	0,0013	0,0014	0,0009	0,0009	0,0013
8	0,0014	0,0015	0,0015	0,0010	0,0015
16	0,0015	0,0016	0,0016	0,0013	0,0020
32	0,0018	0,0017	0,0017	0,0013	0,0021
64	0,0025	0,0019	0,0023	0,0022	0,0038
128	0,0019	0,0025	0,0025	0,0028	0,0042
512	0,0039	0,0034	0,0039	0,0038	0,0051
1024	0,0042	0,0065	0,0057	0,0067	0,0084
1200	0,0063	0,0083	0,0110	0,0118	0,0140

Dari analisis metode yang telah dilakukan pengujian, maka diperoleh data perbandingan rata-rata kecepatan akses pada Tabel 6 berikut :

Tabel 6. Rata-rata perbandingan kecepatan akses

Jumlah data	Cartesian product	Cross join	Inner join	Outer join
1	0,0015	0,0012	0,0015	0,0013
2	0,0015	0,0014	0,0015	0,0009
4	0,0015	0,0014	0,0017	0,0012
8	0,0016	0,0014	0,0017	0,0014
16	0,0021	0,0015	0,0018	0,0016
32	0,0023	0,0017	0,0019	0,0017
64	0,0026	0,0019	0,0029	0,0025
128	0,0055	0,0028	0,0027	0,0028
512	0,0064	0,0037	0,0038	0,0040
1024	0,0196	0,0054	0,0062	0,0063
1200	0,0305	0,0095	0,0129	0,0103
Jumlah	0,1249	0,0473	0,0543	0,0923
Rata-rata	0,0104	0,0039	0,0045	0,0077

Dari Tabel 6 diatas dalam database sistem informasi akademik, antar metode memiliki waktu yang berbeda dalam kecepatan proses data. Maka dapat dilihat bahwa dalam kasus ini query menggunakan cross join memiliki kecepatan proses paling optimal. Hal ini dapat dilihat pada waktu yang dibutuhkan dalam proses data yaitu pada cross join 0,0039 detik sedang pada cartesian product 0,0104 detik, inner join 0,0045 dan outer join 0,0077 detik. Sehingga dalam kasus ini solusi query paling optimal adalah menggunakan metode cross join.

4. KESIMPULAN DAN SARAN

Simpulan

Dari pengujian 4 query relasi, dapat disimpulkan bahwa query yang memiliki kecepatan proses data dengan waktu paling minimal adalah cross join dengan nilai rata rata 0,0039 detik. Sedangkan inner join 0,0045 detik, outer join 0,0077 detik dan cartesian product 0,0104 detik.

Saran

Untuk penelitian berikutnya disarankan untuk dibangun sebuah aplikasi sehingga dapat lebih memudahkan dalam mengetahui kecepatan eksekusi optimasi query. Membandingkan optimasi query pada kasus yang lebih kompleks dan SQL yang lebih variatif.

DAFTAR PUSTAKA

- [1] U. PGRI, A. D. I. Buana, R. Universitas, P. Adi, dan B. Surabaya, "Resistensi Pengguna Terhadap Sistem Informasi Akademik Universitas PGRI Adi Buana Surabaya," *wahana*, vol. 67, no. 1127, hal. 39–46, 2016.
- [2] C. Ordonez dan J. Garcia-Garcia, "Managing Big Data Analytics Workflows with a Database System," *Proc. - 2016 16th IEEE/ACM Int. Symp. Clust. Cloud, Grid Comput. CCGrid 2016*, hal. 649–655, 2016.
- [3] J. L. Harrington, "Relational Database Design and Implementation," *elsevier*, vol. 4th, hal. 935, 2016.
- [4] E. Ramadyani, "Analisa Optimasi Query Database Dalam Pembelajaran Berbasis Web Dengan Metode M2S Crossover Dan Chunk Crossover," *Times*, vol. 5, no. 2, hal. 43–48, 2016.
- [5] E. Darmanto, F. Teknik, P. Studi, S. Informasi, dan U. M. Kudus, "Analisa Optimalisasi Bahasa Sql Berdasarkan Relational," *Simetris*, vol. 6, no. 2, hal. 405–414, 2015.
- [6] C. Analysis, "Analisis Perbandingan Optimasi Query Nested Join dan Hash Join pada MySQL Server," *CSRID J.*, vol. 9, no. 1, hal. 31–41, 2017.
- [7] yeyen dwi atma Suharti, "Optimalisasi Query Sederhana Guna Kecepatan Query pada Database Server," *citec amikom*, 2015.