

Perbandingan Query Response Time pada Model Query View dan Cross Product

Comparison of Query Response Time in the Query View and Cross Product

Noviyanti P¹, Agatha Deolika², Siti Hartinah³, Celine Aloyshima Haris⁴, Tutik Maryana⁵,
Nindy Devita Sari⁶

^{1,2,3,4,5,6}Magister Teknik Informatika Universitas AMIKOM Yogyakarta

E-mail: *¹novi.jiuu94@gmail.com, ²agathadeolika@gmail.com, ³sitihartina361@gmail.com
⁴caloyshima@gmail.com, ⁵tutik.maryana11@gmail.com, ⁶nindydevitasari98@gmail.com

Abstrak

Perkembangan teknologi saat ini sangat dipengaruhi oleh sekumpulan himpunan data yang tidak terstruktur atau biasa disebut dengan big data. Big data menjadi inovasi yang sangat bermanfaat untuk dapat memberikan suatu perkembangan pada berbagai bidang, salah satunya yaitu bidang pendidikan. Data yang terlampau banyak tentunya memunculkan suatu masalah bagi pengelola database sehingga dibutuhkan suatu solusi. Salah satu solusi yang dilakukan pada penelitian ini adalah membandingkan response time model penulisan query. Bidang pendidikan seperti universitas tentunya memiliki berbagai macam data yang berhubungan dengan mahasiswa. Pengujian pada penelitian ini adalah database pengelolaan nilai mahasiswa yang terdiri dari data tabel dosen, data tabel mahasiswa, data tabel mata kuliah, dan data tabel nilai. Tujuan dari penelitian ini adalah untuk membandingkan respon waktu pada 2 model penulisan query yaitu query view dan query Cross Product dalam tabel non partisi agar dapat menghasilkan rekomendasi model penulisan query dengan respon waktu yang lebih cepat. Setelah melakukan 3 kali pengujian dengan sampel data 100, 500, 1000 yang dilakukan dengan hardware yang sama, penulis dapat menyimpulkan bahwa query view lebih cepat dengan selisih 0,0014 sampai 0,0040 detik.

Kata Kunci— Database, Response Time, Select View, Cross Product.

Abstract

Current technological developments are very important by a set of unstructured data sets or commonly called large data. Big data becomes an innovation that is very useful to provide development in various fields, one of which is education. Too much data helps to create a problem for database managers so that a solution is needed. One solution that was done in this study was to compare the response times of the comparable models. Education such as universities has various kinds of data relating to students. The test in this study is the management of student value data consisting of lecturer table data, student table data, course table data, and value table data. The purpose of this study is to compare the time response in 2 models to receive requests, namely the display of Cross Product requests and requests in non-partition tables in order to produce evaluation models for request submission with faster response times. After doing 3 tests with a sample of 100, 500, 1000 data that is done with the same hardware, the author can consider requesting a faster display with a difference of 0.0014 to 0.0040 seconds.

Keywords — Database, Response Time, Select View, Cross Product.

1. PENDAHULUAN

1.1 LATAR BELAKANG

Seiring berjalannya waktu, teknologi berkembang dengan menghadirkan berbagai inovasi untuk dapat semakin mempermudah pekerjaan manusia. Perkembangan teknologi

tentunya dipengaruhi oleh berbagai macam data yang kemudian diolah dan diperoleh suatu informasi sehingga berkembanglah suatu inovasi. Untuk dapat mengolah data tersebut tidak hanya dengan mengumpulkan data secara manual, tetapi diperlukan beberapa cara untuk dapat mengolah kemudian menampilkan data yang diperlukan saja.

Perkembangan teknologi saat ini juga sangat dipengaruhi oleh sekumpulan himpunan data yang tidak terstruktur atau biasa disebut dengan Big Data. Big Data merupakan sekumpulan data yang banyak dan tak terhingga dengan berbagai macam jenis dan sumber data, sehingga untuk mengelola data tersebut diperlukan suatu metode dan perangkat bantu yang cara kerjanya sesuai [1]. Big data menjadi inovasi yang sangat bermanfaat untuk dapat memberikan suatu perkembangan pada berbagai bidang dan komunitas, salah satunya bidang pendidikan. Bidang pendidikan seperti perguruan tinggi tentunya memiliki berbagai macam data yang berhubungan dengan mahasiswa. Data yang dimiliki tiap perguruan tinggi tentunya sangat banyak. Untuk dapat menampilkan data dengan cepat diperlukan cara dalam menampilkan data tersebut dengan menggunakan model query yang lebih cepat dan efisien agar dapat mempermudah dan mempercepat pekerjaan manusia dalam hal menampilkan data yang diperlukan tersebut.

Untuk mengolah suatu data diperlukan sebuah tool. Tool yang digunakan pada penelitian ini adalah MySQL. MySQL merupakan sebuah *software* yang dilengkapi dengan sebuah *database server* yang dapat digunakan secara bebas dan legal untuk keperluan pribadi ataupun untuk usaha dan dapat diperoleh secara gratis tanpa harus membayar lisensinya [2].

Pada penelitian ini digunakan suatu data yang terdapat pada bidang pendidikan. Sampel data yang digunakan adalah data penilaian dalam perkuliahan yang terdiri dari data tabel dosen, data tabel mahasiswa, data tabel mata kuliah, dan data tabel nilai. Model penulisan *query* didalam suatu basis data terdiri dari *query view* dan *cross product*. Dengan menggunakan model penulisan *query* tersebut maka akan diuji model penulisan *query* mana yang lebih cepat dalam proses menampilkan suatu data.

Pada penelitian ini membandingkan kedua model penulisan query tersebut dengan menggunakan sebanyak 100 data, 500 data, dan 1.000 data. Tujuan dari penelitian ini adalah untuk membandingkan *response time* menggunakan model penulisan *query* dengan *query view* dan *cross product* dalam tabel non partisi agar dapat menghasilkan rekomendasi model penulisan *query* yang lebih cepat.

Berdasarkan latar belakang yang telah dijelaskan tersebut, maka penulis mengangkat sebuah judul yang akan dilakukan pada penelitian ini, yaitu “Perbandingan *Query Response Time* pada *Query View* dan *Cross Product*”.

1.2 LANDASAN TEORI

1.2.1 Database

Basis data adalah kumpulan catatan elektronik tentang entitas. Entitas mungkin buku, artikel jurnal, perangko, kaset video, peminjam, asosiasi, dan lain-lain. Pengumpulan informasi tentang satu jenis entitas adalah file. Dalam program manajemen basis data relasional, file ini disebut tabel [3].

Ketika mempertimbangkan pembuatan database baru, atau mungkin dalam membenarkan kelanjutan dari yang sudah ada, hal-hal yang berkaitan dengan konten akan menjadi perhatian utama. Beberapa kriteria penting yang mengatur pembenaran untuk keberadaan database meliputi:

- a. Domain cakupan
- b. Aksesibilitas konten
- c. Dapat diprediksi cakupan
- d. Kesenambungan cakupan
- e. Mata uang cakupan

f. Massa kritis

Ini semua terkait dalam beberapa cara dengan isi database.

Basis data (database) juga merupakan sekumpulan data yang saling berelasi dan dapat digunakan untuk memperoleh suatu informasi. Data yang ada kemudian diolah dimana data tersebut terdiri dari fakta mengenai obyek, orang, dan lain-lain. Data pada umumnya dinyatakan dengan suatu *value* dimana *value* nya berupa angka, deretan karakter, atau simbol. Tujuan database adalah untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali [4].

1.2.2 MySQL

MySQL, database SQL open source paling populer, dikembangkan dan disediakan oleh MySQL AB. MySQL AB adalah perusahaan komersial yang membangun bisnisnya menyediakan layanan di sekitar database MySQL. Kami pernah memulai dengan niat menggunakan mSQL untuk terhubung ke tabel kami menggunakan rutinitas tingkat rendah cepat (ISAM) kami sendiri. Namun, setelah beberapa pengujian kami sampai pada kesimpulan bahwa mSQL tidak cukup cepat atau cukup fleksibel untuk kebutuhan kita. Ini menghasilkan antarmuka SQL baru ke basis data kami tetapi dengan antarmuka API yang hampir sama dengan mSQL. API ini dipilih untuk memudahkan porting kode pihak ketiga. Derivasi dari nama MySQL tidak sepenuhnya jelas. Direktori basis kami dan sejumlah besar perpustakaan dan alat kami telah memiliki awalan "saya" selama lebih dari 10 tahun. Namun, putri Monty (beberapa tahun lebih muda) juga bernama My. Yang mana dari keduanya memberi namanya ke MySQL masih menjadi misteri, bahkan bagi kita [5].

1.2.3 XAMPP

XAMPP adalah distribusi Apache yang dikembangkan oleh Kai "Oswald" Seidler dan Kay Vogelgesang yang berisi Apache, MySQL, dan PHP. Faktanya, itulah yang dimaksud AMP di XAMPP untuk Apache, MySQL, dan PHP. P terakhir adalah untuk Perl, bahasa skrip lain. Huruf pertama dalam akronim, X, berarti *cross-platform*, dan menyiratkan bahwa XAMPP tersedia untuk sistem operasi Windows, Mac, dan Linux, XAMPP sangat mudah untuk diinstal, dan secara otomatis akan mengkonfigurasi *server* web Apache bersama dengan MySQL, PHP, dan Perl [6].

1.2.4 Tabel Non Partisi

Didalam database terdapat dua jenis tabel yang dapat digunakan, yaitu tabel partisi dan tabel tidak partisi atau non-partisi. Filosofi partisi adalah memecah tabel ke dalam beberapa *segment* (partisi atau subpartisi), di mana tabel konvensional hanya mempunyai satu *segment*. Misalkan kita punya tabel PENJUALAN dengan 8 juta *records*, jika ingin *query* data untuk kuartal pertama tahun ini. Pada tabel konvensional (non partition), *query* akan men-scan keseluruhan 8 juta *records* data tersebut karena berada dalam 1 *segment*. Apabila tabel itu dipartisi (by range untuk kolom tanggal penjualan) maka *query* akan men-scan khusus *segment* di mana data itu berada; tidak semua 8 juta *records* data itu di-scan, sehingga proses *query* lebih cepat [7].

1.2.5 Query

Dari tiga *query* khusus SQL, *query* definisi data adalah yang paling tidak berguna terhadap tabel lokal. Segala sesuatu yang dapat dilakukan dengannya juga dapat dilakukan dengan menggunakan alat desain di *Access*. Namun, *query* definisi data adalah cara yang efisien untuk membuat atau mengubah objek database. Dengan kueri definisi data, pernyataan SQL ini dapat digunakan [8]:

a) CREATE TABLE

- b) ALTER TABEL
- c) DROP TABEL
- d) CREATE INDEKS
- e) DROP INDEKS

1.2.6 Response Time

Hal yang menjadi penilaian dalam performansi suatu basis data adalah *response time* [9]. *Response Time* adalah waktu tanggap yg diberikan oleh antar muka/*interface* ketika *user* mengirim permintaan ke komputer. Secara umum, pengguna menginginkan bahwa program aplikasinya dapat memberikan waktu tanggap yang sependek – pendeknya. Tetapi waktu tanggap yang baik memang tidak dapat ditentukan, karena ada beberapa aspek yang mempengaruhi, antara lain yakni ragam interaksi yang diinginkan dan kefasihan pengguna dalam menjalankan program aplikasi tersebut.

1.2.7 Optimalisasi Query

Optimalisasi *query* merupakan sebuah prosedur untuk meningkatkan strategi evaluasi proses akses data menjadi lebih efektif. Optimasi *query* mencakup beberapa teknik seperti transformasi *query* ke dalam bentuk logika yang sama, memilih akses data yang tercepat dan mengoptimalkan penyimpanan data [10].

1.3 TINJAUAN PUSTAKA

Penelitian yang dilakukan oleh Moh Silhan dan Isa Anshori tentang Perbandingan *Subset Query* pada *Multiple Relasi* menggunakan Tabel Terpartisi dan Tabel Tidak terpartisi dengan Metode Cost-Based. Hasil perbedaan *query* yang di akses pada tabel non-partisi dan tabel yang telah di partisi waktu yang paling cepat adalah dari tabel yang dipartisi terbukti bahwa tabel yang di partisi waktu yang di perlukan 0.0736 sedangkan waktu yang diperlukan untuk akses data pada tabel non-partisi 0.1725. Untuk akses data yang di akses menggunakan *card LAN* lebih cepat dibandingkan akses dengan menggunakan jaringan *wireless* karena akses data dengan menggunakan *wireless* dipengaruhi jarak dari akses data [11].

Penelitian yang dilakukan oleh Suhartati dan Yeyen Dwi Atma membahas tentang “Optimasi *Query* Sederhana Guna Kecepatan *Query* pada Database *Server*. Dalam penelitian tersebut membahas bahwa untuk memperoleh suatu kinerja aplikasi dibutuhkan teknik optimasi *query* dengan mempraktikkan aturan penulisan *query*, sehingga diperoleh trik *query* sederhana yang bisa diterapkan untuk mendapatkan keuntungan kinerja secara optimal [12].

2. METODE PENELITIAN

2.1. Alur Penelitian

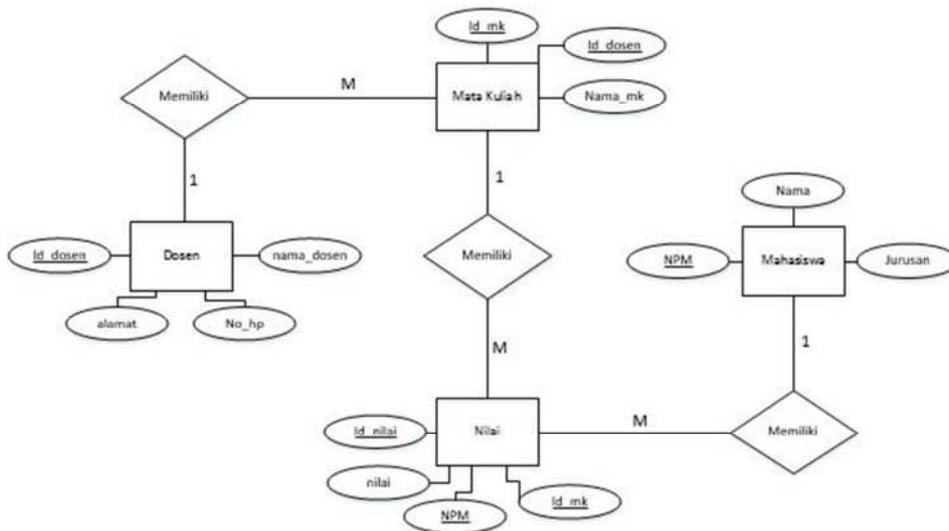
Alur metodologi penelitian ini, yaitu dimulai dengan studi pustaka, kemudian pengumpulan data, analisis kebutuhan sistem seperti hardware dan software yang digunakan untuk penelitian, pembuatan database termasuk juga pembuatan tabel, setelah itu melakukan pengujian dengan membandingkan model penulisan *query* view dan cross product, dan selesai. Adapun alur metodologi penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

2.1.1. ERD

ERD (Entity Relationship Diagram) merupakan suatu gambaran awal dari sebuah rancangan basis data yang akan dibuat. Pada Gambar 2 merupakan rancangan ERD yang akan diuji.

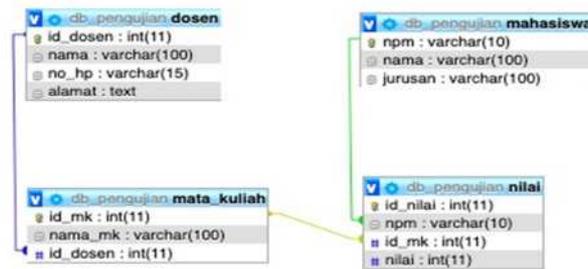


Gambar 2. Entity Relationship Diagram

Pada Gambar 2 terdapat entitas dosen yang berelasi one to many dengan entitas mata kuliah, entitas mata kuliah berelasi one to many dengan entitas nilai, dan entitas nilai berelasi many to one dengan entitas mahasiswa.

2.1.2. Relasi Tabel

Relasi tabel dalam basis data yang akan diuji seperti pada Gambar 3.



Gambar 3. Relasi Tabel

Pada Gambar 3 di tabel dosen terdapat id_dosen sebagai primary key, di tabel mata kuliah terdapat id_mk sebagai primary key dan id_dosen sebagai foreign key, di tabel nilai terdapat id_nilai sebagai primary key, npm dan id_mk sebagai foreign key.

2.2.Skala Uji Data

Skala uji data yang akan digunakan dalam penelitian ini seperti pada Tabel 1.

Tabel 1. Skala Uji Data

	Nama	Jumlah Data	Group	Keterangan
1	Data_100	100	Kecil	Limit 100
2	Data_500	500	Sedang	Limit 500
3	Data_1000	1000	Besar	Limit 1000

3. HASIL DAN PEMBAHASAN

3.1.Tabel Non Partisi

Dalam membuat tabel non partisi dalam penelitian ini salah satu contohnya, yaitu membuat tabel nilai dengan penulisan *query* sebagai berikut :

```
CREATE TABLE `nilai` (
  `id_nilai` int(11) NOT NULL,
  `npm` varchar(10) NOT NULL,
  `id_mk` int(11) NOT NULL,
  `nilai` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `nilai`
  ADD CONSTRAINT `nilai_ibfk_1` FOREIGN KEY (`npm`) REFERENCES `mahasiswa` (`npm`),
  ADD CONSTRAINT `nilai_ibfk_2` FOREIGN KEY (`id_mk`) REFERENCES `mata_kuliah`
  (`id_mk`);
COMMIT;
```

Dalam membuat tabel nilai terdapat atribut *id_nilai* sebagai *primary key*, atribut *npm* sebagai *foreign key* dari tabel mahasiswa, atribut *id_mk* sebagai *foreign key* dari tabel mata kuliah, dan atribut nilai.

3.2. Query View

Penulisan *query* berikut menunjukkan pembuatan tabel *view_nilai*.
 CREATE VIEW view_nilai as SELECT mh.nama, mh.jurusan,mk.nama_mk,d.nama_dosen,n.nilai
 from nilai n JOIN mata_kuliah mk on n.id_mk=mk.id_mk
 JOIN dosen d on mk.id_dosen=d.id_dosen
 JOIN mahasiswa mh on n.npm=mh.npm
 Setelah dilakukan pembuatan tabel selanjutnya menampilkan tabel *view_nilai* dengan limit 100 data menggunakan penulisan *query* berikut :
 select * from view_nilai limit 100

3.3. Cross Product

Dalam proses menampilkan tabel *view_nilai* dengan limit 100 data menggunakan model penulisan *query cross product* dapat dilakukan dengan secara langsung dengan menggunakan penulisan *query* seperti berikut :
 SELECT mh.nama, mh.jurusan, mk.nama_mk, d.nama_dosen, n.nilai from nilai n JOIN
 mata_kuliah mk on n.id_mk=mk.id_mk
 JOIN dosen d on mk.id_dosen=d.id_dosen
 JOIN mahasiswa mh on n.npm=mh.npm limit 100

3.4. Hasil Analisis

Pengujian dalam penelitian ini dilakukan pada server local yang menggunakan perangkat dengan spesifikasi :

- a) Sistem operasi Windows 8.1 64 Bit
- b) Processor Intel Core i7-4702MQ
- c) Ram 16GB
- d) Hardisk 1TB

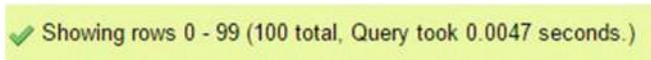
Dalam penelitian ini membandingkan penulisan *query view* dengan *cross product* dengan banyaknya 100 data, 500 data, dan 1000 data.

- 1) Pengujian 100 Data



Gambar 4. *Response Time* dengan *Query View*

Pada Gambar 4. menunjukkan *response time* menggunakan *query view* untuk menampilkan 100 data dengan waktu 0,0033 detik.



Gambar 5. *Response Time* dengan *Cross Product*

Pada Gambar 5. menunjukkan *response time* menggunakan *cross product* untuk menampilkan 100 data dengan waktu 0,0047 detik.

Dalam Tabel 2. menunjukkan tampilan isi data sebanyak 100 data. Diawali data pertama adalah nama Yosi Wulandari yang mengambil mata kuliah Kimia Dasar IA dan diakhiri data ke 100 adalah nama Tri Wahyu Sarwiyata yang mengambil mata kuliah Akuntansi Manajemen.

Tabel 2. *Record 100 Data*

No	Nama	Jurusan	Nama_mk	Nama_dosen	Nilai
1	Yosi Wulandari	Kedokteran Hewan	Kimia Dasar IA	Dodi Muryanto	80
2	Nurina	Kesehatan Lingkungan	Kimia Dasar IA	Nuraini	70

No	Nama	Jurusan	Nama_mk	Nama_dosen	Nilai
3	Ario	Keselamatan & Kesehatan Kerja	Tata Tulis Karya Ilmiah	Kurnia Sari	65
4	Roni	Farmasi	Pengantar Keilmuan MIPA	Bobby	74
5	I Wayan Sudana M.K	Nutrisi & Teknologi Pangan	Mekanika	Nono	87
...					
100	Tri Wahyu Sarwiyata	Tata Boga	Akuntansi Manajemen	Nirina	90

2) Pengujian 500 Data

✔ Showing rows 0 - 499 (500 total, Query took 0.0086 seconds.)

Gambar 6. *Response Time* dengan *Query View*

Pada Gambar 6 menunjukkan response time menggunakan query view untuk menampilkan 500 data dengan waktu 0,0086 detik.

✔ Showing rows 0 - 499 (500 total, Query took 0.0126 seconds.)

Gambar 7. *Response Time* dengan *Cross Product*

Pada Gambar 7 menunjukkan response time menggunakan cross product untuk menampilkan 500 data dengan waktu 0,0126 detik.

Dalam Tabel 3 menunjukkan tampilan isi data sebanyak 500 data. Diawali data pertama adalah nama Yosi Wulandari yang mengambil mata kuliah Kimia Dasar IA dan diakhiri data ke 500 adalah nama Yosi Wulandari yang mengambil mata kuliah Fisika Dasar IA.

Tabel 3. *Record* 500 Data

No	Nama	Jurusan	Nama_mk	Nama_dosen	Nilai
1	Yosi Wulandari	Kedokteran Hewan	Kimia Dasar IA	Dodi Muryanto	80
2	Nurina	Kesehatan Lingkungan	Kimia Dasar IA	Nuraini	70
3	Ario	Keselamatan & Kesehatan Kerja	Tata Tulis Karya Ilmiah	Kurnia Sari	65
4	Roni	Farmasi	Pengantar Keilmuan MIPA	Bobby	74
5	I Wayan Sudana M.K	Nutrisi & Teknologi Pangan	Mekanika	Nono	87
...					
500	Anjeli	Kedokteran Hewn	Fisika Dasar IA	Sharena	90

3) Pengujian 1000 Data

✔ Showing rows 0 - 999 (1000 total, Query took 0.0152 seconds.)

Gambar 8. *Response Time* dengan *Query View*

Pada Gambar 8 menunjukkan response time menggunakan query view untuk menampilkan 1000 data dengan waktu 0,0152 detik.

✔ Showing rows 0 - 999 (1000 total, Query took 0.0164 seconds.)

Gambar 9. *Response Time* dengan *Cross Product*

Pada Gambar 9 menunjukkan response time menggunakan cross product untuk menampilkan 1000 data dengan waktu 0,0164 detik.

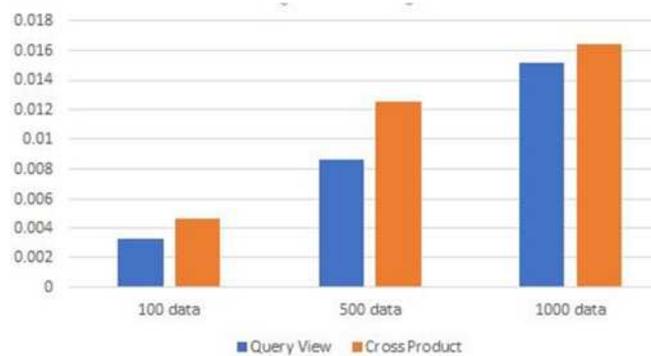
Dalam Tabel 4 menunjukkan tampilan isi data sebanyak 1000 data. Diawali data pertama adalah nama Yosi Wulandari yang mengambil mata kuliah Kimia Dasar IA dan diakhiri data ke 1000 adalah nama f.dr.Sutomo Kasiman yang mengambil mata kuliah Pengantar Akuntansi.

Tabel 4. Record 1000 Data

No	Nama	Jurusan	Nama_mk	Nama_dosen	Nilai
1	Yosi Wulandari	Kedokteran Hewan	Kimia Dasar IA	Dodi Muryanto	80
2	Nurina	Kesehatan Lingkungan	Kimia Dasar IA	Nuraini	70
3	Ario	Keselamatan & Kesehatan Kerja	Tata Tulis Karya Ilmiah	Kurnia Sari	65
4	Roni	Farmasi	Pengantar Keilmuan MIPA	Bobby	74
5	I Wayan Sudana M.K	Nutrisi & Teknologi Pangan	Mekanika	Nono	87
...					
1000	Sutomo Kasiman	Sosiologi	Pengantar Akuntansi Lanjutan	Dani Ismlyatie	90

3.5. Pembahasan

Berdasarkan pengujian yang telah dilakukan untuk menampilkan tabel view_nilai menggunakan penulisan model query view dan cross product diperoleh hasil yang ditunjukkan oleh gambar 10 grafik perbandingan response time query view dan cross product seperti berikut :



Gambar 10. Grafik Perbandingan Response Time

Pada Gambar 10 merupakan grafik perbandingan response time dengan penulisan query view dan cross product pada 100 data, 500 data, dan 1000 data dimana akses dilakukan melalui local server. Hal ini menunjukkan bahwa untuk menampilkan tabel nilai dengan 100 data menggunakan query view dengan selisih 0.0014 detik lebih cepat, dengan 500 data menggunakan query view dengan selisih 0.0040 detik lebih cepat, dan 1000 data menggunakan query view dengan selisih 0.0012 lebih cepat daripada menggunakan cross product.

4. KESIMPULAN

Berdasarkan hasil pengujian yang sudah dilakukan kesimpulan dapat diuraikan sebagai berikut :

1. Dalam penggunaan *query view* harus dibuatkan dulu tabel *view* dan setelah itu baru bisa dilakukan panggilan menggunakan *query select * from view_nilai* yang bertujuan untuk menampilkan data lebih cepat, Akan tetapi *query view* berdampak pada kapasitas *storage* database apa bila pengguna membuat beberapa tabel *query view* untuk pemanggilan *view*.
2. Dalam penelitian ini dilakukan 3 kali pengujian dan penulis dapat menyimpulkan bahwa *query view* lebih cepat dan efisien dengan perbandingan yang tidak terlalu signifikan. Pada saat menampilkan data, *query view* dan *query Cross Product* menghasilkan *response time* 100 data dengan selisih 0,0014 detik, 500 data dengan selisih 0,0040 detik, dan 1000 data dengan selisih 0,0012 detik.

5. SARAN

Adapun saran pada penelitian, yaitu dapat dilakukan pengembangan dengan menerapkan suatu metode, seperti *Heuristic*. Dan juga dapat membandingkan dengan model penulisan query lain, seperti model penulisan *subset query* menggunakan tabel partisi guna membandingkan respon waktu yang lebih cepat.

Daftar Pustaka

1. Maryanto, B. (2017). Big Data dan Pemanfaatannya dalam Berbagai Sektor. *Media Informatika*, Vol. 16 No. 2.
2. Nugroho, B. (2005). *Database Relasional dengan MySQL*. Yogyakarta : ANDI.
3. Péter Jacsó, F. W. (1999). *BUILD YOUR OWN DATABASE*. Chicago and London: AMERICAN LIBRARY ASSOCIATION.
4. Kusriani. (2007). Strategi Perancangan dan Pengelolaan Basis Data. Yogyakarta: Penerbit Andi.
5. Michael Widenius, D. A. (2002). *MySQL Reference Manual Documentation from the Source*. Helsingfors, Finland: O'REILLY COMMUNITY PRESS.
6. Harwani, B. M. (2019). *Foundation Joomla!* New York City: Apress.
7. Lubis, J. H. (2017). ANALISA PERFORMANSI QUERY PADA DATABASE SMELL. *Jurnal Mantik Penusa*, ISSN:2088-3943.
8. Michael R. Groh, J. C. (2007). *Access 2007 Bible*. Indianapolis, Indiana: Wiley Publishing, Inc.
9. Martha, R., Firdaus, Y., & Laksitowening, K. A. (2010). Analisa Perbandingan Response Time dan Throughput pada XML dan DBMS sebagai Media Penyimpanan Data. *Seminar Nasional Aplikasi Teknologi Informasi*, ISSN : 1907-5022.
10. Fatkhurrochman, Wildan L, A., & Juwari. (2018). Analisis Perbandingan Cartesian Product, Cross Join, Inner Join dan Outer Join dalam Si Akad. *TECHNO*, Vol. 19, No.1.
11. Sulhan, M., & Anshori, I. (2016). Perbandingan Subset Query pada Multiple Relasi menggunakan Tabel Terpartisi dan Tabel Tidak Terpartisi dengan Metode Cost-Based. *SMATIKA Jurnal*, Volume 06, No. 2.
12. Suhartati, & Atma, Y. D. (2017). Optimasi Query Sederhana Guna Kecepatan Query pada Database Server. *Metik Jurnal*, P-ISSN: 2580-1503.

Biodata Penulis

Penulis pertama bernama Noviyanti. P, lahir di Pontianak, 29 September 1993, yang beralamat di Desa Salatiga, Kecamatan Mandor, Kabupaten Landak, Kalimantan Barat. Tahun 2019 lulus program sarjana yaitu strata 1 di Universitas Tanjungpura Pontianak program studi

rekayasa sistem komputer. Tahun 2019 sedang menempuh program pascasarjana yaitu magister di Universitas Amikom Yogyakarta.

Penulis kedua bernama Agatha Deolika, lahir di Purwodadi, 3 November 1994, yang beralamat di Jalan Ahmad Yani Kecamatan Teweh Tengah, Kabupaten Barito Utara, Kalimantan Tengah. Tahun 2019 lulus program sarjana yaitu strata 1 di STIMIK Banjar baru program studi teknik informatika. Tahun 2019 sedang menempuh program pascasarjana yaitu magister di Universitas Amikom Yogyakarta.

Penulis ketiga bernama Siti Hartinah, lahir di Tanah Merah, 2 September 1996, yang beralamat di Jalan Batang Tuaka, Tembilahan Pekanbaru. Tahun 2017 lulus program sarjana yaitu strata 1 di Universitas Amikom Yogyakarta program studi sistem informasi. Tahun 2019 sedang menempuh program pascasarjana yaitu magister di Universitas Amikom Yogyakarta.

Penulis keempat bernama Celine Aloyshima Haris, lahir di Balikpapan, 22 Juni 1998, yang beralamat di Jalan Mulawarman Gang Wahid RT.39 No.02 Balikpapan, Kalimantan Timur. Tahun 2019 lulus program sarjana yaitu strata 1 di Universitas Mulawarman Samarida program studi teknik informatika. Tahun 2019 sedang menempuh program pascasarjana yaitu magister di Universitas Amikom Yogyakarta.

Penulis kelima bernama Tutik Maryana, lahir di Magelang, 11 Maret 1995, yang beralamat di Gemulung, Kelurahan Banyuwangi, Kecamatan Bandongan, Magelang, Jawa Tengah. Tahun 2017 lulus program diploma yaitu diploma 3 di Universitas Amikom Yogyakarta, tahun 2018 lulus program sarjana yaitu strata 1 di Universitas Amikom Yogyakarta dan tahun 2019 lulus program sarjana yaitu strata 1 di Universitas Amikom Yogyakarta program studi sistem informasi. Tahun 2019 sedang menempuh program pascasarjana yaitu magister di Universitas Amikom Yogyakarta.

Penulis keenam bernama Nindy Devita Sari, lahir di Purwodadi, 19 April 1998, yang beralamat di Kerujon, Kecamatan Semendawai Suku III, Sumatera Selatan. Tahun 2019 lulus program sarjana yaitu strata 1 di Universitas Amikom Yogyakarta program studi teknik informatika. Tahun 2019 sedang menempuh program pascasarjana yaitu magister di Universitas Amikom Yogyakarta.