PERBANDINGAN *CROSS-PRODUCT* DAN *SUBSET QUERY* PADA *MULTIPLE* RELASI MENGGUNAKAN PARTISI TABEL DENGAN METODE COST-BASED

Isa Anshori
Moh. Sulhan

[1] Sistem Informasi, Universitas Kanjuruhan Malang, isya_it@yahoo.com
[2] Sistem Informasi, Universitas Kanjuruhan Malang, hans2net@yahoo.com

ABSTRAK

Penggunaan *Cross Product* dan *Subset Query* menggunakan tipe data yang sama dalam tabel atau indeks menghasilkan dua model penggunaan *query*, namun perlu diingat yang adalah cara yang lebih optimal sehingga pada akhirnya mendapat *query* dengan waktu akses yang paling *minimum* dengan menggunakan partisi, untuk mengetahui perbedaan hasil query yang menghasilkan tabel partisi menggunakan atau tidak menggunakan partisi.

Hasil yang diperoleh ketika *query* untuk data dengan kelompok besar untuk waktu tercepat adalah kombinasi dari 1 untuk mengakses bagian data, dan kemudian kelompok data sedang dengan akses data penuh untuk *query* tercepat adalah kombinasi dari data kecil 1.Group dengan waktu tercepat adalah *query* dengan tabel partisi tipe skalar dengan 3 catatan. Dengan pengujian *online* untuk akses jaringan *wireless* dan lan Card dengan *bandwidth* 5 Mb.

ABSTRACT

Use of Cross Product and Subset Query using the same data type in the table or index generates two models use the query, but keep in mind that is more optimal way so that in the end gets the query with the most minimum access time by using partitions, to determine differences in query results which resulted in the partition table using or not using partitions.

The results obtained when querying for data with a large group for the fastest time is a combination of 1 to access the data section, and then group the data is being with full data access for the fastest query is a combination of small data 1.Group with the fastest time is the query with the partition table scalar type with 3 notes. With online testing for network access and wireless lan card with a bandwidth of 5 Mb.

Keyword : *Cross Product Comparison and Subset Queries, Multiple Relationships, Partition Table, Cost-Based.*

1. Pendahuluan

Secara umum, kebanyakan perangkat lunak menggunakan sistem basis- data mandiri yang digunakan oleh pengguna untuk menyimpan dan melihat kembali informasi yang diinginkan. Kebanyakan perangkat lunak seperti ini digunakan untuk kegiatan bisnis, pemerintahan, atau kegiatan administrasi lainnya. Untuk tujuan penggunaan yang selalu mengalami perubahan, misalnya bisnis, basisdata yang digunakan tentu harus dapat diakses dengan cepat. Selain itu, proses pengaksesan tersebut tidak boleh memberatkan perangkat keras yang digunakan, sehingga sistem tidak terganggun dan proses bisnis tetap berjalan.

Dari hasil penelitian yang dilakukan oleh (Metta Santiputri dkk, 2010) bahwa *Subset query* pada bentuk scalar dan kombinasi 1 merupakan model yang lebih baik dalam pencarian data jika

dibandingkan dari hasil *cross-product*, baik dalam 2 relasi maupun 3 relasi. Jika menggunakan model dalam pencarian data, maka pada bagian kolom yang diuji harus diindekskan. Pengujian yang dilakukan menggunakan data paling besar 500.000 data, kemungkinan jika data diatara 1.000.000 sampai 2.000.000 waktu akses akan lama dan belum meneliti tentang penggunaan partisi pada tabel dengan menerapkan *cross product* dan *Subset query*.

Penggunaan *cross product* dan *Subset query* dengan menggunakan tipe data sama atau indeks di tabel menghasilkan *query* dengan Dua model *query* yang umum antara lain adalah *cross product* dan *subset query*, dimana kedua model ini dapat menghasilkan data yang sama. Namun perlu diperhatikan cara mana yang lebih optimal sehingga pada akhirnya didapatkan *query* dengan waktu akses yang paling minimum pada tabel yang menggunakan partisi, agar bisa mengetahui perbedaan hasil *query* yang di hasilkan antara tabel yang menggunakan partisi atau tidak menggunakan partisi.

Berdasarkan uraian di atas maka perlu dilakukan penelitian untuk mendapatkan hasil optimal untuk mengakses data di *database* dengan menggunakan *cross product* dan *subset query* dengan menggunakan metode *cost-based* untuk pengujian yang akan di pakai.

2. Tinjauan Pustaka

MySQL adalah salah satu aplikasi *Database Management System* (DBMS) yang menggunakan fungsi relasional dalam pengaksesan basis data. MySQL berjalan sebagai sebuah server sehingga akses basis data dapat dilakukan oleh beberapa pengguna sekaligus. MySQL dapat berjalan dengan baik dengan semua sistem operasi dan dapat diimplementasikan dengan berbagai API (Application Programming Interface) yang berjalan dalam bahasa C, C++. C#, Java PHP Delphi, dan lainnya (Margaretha Siahaan, 2010/2011).

MySQL Merupakan bahasa pemrograman RDBMS (*Relational Database Management System*) open source yang paling popular dan banyak digunakan di sistem operasi LINUX (juga tersedia pada sistem operasi WINDOWS). Kepopuleran ini karena ditunjang oleh performansi *query* dari *database*-nya yang jarang bermasalah (Allen, 2002, pg.220).

Subset query pada bentuk scalar dan kombinasi 1 merupakan model yang lebih baik dalam pencarian data jika dibandingkan dengan cross product, baik dalam 2 relasi maupun 3 relasi. Jika menggunakan model cross product dalam pencarian data, maka pada bagian kolom yang dikondisi harus diindekskan. Perlu diperhatikan bahwa penelitian ini dilakukan pada tabel non partisi, sehingga untuk penelitian lebih lanjut perlu dilakukan juga pada tabel yang menggunakan partisi. Tapi sebelum-nya melakukan partisi tabel cek lebih dahulu apa databasenya suport partisi atau tidak.

adalah membuat partisi berdasarkan sebuah range, misalkan berdasarkan tanggal, sebuah nilai, contoh nya seperti di bawah ini

```
CREATE TABLE tbl_penjualan (
id INT NOT NULL,
nama VARCHAR(30),
tgl_transaksi DATE NOT NULL DEF
AULT '1970-01-01'  )
PARTITION BY RANGE ( YEAR(tgl_t
ransaksi) ) (
PARTITION p0 VALUES LESS THAN (
1990),
PARTITION p1 VALUES LESS THAN (
1995),
PARTITION p2 VALUES LESS THAN (
2000),
PARTITION p3 VALUES LESS THAN
 MAXVALUE
```

3. Pembahasan

3.1 Analisa

Studi kasus yang diambil sebagai contoh uji coba adalah *database* yang ada di sistem penerimaan mahasiswa baru di Universitas Kanjuruhan Malang. Sistem penerimaan mahasiswa baru ini bertujuan untuk melakukan perhitungan jumlah mahasiswa yang mendaftar dari waktu yang di tentukan dari pihak Universitas Kanjurhan Malang. Selain Pendaftaran dan pendataan oleh panitia PMB (Penerimaan Mahasiswa Baru) yang dilakukan manual juga menggunakan sistem aplikasi berbasi web, Aplikasi ini tentunya membantu untuk melakukan

pencarian dan pendataan yang nanti akan dilakukan proses selanjutnya ketika mahasisnya sudah diterima. Data akan terpusat di satu database penerimaan mahasiswa baru.

Secara definisi partisi tabel adalah memecah tabel menjadi beberapa bagian/segmen, tujuan dari menggunakan partisi adalah memungkinkan untuk mempercepat proses query. Misalkan kita memilik data sebanyak 10 juta record, bila pada tabel *konvensional* maka data sebanyak itu akan di simpan dalam satu segmen, tetapi bila menggunakan partisi maka data sebanyak itu akan di pecah-pecah ke banyak partisi berdasarkan sebuah kondisi, misalkan berdasarkan tanggal, maka ketika melakukan query hanya men-scan segmen dimana data itu berada, tidak 10 juta record itu di scan, sehingga kemungkinan proses query menjadi lebih cepat. Hasil query yang di dapat tentunya adalah dari hasil query dengan partisi tabel dengan berbagai kemungkinan untuk mengetahui hasil perbandingan.

3.2 Analisa Kebutuhan Uji

Pada setiap data yang tersimpan di database ada bebarapa faktor yang mempengaruhi untuk kecepatan akses data yang akan di tampilkan,

a. Jumlah Data (Tanpa Partisi Tabel)
b. Jumlah Data (Partisi Tabel)
c. Jumlah Kolom (Partisi Tabel)
d. Jumlah Kolom (Tanpa Partisi Tabel)
e. Relasi Antar Tabel (Tanpa Partisi Tabel)
f. Relasi Antar Tabel (Partisi Tabel) dan Menggunakan Index
g. Range Partisi 1000,2500,5000,10000 dan range 5000,10000
h. Pengujian Server Lokal dan Online

Untuk Penjelasan query yang di gunakan dalam pengujian di faktor-faktor diatas akan di jelaskan di bagian Benchmark Query di bagian 3.4.

Dalam pengujian yang di lakukan di server online dengan spesifikasi kebutuhan uji untuk perbandingan sebagai berikut :

a. Bandwith 5 MB.
b. Jarak akses skitar 10 Meter
c. Akses Menggunakan jaringan *Wireless* Analisa data Master di *database*

3.3 Skenario Pengujian

Pengujian model query dibagi atas beberapa group. Group dengan perbedaan jumlah data dan group dengan atau Telah menggunakan partisi tabel. Sebagai uji coba penelitian, tabel jenis yang akan di uji jumlah data dan tabel yang menggunakan pasrtisi dengan Skala jumlah data yang terlihat pada tabel, hal inidilakukan untuk memudahkan uji eksekusi query dengan jumlah data yang berbeda-beda.

Tabel 3.2 Skala Data Pengujian

No	Nama	Jumlah Data	Group	Keterangan
1	Data_1000	1.000	Kecil	Limit 1000
2	Data_2500	2.500	Sedang	Limit 2500
3	Data_5000	5.000	Besar	Limit 5000
4	Data_12713	12.713	Sangat Besar	Jumlah 12713

a. Skenario Analisa 3 tabel Query *Subset*

Uji yang dilakukan dengan menggunakan partisi range yang berbeda dengan kriteria untuk 5000, 10000,*maxvalue* range yang digunakan.

Query yang digunakan

```
selectREGNO,NAMA,ALAMAT_ASAL,KD
_AGAMA,KODEPRODI
from datacama_hist,tmprodi
wheredatacama_hist.KODEPRODI=tm
prodi.kode and
datacama_hist.KD_AGAMA in
(select KD_AGAMA from tmagama)
```

b. Skenario Analisa 3 tabel partisi range berbeda dengan *Cross product*

Uji yang dilakukan dengan menggunakan partisi range yang berbeda dengan kriteria untuk 1000, 5000, 10000, *maxvalue* range yang digunakan.

```
Select
REGNO,NAMA,TEMP_LAHIR,ALAMAT_AS
AL, datacama_hist.KD_AGAMA,
KODEPRODI from
datacama_hist,tmagama,tmprodi
where
datacama_hist.KD_AGAMA=tmagama.
KD_AGAMA and
datacama_hist.KODEPRODI=tmprodi
.kode
```

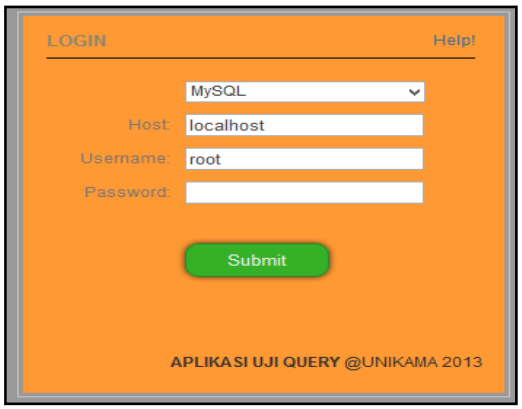
3.4 Skenario *Query* Pengujian di Server Lokal

Untuk tabel yang digunakan di pengujian kategori ini adalah dengan mengakses tabel secara penuh dengan menggunakan dua model *query* yaitu Cross Product dan Scalar dengan berbagai group data dari kecil sampai sangat besar, untuk jenis tabel belum terpartisi.

4. Hasil Uji Query dan Analisa

a. Form Login

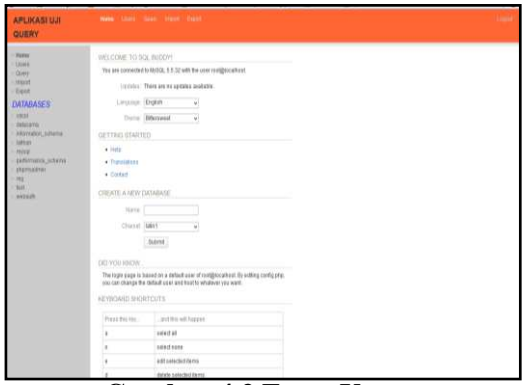
Halaman login ini adalah tampilan awal sebelum masuk di menu utama di sistem pengujian *query* yang telah di hasilkan. Form ini berisi tentang setting awal untuk koneksi antara program dan database yang telah tersedia dengan pengisian jenis database, host atau alamat server, user dan password.



Gambar 4.1 Form Login Aplikasi

b. Halaman Utama Pengguna

Form ini adalah tampulan menu utama dari aplikasi uji *query*. Di menu ini user bisa memilih database yang akan di gunakan pada waktu pengujian, dan Form ini akan tampil apabila proses login telah sukses.

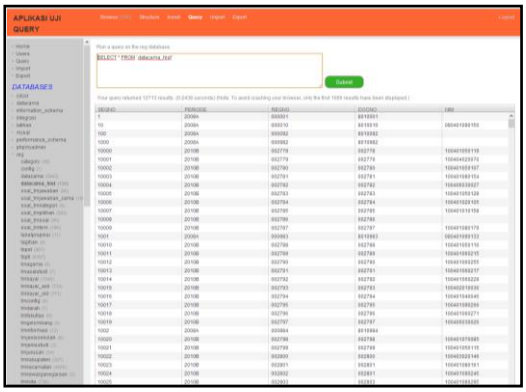


Gambar 4.2 Form Utama

c. Form Input *Query*

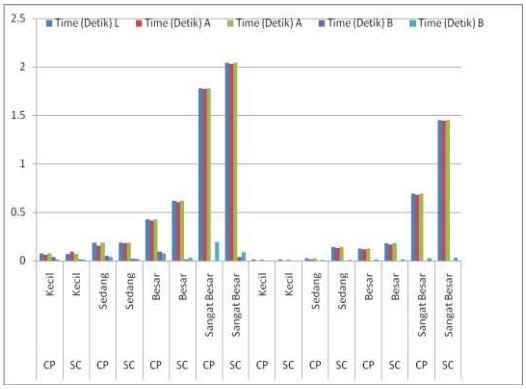
Form ini digunakan untuk memasukkan *query* yang akan di jalankan untuk mengetahui hasil data dan waktu yang bisa terbaca di sistem, ketika *query* sudah selesai di inputkan maka harus click tombol submit untuk mengetahui data yang dihasilkan dan waktu yang tercatat untuk lama eksekusi *query* yang di masukkan.

Untuk hasil data setelah di submit maka jumlah data yang di hasilkan dengan keterangan berapa record yang di hasilkan, untuk waktu yang tercatat ada dalam hitungan detik



Gambar 4.3 Form Input *Query*

d. Hasil Analisa Semua Kriteria Dari Tabel Pengujian



Gambar 4.17 Pengujian 2 Tabel Partisi Pada Server Online semua kriteria

Keterangan :

- Time detik L : database di server local
- Time detik A : database server online dengan range partisi 1000,2500,5000,10000 dengan akses via wireless

- Time detik A : database server online dengan nrange partisi 1000 , 2500, 5000,10000 dengan akses via lan card
- Time detik B : database server online dengan range partisi 5000,10000 dengan akses via lan wireless
- Time detik B : database server online dengan range partisi 5000,10000 dengan akses via lan card

Dapat dilihat dari grafik waktu yang di hasilkan oleh Aplikasi Penguji *Query* ketika *query cross product* dan subset *query* dengan jenis *scalar* di jalankan, untuk hasil waktu dapat di lihat di tabel 5.2 hasil pengujian.

Hasil yang diperoleh saat *Query* dengan data sangat besar untuk akses data sebagian dan penuh waktu yang paling cepat adalah *query* dengan model range partisi lebih sedikit dengan range 5000,1000 Dan dapat dilihat pada gambar dengan keterangan time B untuk waktu yang diperoleh jelas perbedaan sangat besar dibandingkan pada tabel A dengan rang partisi kecil 1000,2500,5000,1000 lihat pada gambar 4.17 hasil pengujian dengan Kriteria Range Partisi 5000 dan 1000 data dengan keterangan time (detik) B.

4. Kesimpulan

Berdasarkan hasil uji *query* dan analisa partisi tabel pada *multiple* relasi yang pengambilan data di Sistem Penerimaan Mahasiswa Baru Universitas Kanjuruhan Malang Menggunakan Metode *Cost Based* ini, maka dapat diambil kesimpulan bahwa pengujian yang dilakukan untuk hasil perbedaan *query* yang di akses di tabel belum partisi dan tabel yang telah di partisi waktu yang paling cepat adalah dari tabel yang dipartisi terbukti Untuk tabel yang di partisi waktu yang di perlukan 0.0736 sedangkan waktu yang diperlukan untuk akses data pada tabel nonpartisi 0.1725. Untuk akses data yang di akses menggunakan lan card lebih cepat dibandingkan akses dengan menggunakan jaringan *wireless* karena akses data dengan menggunakan wireless dipengaruhi jarak dari akses data.

Hasil yang diperoleh saat *Query* dengan data sangat besar untuk akses data sebagian dan penuh waktu yang paling cepat adalah *query* dengan model range partisi lebih sedikit dengan range 5000,1000 Dan dapat dilihat pada gambar dengan keterangan time B untuk waktu yang diperoleh jelas perbedaan sangat besar dibandingkan pada tabel A dengan rang partisi kecil 1000,2500,5000,1000 lihat pada gambar 4.17 hasil pengujian dengan Kriteria Range Partisi 5000 dan 1000 data dengan keterangan time (detik) B.

Subset query pada bentuk scalar dan kombinasi 1 merupakan model yang lebih baik dalam pencarian data jika dibandingkan dengan cross product,baik dalam 2 relasi maupun 3 relasi dengan indexs dan tipe data yang sama

5. Saran

Pada uji dan analisa ini telah membantu dalam memberikan gambaran dan hasil untuk uji *query* yang di partisi tabel, namun dalam menganalisa partisi masih jauh dari sempurna. Dalam pengujian ini terdapat kekurangan untuk bisa di kembangkan lagi diantaranya yaitu :

1. Untuk penelitian uji kecepatan *query* selanjutnya diharapkan menggunakan banyak jenis partisi untuk mengetahui partisi yang cocok untuk desain *database* sebuah sistem.
2. Untuk penelitian lebih lanjut diharapkan agar bisa diimplementasikan pada metode selain yang digunakan penulis dalam pengujian *query*.
3. Untuk penelitian uji kecepatan *query* selanjutnya diharapkan lebih menggunakan jumlah data missal data lebih dari 10.000.000 data untuk di analisa dengan berbagai macam optimasi tabel.

6. Daftar Pustaka

Chen,P.P.,1976. The Entity-Relationship Model: Toward a Unified View of Data, ACM Transactions on Database Systems.

Elmasri, Ramez dan Navathe, Shamkant B. 2003. Fundamentals Of Database Design, Fourth Edition. Pearson-Addison Weasley.

Immanuel Chan, Oracle Database Performance Tuning Guide, 10g Release 2

(10.2),Redwood City, CA, Maret 2008, pp. 379-403.

Kusrini, S. Kom., Optimasi Query Untuk Pencarian Data dengan Subset Query, Bandung, 2006.

Lightstone, Sam., Teorey, Toby., and Nadeau, Tom. (2007). Physical Database Design: The Database Professional's Guide to Exploiting Indexes, Views, Storage, and More, Morgan Kaufmann Publishers.

Munir, Rinaldi. 2009. Diktat Kuliah IF3051 Strategi Algoritma.Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung

Powel, Gavin. 2006. Beginning Database Design, Wiley Publishing, Inc.,10475 Crosspoint Boulevard.

Setiawan, M.A., Optimasi SQL Query untuk Informasi Retrievel pada Aplikasi Berbasis Web, Proceedings Seminar Nasional Aplikasi Teknologi Informasi UII, Yogyakarta, 2004.

Sandra Cheevers, Oracle Database Product Familiy, An Oracle White Paper, Redwood Shores, CA USA, Agustus 2006, pp 3-4.

Sagi Arsyad, Pengenalan .NET dan C#, Microsoft Innovation Center, Universitas Indonesia, Jakarta, 2008.

Silberschatz−Korth−Sudarshan.2001. Database System Concepts, Fourth Edition, The McGraw−Hill Inc.

Sumathi, S. and Esakkirajan, S. 2007. Fundamentals of Relational Database Management Systems, SpringerVerlag Berlin Heidelberg.

Tom Best dan M.J. Billings, Oracle Database 10g: Administration Workshop I,Electronic Presentation, Redwood Shores, California USA, November 2005, pp. 41-42.

Kofler, Michael. 2005. The Definitive Guide to MySQL 5, Third Edition. Apress.

MySQL 5.1 Reference Manual. http://dev.mysql.com/doc/refman/5.1/en/partitioning.html diakses pada [20 Oktober 2013, 09:00]