

ISSN 2087-0256

smatika Jurnal

STIKI Informatika Jurnal

Volume 06, Nomor 02 Tahun 2016



smatika Jurnal

ISSN 2087-0256

STIKI Informatika Jurnal

Volume 06, Nomor 02 Tahun 2016

Perbandingan *System Functionality*, *System Interactivity*, dan *Usability* pada *Instant Messaging* (IM) sebagai Media Pembelajaran Sinkron

Faizatul Amalia, Admaja Dwi Herlambang, Tri Afirianto

Peran *E-Journal* dalam *Knowledge Sharing* sebagai Basis Pengelolaan Pengetahuan di Universitas Kristen

Satya Wacana

Suroyo, Andeka Rocky Tanaamah

Penjaminan Kualitas Perangkat Lunak *Learning Management System Open Source* di Politeknik Kota

Malang

Betta Wahyu RM, Dwi Wijonarko

Perbandingan *Subset Query* pada *Multiple Relasi* Menggunakan Tabel Terpartisi dan Tabel Tidak

Terpartisi dengan Metode *Cost-Based*

Moh Sulhan, Isa Anshori

Prediksi Volume Sampah TPAS Talangagung dengan Pendekatan Sistem Dinamik

Philip Faster Eka Adipraja, Mufidatul Islamiyah

Penerapan Metode Naive Bayes dalam Pengklasifikasi Trafik Jaringan

Sigit Riyadi



LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT

STIKI

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA

Jl. Raya Tidar 100, Malang; Phone: 0341-560823; Fax: 0341-562525; <http://www.stiki.ac.id>; mail@stiki.ac.id

PENGANTAR REDAKSI

STIKI Informatika Jurnal (SMATIKA Jurnal) merupakan jurnal yang diterbitkan oleh Lembaga Penelitian & Pengabdian kepada Masyarakat (LPPM), Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang.

Pada edisi ini, SMATIKA Jurnal menyajikan 6 (*enam*) naskah dalam bidang sistem informasi, jaringan, pemrograman web, perangkat bergerak dan sebagainya. Redaksi mengucapkan terima kasih dan selamat kepada Pemakalah yang diterima dan diterbitkan dalam edisi ini, karena telah memberikan kontribusi penting pada pengembangan ilmu dan teknologi.

Pada kesempatan ini, redaksi kembali mengundang dan memberi kesempatan kepada para Peneliti di bidang Teknologi Informasi untuk mempublikasikan hasil-hasil penelitiannya melalui jurnal ini. Bagi para pembaca yang berminat, Redaksi memberi kesempatan untuk berlangganan.

Akhirnya Redaksi berharap semoga artikel-artikel dalam jurnal ini bermanfaat bagi para pembaca khususnya dan bagi perkembangan ilmu dan teknologi di bidang Teknologi Informasi pada umumnya.

REDAKSI

smatika Jurnal

ISSN 2087-0256

STIKI Informatika Jurnal

Volume 06, Nomor 02 Tahun 2016

Pelindung
Yayasan Perguruan Tinggi Teknik Nusantara

Penasehat
Ketua STIKI

Pembina
Pembantu Ketua Bidang Akademik STIKI

Mitra Bestari
Prof. Dr. Ir. Kuswara Setiawan, MT (UPH Surabaya)
Dr. Ing. Setyawan P. Sakti, M.Eng (Universitas Brawijaya)

Ketua Redaksi
Subari, M.Kom

Section Editor
Jozua F. Palandi, M.Kom
Nira Radita, S.Pd., M.Pd

Layout Editor
Saiful Yahya, S.Sn, MT.

Tata Usaha/Administrasi
Muh. Bima Indra Kusuma

SEKRETARIAT

Lembaga Penelitian & Pengabdian kepada Masyarakat
Sekolah Tinggi Informatika & Komputer Indonesia (STIKI)
Malang

smatika jurnal

Jl. Raya Tidar 100 Malang 65146

Tel. +62-341 560823

Fax. +62-341 562525

Website: jurnal.stiki.ac.id

E-mail: jurnal@stiki.ac.id, lppm@stiki.ac.id

DAFTAR ISI

Perbandingan <i>System Functionality</i>, <i>System Interactivity</i>, dan <i>Usability</i> pada <i>Instant Messaging</i> (IM) sebagai Media Pembelajaran Sinkron	01 - 04
Faizatul Amalia, Admaja Dwi Herlambang, Tri Afirianto	
<hr/>	
Peran <i>E-Journal</i> dalam <i>Knowledge Sharing</i> sebagai Basis Pengelolaan Pengetahuan di Universitas Kristen Satya Wacana	05 - 12
Suroyo, Andeka Rocky Tanaamah	
<hr/>	
Penjaminan Kualitas Perangkat Lunak <i>Learning Management System Open Source</i> di Politeknik Kota Malang	13 - 18
Betta Wahyu RM, Dwi Wijonarko	
<hr/>	
Perbandingan <i>Subset Query</i> pada <i>Multiple</i> Relasi Menggunakan Tabel Terpartisi dan Tabel Tidak Terpartisi dengan Metode <i>Cost-Based</i>.....	19 - 23
Moh Sulhan, Isa Anshori	
<hr/>	
Prediksi Volume Sampah TPAS Talangagung dengan Pendekatan Sistem Dinamik	24 - 28
Philip FASTER Eka Adipraja, Mufidatul Islamiyah	
<hr/>	
Penerapan Metode Naive Bayes dalam Pengklasifikasi Trafik Jaringan.....	29 - 36
Sigit Riyadi	

Undangan Makalah

smatika Jurnal Volume 07, Nomor 01 Tahun 2017

Perbandingan *Subset Query* pada *Multiple Relasi* Menggunakan Tabel Terpartisi dan Tabel Tidak Terpartisi dengan Metode *Cost-Based*

Moh Sulhan¹⁾, Isa Anshori²⁾

^{1,2)} Prodi Sistem Informasi, Fakultas Teknologi Informasi, Universitas Kanjuruhan Malang

Jl. Soedanco Supriadi No.48 Malang

Telp: (0341) 801488, Ext:432

Email:

¹⁾sulhan@unikama.ac.id

²⁾isaanshori@unikama.ac.id

ABSTRAK

Penggunaan Subset Query menggunakan tipe data yang sama dalam tabel atau indeks menghasilkan model penggunaan sub 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.

Kata Kunci: *Perbandingan Subset Queries, Multiple Relasi, Tabel terpartisi, Tabel tidak terpartisi, 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 terganggu dan proses bisnis tetap berjalan.

Dari hasil penelitian yang dilakukan oleh (Metta Santiputri dkk, 2010) bahwa *Subset query* pada bentuk skalar 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 *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 mengakses 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 populer 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).

Berikut 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
  DEFAULT '1970-01-01' )
  PARTITION BY RANGE ( YEAR(tg
l_transaksi) ) (
  PARTITION p0 VALUES LESS THA
N (1990),
  PARTITION p1 VALUES LESS THA
N (1995),
  PARTITION p2 VALUES LESS THA
N (2000),
  PARTITION p3 VALUES LESS THA
N
  MAXVALUE
  );
```

3. METODOLOGI PENELITIAN

Berikut akan dijelaskan pembahasan analisis data, kebutuhan uji coba, serta menentukan skenario pengujian.

a. Analisis

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 Kanjuruhan

Malang. Selain Pendaftaran dan pendataan oleh panitia PMB (Penerimaan Mahasiswa Baru) yang dilakukan manual juga menggunakan sistem aplikasi berbasis 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 tabel terpartisi atau partisi tabel adalah memecah tabel menjadi beberapa bagian/segmen, tujuan dari menggunakan partisi adalah memungkinkan untuk mempercepat proses query. Misalkan kita memiliki 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

b. Analisis Kebutuhan Uji

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

- jumlah data (tanpa partisi tabel)
- jumlah data (partisi tabel)
- jumlah kolom (partisi tabel)
- jumlah kolom (tanpa partisi tabel)
- relasi antar tabel (tanpa partisi tabel)
- relasi antar tabel (partisi tabel) dan Menggunakan Index
- range partisi 1000, 2500, 5000, 10000 dan range 5000, 10000
- pengujian server lokal dan online

Untuk Penjelasan query yang di gunakan dalam pengujian di faktor-faktor di atas akan di jelaskan di bagian Benchmark Query di bagian Skenario Query Pengujian.

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

- bandwith 5 MB.
- jarak akses skitar 10 meter
- akses menggunakan jaringan *wireless*
- analisis data master di *database*

c. 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 1. 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

Skenario Analisis 3 tabel Query Subset

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

Query yang digunakan

```
Select REGNO, NAMA, ALAMAT_ASAL,
KD_AGAMA, KODEPRODI
from
datacama_hist, tmprodi
wheredatacama_hist.KODEPRODI=tmprodi.kode and
datacama_hist.KD_AGAMA in
(select KD_AGAMA from
tmagama)
```

Skenario Analisis 3 tabel Belum partisi range berbeda dengan Subset Query

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

```
Select REGNO, NAMA, ALAMAT_ASAL,
KD_AGAMA, KODEPRODI
from datacama_hist, tmprodi
wheredatacama_hist.KODEPRODI=tmprodi.kode and
datacama_hist.KD_AGAMA in
(select KD_AGAMA from
tmagama)
```

d. Skenario Query Pengujian di Server Lokal

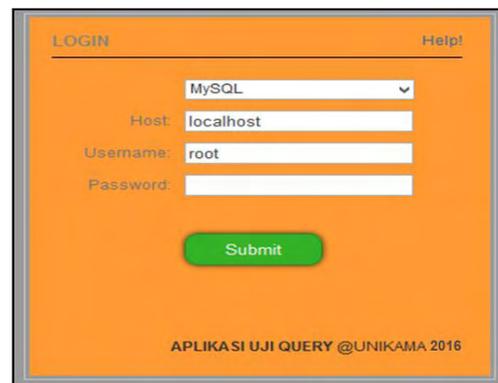
Untuk tabel yang digunakan di pengujian kategori ini adalah dengan mengakses tabel secara penuh dengan menggunakan dua model *query* yaitu *Subset Query* dan *Scalar* dengan berbagai group data dari kecil sampai sangat besar, untuk jenis tabel belum terpartisi.

4. HASIL DAN PEMBAHASAN

Berikut akan dijelaskan hasil uji Query dan penjelasan analisisnya.

a. Form Login

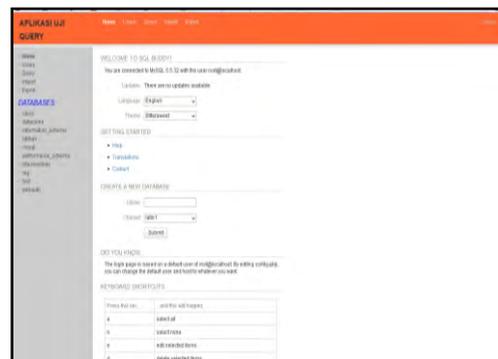
Halaman login ini adalah tampilan awal sebelum masuk di menu utama di sistem pengujian *query* yang telah di hasilkan. Form ini berisi tentang setting awal untuk koneksi antara program dan database yang telah tersedia dengan pengisian jenis database, host atau alamat server, user dan password.



Gambar 1. Form Login Aplikasi

b. Halaman Utama Pengguna

Form ini adalah tampilan 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.

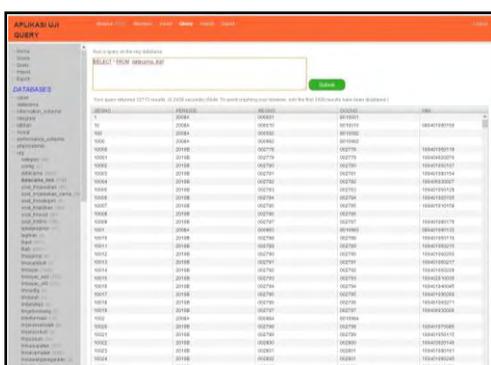


Gambar 2. Form Utama

c. Form Input Query

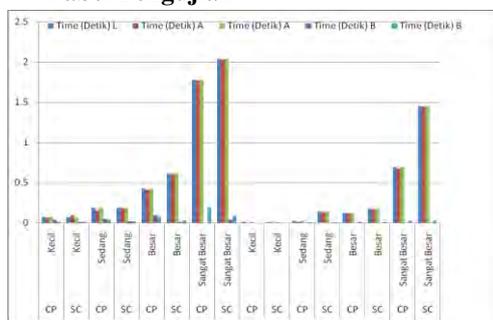
Form ini digunakan untuk memasukkan *query* yang akan di jalankan untuk mengetahui hasil data dan waktu yang bisa terbaca di sistem, ketika *query* sudah selesai di inputkan maka harus click tombol submit untuk mengetahui data yang dihasilkan dan waktu yang tercatat untuk lama eksekusi *query* yang di 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 3. Form Input Query

d. Hasil Analisa Semua Kriteria Dari Tabel Pengujian



Gambar 4. Pengujian 2 Tabel Partisi Pada Server Online semua kriteria

Keterangan:

1. Time detik L: database di server local
2. Time detik A: database server online dengan range partisi 1000, 2500, 5000, 10000 dengan akses menggunakan wireless
3. Time detik A: database server online dengan nrange partisi 1000, 2500, 5000,10000 dengan akses menggunakan Card LAN
4. Time detik B: database server online dengan range partisi 5000,10000 dengan akses menggunakan LAN wireless

5. Time detik B: database server online dengan range partisi 5000,10000 dengan akses menggunakan Card LAN

Dapat dilihat dari grafik waktu yang di hasilkan oleh Aplikasi Penguji *Query* ketika subset *query* dengan jenis *scalar* di jalankan, untuk hasil waktu dapat di lihat pada gambar 3. 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 3. hasil pengujian dengan Kriteria Range Partisi 5000 dan 1000 data dengan keterangan time (detik) B.

5. 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 card LAN lebih cepat dibandingkan akses dengan menggunakan jaringan akses dengan menggunakan *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 table partisi pada bentuk *scalar* dan kombinasi 1 merupakan model

yang lebih baik dalam pencarian data jika dibandingkan dengan *subset query table belum partisi*, baik dalam 2 relasi maupun 3 relasi dengan indexs dan tipe data yang sama

6. REFERENSI

- [1] Best, Tom & Billings. (2005). Oracle Database 10g: Administration Workshop I, Electronic Presentation, Redwood Shores, California USA, November, pp. 41-42.
- [2] Chan, Immanuel. (2008). Oracle Database Performance Tuning Guide, 10g Release 2 (10.2), Redwood City, CA, pp. 379-403.
- [3] Chen, P.P. (1976). The Entity-Relationship Model: Toward a Unified View of Data, ACM Transactions on Database Systems.
- [4] Elmasri, Ramez, Navathe, Shamkant B. (2003). Fundamentals of Database Design, Fourth Edition. Pearson-Addison Wesley.
- [5] Kusriani. (2006). Optimasi Query Untuk Pencarian Data dengan Subset Query, Bandung.
- [6] Lightstone, Sam., Teorey, Toby., & Nadeau, Tom. (2007). Physical Database Design: The Database Professional's.
- [7] MySQL 5.1 Reference Manual. (online) <http://dev.mysql.com/doc/refman/5.1/en/partitioning.html>, diakses pada 20 Oktober 2016.
- [8] Setiawan. (2004). Optimasi SQL Query untuk Informasi Retrieval pada Aplikasi Berbasis Web. *Proceedings Seminar Nasional Aplikasi Teknologi Informasi UII, Yogyakarta*.

