

Optimasi Query untuk Sistem Informasi Penjadwalan Mata Pelajaran Sekolah Menggunakan View (Studi Kasus : SMK VIP Purworejo)

Pipit Kurnia Safitri¹, Wing Wahyu Winarno², Eko Pramono³

^{1,2,3}Magister Teknik Informatika, Universitas AMIKOM Yogyakarta
Jl. Ring Road Utara, Condongcatur, Sleman, Yogyakarta-Indonesia.
¹pipitkurnia665@gmail.com, ²wing@amikom.ac.id, ³eko.p@amikom.ac.id

INTISARI

Perkembangan teknologi pendidikan di Jawa Tengah khususnya di daerah purworejo berkembang dengan pesat. SMK VIP Purworejo saat ini menggunakan sistem informasi penjadwalan berbasis web. Jumlah mata pelajaran yang banyak dan sering berubah sewaktu-waktu sering menimbulkan masalah. Banyaknya jumlah mata pelajaran dan guru yang mengajar maka akan semakin membutuhkan waktu yang lama untuk mengetahui mata pelajaran yang diampu oleh guru tersebut dan untuk mengetahui jam mengajar. View merupakan virtual table dimana isinya (kolom dan baris) didefinisikan dari satu query yang dapat melibatkan beberapa tabel sekaligus. Tujuan dari pembuatan view adalah untuk kenyamanan (mempermudah penulisan query), untuk keamanan (menyembunyikan beberapa kolom yang bersifat rahasia), atau bisa digunakan untuk mempercepat proses menampilkan data (terutama jika kita akan menjalankan query tersebut secara berulang). Eksekusi query basisdata membutuhkan response time. Apabila query yang dieksekusi membutuhkan response time yang lama, maka performansi basisdata berkurang. Salah satu cara untuk meningkatkan performansi basisdata adalah dengan menerapkan view. Tujuan dari penelitian ini adalah untuk mengetahui efisiensi sintak dan performa kecepatan dengan menggunakan query join dan view.

Kata Kunci— Database, Optimasi query, view

ABSTRACT

The development of education technology in Central Java khususnya in the region purworejo growing rapidly. SMK VIP Purworejo currently uses web-based scheduling information system. The number of subjects that are overwhelming and often changing at times often creates problems. The large number of subjects and teachers who teach it will take a long time to find out the subjects being taught by the teacher and to know the hours of teaching. View is a virtual table where the contents (columns and rows) are defined from a single query that can involve multiple tables at once. The purpose of view creation is for convenience (ease of writing queries), for security (hiding some secret columns), or can be used to speed up the process of displaying data (especially if we will run the query repeatedly). Database query execution requires response time. If the query that is executed requires a long response time, then the performance of the database is reduced. One way to improve database performance is by applying view. The purpose of this research is to know the efficiency of syntax and speed performance by using join and view query.

Keywords- Database, query optimization, view

I. PENDAHULUAN

Kehadiran jaringan komputer luas yang dikenal dengan istilah internet telah menyedot perhatian siswa. Didukung dengan mudahnya akses saat mobilitas dan *real time* sistem yang membawa kemudahan dalam pengaksesan data. Kedudukan IT bagi Pendidikan Sudah selayaknya lembaga-lembaga pendidikan yang

ada segera memperkenalkan dan memulai penggunaan teknologi informasi dan komunikasi (TIK) sebagai basis pembelajaran yang lebih mutakhir. Hal ini menjadi penting, mengingat penggunaan IT merupakan salah satu faktor penting yang memungkinkan kecepatan transformasi ilmu pengetahuan kepada para peserta didik, generasi bangsa ini secara lebih

luas. Dalam konteks yang lebih spesifik, dapat dikatakan bahwa kebijakan penyelenggaraan pendidikan, baik yang diselenggarakan oleh pemerintah, pemerintah daerah, maupun masyarakat harus mampu memberikan akses pemahaman dan penguasaan teknologi mutakhir yang luas kepada para peserta didik.

Kemajuan zaman dan teknologi saat ini telah merubah pola dalam perusahaan atau pun organisasi-organisasi. Data-data yang besar dan telah terkumpul dalam waktu yang lama dapat di olah menjadi sumber informasi yang dapat membantu menganalisis eksistensi dari sebuah perusahaan atau organisasi. Analisa otomatis dari data yang berjumlah besar atau kompleks yang terbentuk sebagai data *mining*, dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaannya saat ini telah banyak di lakukan. Pengolahan data tersebut dapat dilakukan dengan mengakses data yang terdapat dalam *database*. Pengaksesan data tersebut dilakukan dengan melakukan *query* pada basisdata dengan *database* manajemen sistem. Pengaksesan data pada *database* perlu memperhatikan kegunaan implementasi dari data itu sendiri serta waktu prosesnya. Ada banyak cara yang dapat dilakukan oleh database manajemen sistem dalam memproses dan menghasilkan jawaban sebuah *query*. Semua cara pada akhirnya akan menghasilkan jawaban (*output*) yang sama tetapi pasti mempunyai harga yang berbeda-beda, seperti misalnya total waktu yang diperlukan untuk menjalankan sebuah *query*.

Database atau sering kita kenal basis data merupakan sekumpulan data yang tersusun dan tersimpan rapi dalam komputer, dan dapat diolah maupun dimanipulasi dengan menggunakan software atau perangkat lunak untuk dijadikan sebagai informasi. *Database* adalah kumpulan informasi atau data yang tersimpan secara sistematis sehingga temu kembali informasinya menjadi mudah dan cepat [1]

View merupakan cara efisien untuk merepresentasikan data tanpa harus memeliharanya. *View* bukan merupakan tabel, dan bersifat *virtual* yang tidak membutuhkan *storage* tetap. *View* memiliki baris dan kolom seperti tabel, namun untuk dapat melakukan *insert*, *update* atau *delete* tergantung pada definisi *view* itu sendiri. Penggunaan *view*

biasanya ditujukan untuk alasan keamanan data yang sensitif. Selain itu *view* juga dapat digunakan untuk mengubah tabel tanpa mempengaruhi aplikasi, memungkinkan untuk mengakses informasi lebih dari satu sumber, dan melakukan *summary* terhadap tabel seperti *sum*, *min*, *max* atau rata-rata [2].

Join table adalah penggabungan tabel menggunakan *query* yang dilakukan melalui kolom/*key* tertentu yang memiliki nilai terkait untuk mendapatkan satu *set* data dengan informasi lengkap. Lengkap disini artinya kolom data didapatkan dari kolom-kolom hasil *join* antar tabel tersebut. *Join* diperlukan untuk mengurangi redundansi. Pengaksesan data atau pencarian data dengan menggunakan *query* atau *join* pada *database* perlu memperhatikan ketepatan implementasi dari data itu sendiri serta waktu prosesnya. Ada banyak cara untuk menemukan sebuah *query*. Walau banyaknya *query* yang dapat digunakan dalam sebuah kasus atau soal dalam DBMS, misalnya *query* yang digunakan untuk pencarian data, akan menghasilkan *output* yang sama, namun dengan proses yang berbeda tergantung bagaimana struktur dan banyaknya *record* dalam *database* tersebut. Dari semua *query* yang dapat digunakan, khususnya *query* yang digunakan untuk pencarian data, sangat diperhatikan waktu kecepatan akses yang diperlukan untuk mencari sebuah data, *query* mana saja yang memiliki akses lebih cepat untuk mendapatkan sebuah nilai, khususnya untuk *record* data yang besar, dibutuhkan sebuah perbandingan. [3] SMK VIP Purworejo saat ini menggunakan sistem informasi penjadwalan berbasis *web*. Jumlah mata pelajaran yang banyak dan sering berubah sewaktu-waktu sering menimbulkan masalah. Banyaknya jumlah mata pelajaran dan guru yang mengajar maka akan semakin membutuhkan waktu yang lama untuk mengetahui mata pelajaran yang diampu oleh guru tersebut dan untuk mengetahui jam mengajar. Dari permasalahan tersebut terlihat bahwa SMK VIP Purworejo membutuhkan optimasi *query* dalam penyusunan struktur *database* agar sintak yang digunakan lebih sedikit tetapi memiliki performa yang cepat dalam menampilkan *output*.

Tinjauan pustaka dari penelitian ini adalah Basis data merupakan salah satu unsur penting dalam perancangan suatu sistem, pada penelitian Dhaval Palel [2015] yang berjudul

"An Approach for Query Optimization by using Schema Object Base View" menyatakan bahwa untuk mengoptimalkan sebuah *query* SQL dapat dilakukan dengan pendekatan objek berbasis *view* [4]. Hasil dari penelitian ini adalah membandingkan kinerja antara *view* dan *non view* baik dari sisi CPU *cost* dan *input-output cost* menggunakan skema objek.

Pendekatan *incremental* menyebabkan beberapa anomali, pada penelitian Abdulaziz S. Almayzad [2010] yang berjudul "incremental view maintenance : an algorithmic approach" menyatakan bahwa untuk menjaga konsistensi dan konvergensi *database* dapat dilakukan dengan konsep *version store* [5]. Hasil penelitian ini adalah untuk mendeteksi pesan pemberitahuan data yang hilang saat memperbaharui *view*.

Data warehousing (DW) merupakan ciri teknologi saat ini, pada penelitian B. Ashadevi [2010] yang berjudul "a framework for the view selection problem in data warehousing environment" menyatakan bahwa ruang dan biaya sering menjadi masalah dalam sebuah gudang data [6]. Hasil penelitian ini adalah mengoptimalkan Biaya pemeliharaan, penyimpanan dan pemrosesan *query* dengan menggunakan *view*.

Tujuan dari penelitian ini adalah untuk mengetahui efisiensi sintak dan performa kecepatan dengan menggunakan *query join* dan *view*.

II. METODOLOGI PENELITIAN

Selain melakukan pendekatan mengenai informasi tentang SMK VIP Purworejo penelitian ini berbentuk studi kasus dan teknik pengumpulan data menggunakan observasi, wawancara, Identifikasi masalah, studi pustaka, kemudian melakukan tahapan perancangan yaitu menginstal aplikasi-aplikasi pendukung dan merancang *database* selanjutnya adalah melakukan menganalisis hasil perbandingan eksperimen.

1. Observasi

Penulis mengumpulkan informasi mengenai sistem informasi penjadwalan yang digunakan oleh SMK VIP Purworejo

2. Wawancara

Pada tahap ini peneliti melakukan wawancara ke SMK VIP Purworejo untuk mencari informasi mengenai sistem

yang sedang berjalan disana, untuk mengetahui permasalahan yang ada.

3. Identifikasi masalah

Penulis melakukan analisis hasil wawancara mengenai kendala dan permasalahan mengenai sistem yang sedang berjalan di VIP Purworejo Berdasarkan informasi yang telah diperoleh selanjutnya peneliti menentukan tujuan penelitian yaitu Optimasi *query* pada sistem informasi penjadwalan mata pelajaran sekolah menggunakan *view*.

4. Studi Pustaka

Penulis mencari referensi penelitian terdahulu yang berkaitan dengan tujuan penelitian untuk mengetahui kontribusi penelitian. Penulis juga mencari informasi yang dapat membantu penelitian melalui beberapa referensi. Materi referensi diambil dari buku, disertasi, jurnal, skripsi, makalah, dan tesis.

5. Perancangan

Pada tahap perancangan peneliti membuat rancangan *database* terlebih dahulu untuk memasukan optimasi *query* yang akan dibuat,

6. Analisis hasil perbandingan eksperimen

Pada tahap ini peneliti melakukan perbandingan terhadap hasil eksperimen yang telah dibuat. Skema urutan tahapan-tahapan penelitian yang akan ditempuh dapat dilihat pada Gambar 1



Gambar 1. Alur penelitian

hasil perbandingan dari penggunaan *view* dan tanpa penggunaan *view*.

Pengujian pertama yang akan dilakukan adalah pengujian untuk menentukan performa kecepatan *join* dengan penggunaan *view* dan tanpa *view* untuk menampilkan data jadwal yang berelasi dengan data guru, mapel, kelas, tahun_akademik, hari dan waktu.

Sebelum melakukan pengujian untuk dalam menampilkan laporan penjadwalan terlebih dahulu dibuat tabel *view* dengan *query join* sebagai berikut.

```
create view v_jadwal as
select g.*, k.*, m.*, a.*, h.*, w.*
from jadwal j
join kelas k on j.id_kelas=k.id_kelas
      join mapel m on j.id_mapel=m.id_mapel

join hari h on j.id_hari=h.id_hari
join waktu w on j.id_waktu=w.id_waktu
join guru g on j.id_guru=g.id_guru
join tahun_akademik a
on a.id_tahun_akademik=j.id_tahun_akademik
```

Selanjutnya dilakukan pengujian performa kecepatan pemanggilan data pada penggunaan *join* dan *view* pada *query 1* dan *query 2* sebagai berikut:

query 1 merupakan sintak yang berisi perintah untuk menampilkan semua data jadwal pelajaran yang ada didalam database penjadwalan baik itu menggunakan perintah *join* dan *view*.

Query 1 : menggunakan *join*

```
Function lihat(){
    $this->db->select ('g.*,k.*,m.*,a.*,h.*,w.*')
        ->from ('jadwal as j')
        ->join ('kelas k',
'j.id_kelas=k.id_kelas')
        ->join ('mapel m',
'j.id_mapel=m.id_mapel')
        ->join('tahun_akademik
a',j.id_tahun_akademik
=a.id_tahun_akademik')
        ->join ('hari
h',j.id_hari=h.id_hari')
        ->join ('guru
g',j.id_guru=g.id_guru')
        ->join ('waktu
w',j.id_waktu=w.id_waktu');
    $query=$this->db->get();
    return $query;
}
```

Query 1 : menggunakan *view*

```
function lihat(){
    return $this->db->get(get('v_jadwal'));
}
```

Query 2 merupakan sintak yang berisi perintah untuk menampilkan semua jadwal setiap kelas yang ada didalam database penjadwalan baik itu menggunakan perintah *join* dan *view*.

Query 2 : menggunakan *join*

```
function jadwal_siswa($id){
    $this->db->select ('g.*,k.*,m.*,a.*,h.*,w.*')
        ->from ('jadwal as j')
        ->join ('kelas
k',j.id_kelas=k.id_kelas')
        ->join ('mapel
m',j.id_mapel=m.id_mapel')
        ->join ('tahun_akademik
a',j.id_tahun_akademik =a.id_tahun_akademik')
        ->join ('hari
h',j.id_hari=h.id_hari')
        ->join ('guru
g',j.id_guru=g.id_guru')
        ->join ('waktu
w',j.id_waktu=w.id_waktu');
    $this->db->where('k.id_kelas',$id);
    $query = $this->db->get();
    return $query;
}
```

Query 2 : menggunakan *view*

```
function jadwal_siswa($id){
    $this->db->from('v_jadwal');
    $this->db->where('id_kelas',$id);
    $query = $this->db->get();
}
```

IV. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, maka dapat diambil beberapa kesimpulan antara lain :

1. Penggunaan *view* untuk menampilkan semua data jadwal pelajaran dan menampilkan data jadwal pelajaran berdasarkan kelas memiliki penulisan sintak yang lebih sedikit daripada menggunakan *join*.
2. Penggunaan *view* untuk menampilkan semua data jadwal pelajaran dan

menampilkan data jadwal pelajaran berdasarkan kelas memiliki waktu yang lebih cepat daripada menggunakan *join*.

REFERENSI

- [1] Suhartati dan Yeyen Dwi Atma, "Optimasi Query Sederhana Guna Kecepatan Query Pada Database Server," *METIK Jurnal*, vol.1 no.1, 2017
- [2] Gat, "Perancangan Basis Data Perpustakaan Sekolah Dengan Menerapkan Model Data Relasional," *Citec Journal*, vol2 no.4, 2015
- [3] Warnars, Spit, "Perbandingan Penggunaan database OLTP Dan Data Warehouse", *CCIT Journal*, vol8 no.1, 2014
- [4] Wijaya, Antonius Bima Murti, Irya Wisnubhadra dan Benyamin L. Sinaga, "Optimalisasi Algoritma Insert Memanfaatkan Memory Primer dan Bulk Insert Studi Kasus : Pengembangan Sistem Pengolahan Data Perpajakan PNS", *Seminar Nasional Informatika 2012 (semnasIF 2012)*, 30 Juni 2012
- [5] Kumar, K. Kiran, T.M. Santhi Sri dan Voruganti Vamshi Priya, "Introduction to Techniques of Query Processing and Optimization," *International Journal Of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2 no.3, 2015
- [6] Hariyanto, Bambang, Sistem Manajemen Basis Data, Bandung:Penerbit Informatika, 2004
- [7] Habimana, Jean, "Query Optimization Techniques-Tips For Writing Efficient And faster SQL Queries," *International Journal Of Scientific & Technology Research*, vol.4 no.10, 2015
- [8] Mata-Toledo, Ramon A. dan Pauline K.Cushman, "Dasar-Dasar Database Relasional," Surabaya:Penerbit Erlangga, 2004
- [9] Dewi, Melany Mustika dan Novandi Henderi, "Analisis Perbandingan Optimasi Query Nasted Join dan hash Join Pada MySQL Server," *CSRID Journal*, vol.9 No.1, Februari 2017
- [10] Alnawaj'ha, Fawwaz Yousef, "The Performance Of Inner Join Types in SQL," *International Journal Of Advanced Studies In Computer Science And Engineering (IJASCSE)*, vol.5 no.11,2016.