OPTIMASI QUERY PADA DATABASE UNTUK 2-WAY SMS DIPENDA PROVINSI SULAWESI UTARA

Yohakim Benedictus Samponu[1)] dan Ronny Faslah[1]
[1, 2)]S2 Teknik Informatika Universitas AMIKOM Yogyakarta
e-mail: mikahoy045@gmail.com[1)], ronnyfaslah@gmail.com[2)]

ABSTRAK

Optimasi query pada database memegang peranan yang cukup penting baik dalam segi kenyamanan pengguna (user experience) maupun dalam performa suatu sistem. Optimasi dari suatu database biasanya dilakukan dengan melakukan normalisasi dari struktur database yang digunakan dan query yang di eksekusi untuk melakukan pengambilan data dari tabel database. Pentingnya optimasi pada database merupakan latar belakang dasar adanya penelitian ini. Penelitian ini dilakukan untuk lebih memperjelas pengaruh optimasi, khususnya optimasi query pada jenis database MariaDB / MySQL dari sistem pembayaran pajak kendaraan bermotor di Dinas Pendapatan Daerah (DIPENDA) Provinsi Sulawesi Utara terutama untuk bagian sms interaktif (2-way sms) yang pastinya membedakan penelitian ini dengan penelitian serupa yang pernah ada yang tentunya dijadikan sebagai bahan referensi dalam melakukan penelitian ini. Beberapa teori terkait tentang query yang akan dioptimasi ini juga akan dibahas di tulisan ini sebagai acuan dalam melakukan penelitian. Penulis melakukan serangkaian skenario percobaan untuk melakukan perbandingan performa eksekusi query dimana hasilnya dapat dilihat dan akan dibahas lebih detail di tulisan ini. Hasil dari pembahasan optimasi query pada database ini diharapkan mampu menjadi pembelajaran untuk melakukan perbaikan sistem selanjutnya di DIPENDA Prov. Sulut agar performa pelayanan sistem dapat ditingkatkan yang secara tidak langsung meningkatkan layanan untuk user dalam hal ini masyarakat.

Kata Kunci: Database, Optimasi, Query, MariaDB, MySQ

PENDAHULUAN

Seperti yang diketahui hampir seluruh sistem yang berjalan membutuhkan *storage* sebagai gudang penyimpanan data, tidak terbatas untuk sistem khusus. Pada umumnya semakin besar suatu perusahaan atau instansi pemerintah yang menjalankan seluruh proses bisnisnya menggunakan sistem, maka semakin besar pula data yang dimiliki. Proses transaksi data / interaksi antara *user* dan *storage* juga akan semakin intensif sehingga jika tidak di optimasi akan berdampak pada performa penggunaan sistem.

Database merupakan salah satu *storage* yang merupakan sebuah keharusan pada hampir semua sistem yang berjalan saat ini yang karena keistimewaannya jika dibandingkan dengan penggunaan *file* sebagai *storage* untuk aplikasi tunggal. *Database* lebih efisien digunakan sebagai tempat penyimpanan data karena memiliki kemampuan keterkaitan antara data dan lebih mudah untuk dikembangkan. Untuk meningkatkan performa penggunaan sistem pada saat melakukan transaksi data antara *user* dan *storage* data yang dalam hal ini menggunakan *database*, maka dikembangkanlah teknik optimasi. Optimasi *database* merupakan tugas umum dari *Database* Administrator dan memegang peranan penting dalam

meningkatkan performa suatu sistem informasi. Proses optimasi bisa dilakukan dengan 2 cara yaitu dengan melakukan normalisasi *database* dimana seluruh struktur tabel diatur agar tidak terjadi *redundansi* yang malah akan menghambat performa dari sistem itu sendiri.

Dalam penelitian ini akan ditekankan bagian Optimasi *query* dimana proses Optimasi dilakukan dengan cara mengoptimalkan *query* yang digunakan untuk mengakses *database* pada sistem Layanan Pesan Singkat (*sms*) interaktif yang ada pada Dinas Pendapatan Daerah (DIPENDA) Provinsi Sulawesi Utara.

Penelitian ini dilakukan untuk mengembangkan optimasi *query* yang ada pada *database* yang menangani *sms* interaktif (2-way *sms*) di sistem informasi pembayaran kendaraan bermotor DIPENDA Prov. Sulut. Beberapa penelitian sebelumnya telah melakukan beberapa percobaan optimasi *query* secara umum, namun di penelitian ini akan dibahas khusus tentang optimasi query untuk studi kasus sistem yang sudah berjalan, dan penyesuaian *query* optimasi untuk sistem *backend* 2-way *sms*.

Penggunaan studi kasus pada penelitian ini dikarenakan optimasi *query* yang berbeda untuk setiap maksud penggunaannya, dari beberapa referensi yang ada terdapat optimasi query yang lebih baik daripada query yang dieksekusi sebelumnya, tapi tidak akan bisa di implementasikan secara umum karena fungsi query nya yang berbeda.

Selain itu optimasi *query* ini juga lebih berfokus untuk *Database Management System* (DBMS) MariaDB / MySQL dimana *database* tersebut digunakan pada studi kasus yang akan kita bahas kali ini.

Dalam penelitian ini untuk menyesuaikan dengan studi kasus maka penulis menggunakan 2 (dua) dari 5 (lima) notasi aljabar *relasional* dasar yaitu seleksi (*selection*) dan proyeksi (*projection*).

A. Projection

Projection biasa digunakan untuk memilih atau menampilkan atribut / *field* dari suatu tabel atau relasi.Pada DBMS SQL MariaDB / MySQL yang merupakan notasi aljabar *relasional* dasar *projection* adalah :

WHERE condition1
GROUP BY grouping HAVING

B. Selection

Projection biasa digunakan untuk memilih atau menampilkan atribut / *field* dari suatu tabel atau relasi pada DBMS SQL MariaDB / MySQL yang merupakan notasi aljabar *relasional* dasar *projection* adalah :

SELECT field1, field2 FROM TABEL.

METODE PENELITIAN

Percobaan dilakukan dengan membuat sampel contoh *database* yang mirip dengan *database* Sistem Informasi Pembayaran Pajak Kendaraan Bermotor di Dinas Pendapatan Daerah Provinsi Sulawesi Utara yang berfungsi untuk melakukan fungsi 2-way *sms*.

A. Pembuatan Tabel

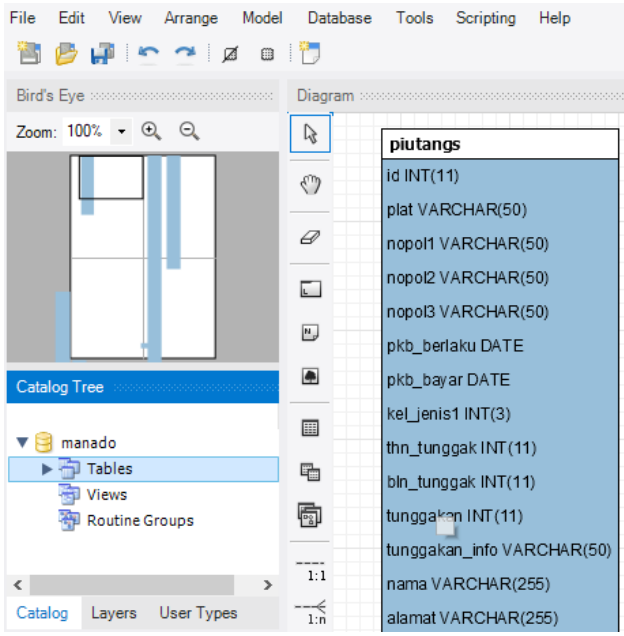
Pertama akan dilakukan pembuatan tabel pada *database* yang *field* tabelnya mirip dengan *database* Sistem Informasi Pembayaran Pajak Kendaraan Bermotor di Dinas Pendapatan Daerah Provinsi Sulawesi Utara. Pembuatan tabel dibuat dengan sedikit modifikasi karena tidak bisa untuk di publikasikan secara umum.

Database dibuat dengan menggunakan MariaDB yang struktur dan sintaknya sama dengan MySQL. Dimana DBMS yang digunakan ini sama dengan DBMS yang di gunakan di DIPENDA

Provinsi Sulawesi Utara. Adapun tabel-tabel yang dibutuhkan dalam pembuatan sistem ini :

1) Tabel Piutangs

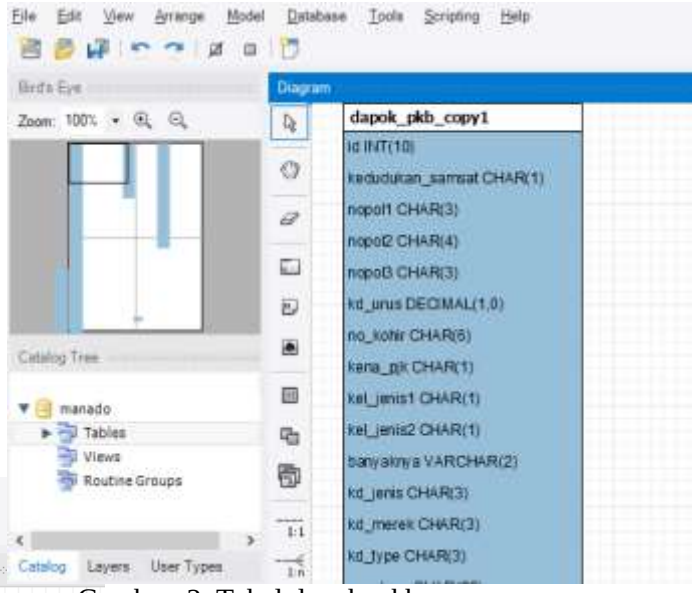
Tabel piutang digunakan untuk melakukan kalkulasi piutang pajak kendaraan bermotor yang dibutuhkan untuk memberikan informasi kepada wajib pajak pengguna kendaraan bermotor yang meminta informasi melalui modul 2-way *sms*. *Field* tabel tidak dapat ditampilkan seluruhnya karena tidak dapat dipublikasikan secara umum.



Gambar. 1. Tabel piutangs

2) Tabel dapok_pkb

Tabel dapok_pkb digunakan untuk menyimpan data kendaraan bermotor termasuk jatuh tempo pembayaran, dan rangkuman total pembayaran untuk memberikan informasi kepada wajib pajak pengguna kendaraan bermotor yang meminta informasi melalui modul 2-way *sms*. *Field* tabel tidak dapat ditampilkan seluruhnya karena tidak dapat dipublikasikan secara umum.



Gambar. 2. Tabel dapok_pkb

Salah satu masalah dalam implementasi Sistem Informasi Pembayaran Kendaraan Bermotor di Dinas Pendapatan Daerah Provinsi Sulawesi Utara adalah tidak adanya normalisasi pada tabel *database* selain itu tidak terdapat relasi antara *Primary Key* dan *Foreign key* pada semua tabel yang ada di *database* sehingga terjadi *redundansi* data yang mengakibatkan pemborosan ruang di *database*, *redundansi* data dapat mengakibatkan terjadinya *inconsistency* data pada *database*.

B. Optimasi Query

Optimasi *query* dilakukan dengan membandingkan kecepatan *query* data dari aplikasi yang digunakan dengan *query* data optimasi. Aplikasi 2-way *sms* dibuat dengan bahasa pemrograman PHP dengan memanfaatkan Framework Laravel 5.2.

Adapun salah satu query yang digunakan dalam aplikasi tersebut untuk mengambil data dari tabel dapok_pkb adalah :

```
 $dapok_kb = Pkb::on($on)-
>where('nopol2', $n1)-
>orwhere('nopol2', $n2)-
>orwhere('nopol2', $n3)->get();
```

Yang dalam mariaDB dapat ditulis :

*SELECT * FROM dapok_pkb WHERE nopol2 = $n1 OR nopol2 = $n2 OR nopol2 = $n3;*

Dapat dioptimasi menjadi :

*SELECT * FROM dapok_pkb WHERE nopol2 IN ($n1,$n2,$n3);*

Selain itu pula dapat dipilih kolom tertentu untuk ditampilkan, dengan cara mengubah *selection*. Misalnya untuk menampilkan *field* nama_pemilik, alamat_pemilik, thn_berlaku_pkb :

SELECT nama_pemilik, alamat_pemilik, thn_berlaku_pkb FROM dapok_pkb WHERE nopol2 IN ($n1,$n2,$n3);

Sehingga *query* solusi pada aplikasi dengan menggunakan *Framework Laravel* menjadi :

```
$dapok_kb = Pkb::on($on)-
>wherein('nopol2', array($n1, $n2,
$n3))->select('nama_pemilik',
'alamat_pemilik',
'thn_berlaku_pkb')->get();
```

Sedangkan salah satu *query* yang digunakan dalam aplikasi untuk mengambil data dari tabel piutangs adalah seperti di bawah :

```
$piutang = Piutang::on($on)-
>where('nopol2', $n1)-
>orwhere('nopol2', $n2)-
>orwhere('nopol2', $n3)->get();
```

Yang dalam mariaDB dapat ditulis :

*SELECT * FROM piutangs WHERE nopol2 = $n1 OR nopol2 = $n2 OR nopol2 = $n3;*

Dapat dioptimasi menjadi :

*SELECT * FROM piutangs WHERE nopol2 IN ($n1,$n2,$n3);*

Selain itu pula dapat dipilih kolom tertentu untuk ditampilkan, dengan cara mengubah *selection*. Misalnya untuk menampilkan pkb_bayar, tunggakan :

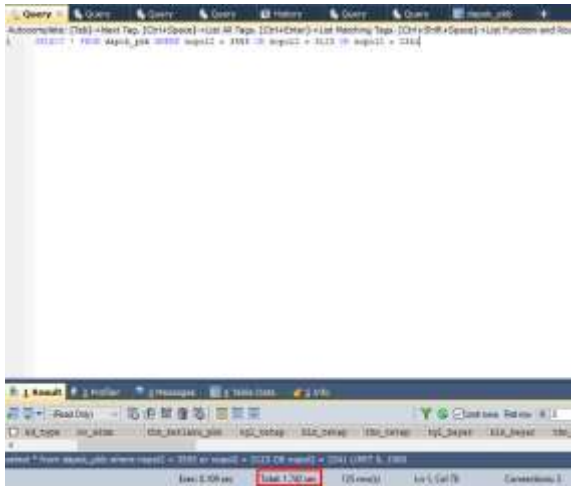
SELECT pkb_bayar, tunggakan FROM piutangs WHERE nopol2 IN ($n1,$n2,$n3);

Sehingga *query* solusi pada aplikasi dengan menggunakan Framework Laravel menjadi :

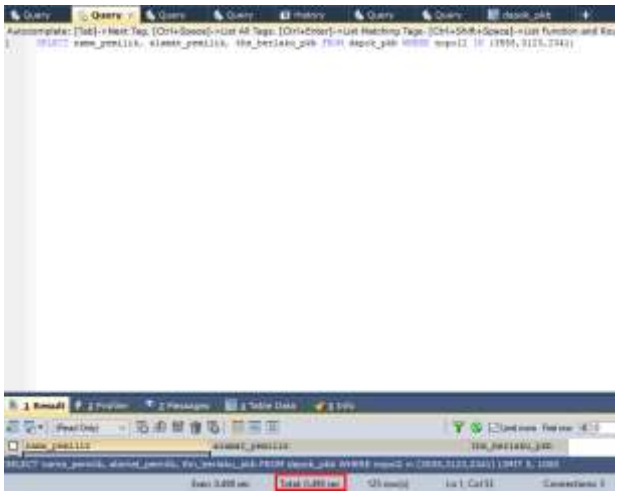
```
$piutang = Piutang::on($on)-
>wherein('nopol2', array($n1, $n2,
$n3))->select('pkb_bayar',
'tunggakan')->get();
```

HASIL DAN PEMBAHASAN

Dari percobaan optimasi *query* di atas maka dihasilkan perbedaan antara waktu eksekusi sebelum menggunakan optimasi untuk tabel dapok_pkb sebagai berikut :



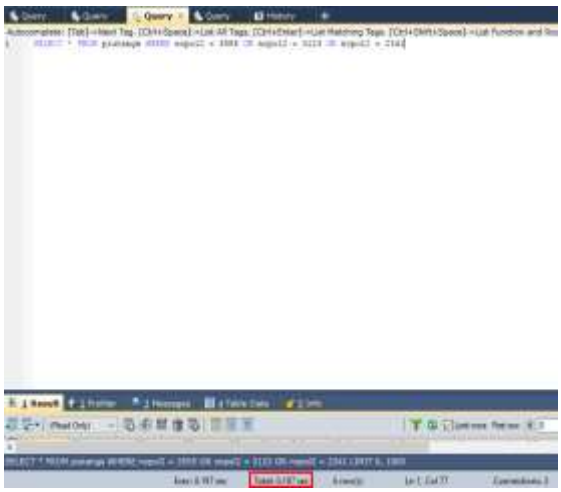
Gambar. 3. Eksekusi *query* dapok_pkb sebelum optimasi



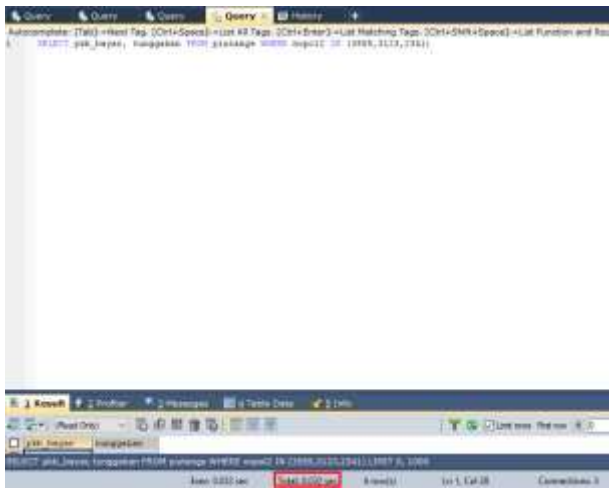
Gambar. 4. Eksekusi *query* dapok_pkb setelah optimasi

Dari gambar di atas terlihat *query* yang sudah dioptimasi lebih cepat di eksekusi (0,498 *sec*) dibandingkan sebelum dioptimasi (1,742 sec) dari 292910 *record*.

Dari percobaan optimasi *query* di atas maka dihasilkan perbedaan antara waktu eksekusi sebelum menggunakan optimasi untuk tabel piutang sebagai berikut :



Gambar. 5. Eksekusi *query* piutang sebelum optimasi



Gambar. 6. Eksekusi *query* piutang setelah optimasi

Dari gambar di atas terlihat *query* yang sudah dioptimasi lebih cepat di eksekusi (0,032 *sec*) dibandingkan sebelum dioptimasi (0,197 sec) dari 59523 *record*.

Dari hasil optimasi *query* dapat dilihat perbedaan yang signifikan antara *query* sebelum optimasi dengan *query* sesudah optimasi, karena dengan cara melakukan *selection* terhadap *field* yang dibutuhkan maka ukuran dari tabel hasil *query* akan mengecil, sehingga mengurangi waktu transaksi antara aplikasi dan *database* sehingga meningkatkan performa dari *query* tersebut. Selain itu juga penggunaan IN dapat meningkatkan kecepatan pengambilan indeks karena hanya akan mencocokkan data dari daftar indeks dengan daftar *query* IN sehingga pengambilan / pencarian data lebih efisien.

KESIMPULAN

Masih banyak kekurangan yang masih ditemukan dalam pengembangan Sistem Informasi Kendaraan Bermotor DIPENDA Provinsi Sulawesi Utara, secara keseluruhan *database* pada aplikasi ini harus di normalisasi dan diatur kembali relasi antara tabel satu dan tabel yang lainnya agar tidak terjadi *redundansi* data dan inkonsistensi data. Sedangkan pada

bagian 2-way *sms* perlu dilakukan optimasi *query* untuk meningkatkan performa / kecepatan respons dari aplikasi ini, mengingat sangat dibutuhkannya kecepatan, ketepatan, dan kemampuan dalam menangani data yang besar dalam waktu singkat.

DAFTAR PUSTAKA

Fuhr N, Rolleke T. 1997. *A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems*. ACM Transactions on Information Systems. Vol. 15 (1): 32-66

Habimana, J. 2015. *Query Optimization Techniques - Tips For Writing Efficient and Faster SQL Queries.* INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH Vol 4 (10) : 22-23.

Silberschatz A, Korth HF, Sudarshan S. 2010. *Database System Concept. In. Internasional 4th Edition: McGraw-Hill;* 140-260.

Whitten, J. L., Bentley, L. D., & Dittman, K. C. 2004. *Metode Desain dan Analisis Sistem Edisi 6*. Yogyakarta: Mc Graw Hill Education. 517-540

.