Perancangan Keamanan Pengambilan Kembali Data Menggunakan Enkripsi Simetris pada Struktur ORDBMS

M. Fairul Filza
Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta
Jl. Ring Road Utara, Ngringin, Condongcatur, Kec. Depok, Kabupaten Sleman, Daerah Istimewa Yogyakarta 55281
fairul.f@amikom.ac.id[1]

INTISARI

Teknologi basis data adalah komponen inti dari banyak sistem komputasi. Basis data memungkinkan data yang akan disimpan dan berbagi secara elektronik. Begitu pula kebutuhan untuk memastikan integritas data dan keamanan data dari akses yang tidak diinginkan. Keamanan basis data dibuktikan dengan peningkatan jumlah kekhawatiran dan insiden kehilangan atau pelanggaran dilaporkan terhadap data yang sensitif. Penelitian ini membahas secara teknis pengamanan dalam pengambilan kembali data pada lapisan datasource dengan enkripsi simetris. Hasil dari penilitian ini adalah pembuatan virtual dan mengkamuflase tabel sehingga ketika diakses oleh penguna publik akan menampilkan rekaman data yang terenkripsi.

ABSTRACT

Data base technology is a core component of many computing systems. The database allows data to be stored and shared electronically. Similarly, the need to ensure data integrity and data security from unwanted access. Database security is evidenced by the increasing number of concerns and incidents of loss or breach is reported against sensitive data. This study discusses the technical safeguards in taking back the data on the layer of the datasource with symmetric encryption. The result of this was the creation of a virtual and camouflage tables when accessed by users so that the public will display the encrypted data records.

Keyword *— Cryptography, Data Security, Aes, Postgesql, Python, RDBMS*

I. PENDAHULUAN

Teknologi basis data adalah komponen inti dari banyak sistem komputasi. Basis data memungkinkan data yang akan disimpan dan berbagi secara elektronik dan jumlah data yang terkandung dalam sistem ini terus tumbuh pada rata-rata eksponensial. Begitu pula kebutuhan untuk memastikan integritas data dan keamanan data dari akses yang tidak diinginkan. The Privacy Rights Clearing House (2010) melaporkan bahwa lebih dari 345 juta catatan pelanggan telah hilang atau dicuri sejak tahun 2005 terhitung dimulainya pelacakan data insiden pelanggaran [1].

Basis data menjadi favorit target bagi para penyerang, dikarenakan basis data menyimpan data yang penting dan konfidensial [2]. Dalam beberapa hal informasi-informasi yang terkandung didalam sebuah basis data terkadang mengandung informasi penting dan rahasia. Informasi yang penting menjadi sebuah resiko yang harus diperhatikan apabila orang yang tidak mempunyai hak mengetahui isinya. Keamanan informasi menjadi menjadi hal penting sehingga isi yang dikandung tidak diketahui oleh sembarang orang. Akibat yang muncul dari masalah-masalah tersebut adalah anggapan tentang cara membuat perlindungan terhadap basis data dari akses-akses yang tidak memiliki hak, akses perubahan ataupun akses penghapusan.

Dalam dunia kriptografi ternyata huruf yang sama pada pesan mempunyai image huruf yang sama juga. Hal ini mempunyai tingkat resiko yang tinggi karena mudah ditebak. Unuk menyelesaikan hal ini maka pesan haruslah disandikan (encoding). Tujuan membuat penyandian adalah agar aman dari para pembongkar sandi sehingga hanya penerima saja yang mengetahui isinya [3]

Berdasarkan latar belakang diatas, maka masalah yang muncul yaitu: Bagaimana cara menyamarkan tabel dengan melewati bidang keilmuan kriptografi tanpa mengurangi waktu perencanaan dan ekseskusi pada basis data?

Tujuan dari penelitian yang dilakukan adalah sebagai berikut: Menyamarkan tabel kedalam bentuk yang sudah dienkripsi sehingga tidak dapat dibaca dengan mudah,

dan Meningkatkan waktu eksekusi saat mengembalikan data.

Pada paper ini memaparkan tinjauan pustaka pada bagian 2. Pada bagian 3 akan dibahas metodologi penelitian. Diikuti pembahasan pada bagian 4. Dan akhir dari paper ini adalah kesimpulan pada bagian terakhir.

II. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode eksperimental. Sebelum memulai program penelitian, penting harus dilakukan tahapan perancangan terhadap program yang akan dikerjakan. Tahapan ini akan dibagi menjadi tiga tahapan, antara lain: perancangan struktur skema, dan pengujian pada hasil yang akan dikerjakan.

Sebelum memulai program penelitian, penting harus dilakukan tahapan perancangan terhadap program yang akan dikerjakan. Tahapan ini akan dibagi menjadi tiga tahapan, antara lain: perancangan struktur skema, dan pengujian pada hasil yang telah selesai dikerjakan.

Data yang akan digunakan adalah data dummy karyawan yang diperoleh pada suatu perusahaan. Atribut pada data juga sudah disederhanakan menjadi lima atribut. Sebelum digunakan, data terlebih dahulu dimigrasikan kedalam Postgres.

Rangkaian percobaan diuji dengan menguji kebenaran data terenkripsi dengan baik. Indikasi kebarhasilan adalah data yang tampil, akan terenkripsi dan sulit untuk dibaca oleh manusia normal. Rangkaian percobaan juga diuji beberapa kali dan dicatat waktu perencanaan dan eksekusinya. Kemudian dibandingkan antara tabel asli (direct table), view, dan materialized view dalam kondisi dan data yang sama.

III. HASIL DAN PEMBAHASAN

Percobaan dilakukan dengan tahapan yang sesuai dengan yang sudah disebutkan dalam metode penelitian. Melewati tahapan antara lain: perancangan struktur skema, dan pengujian pada hasil yang telah selesai dikerjakan. Berikut skema struktur basisdata yang akan direplika kedalam PostgreSQL.

	Atribut	Tipedata	Panjang	Status
2	EmpId	char	4	Not Null
3	Name	varchar	50	Not Null
4	Gender	char	1	Not Null
5	UserAcc	varchar	30	Not Null
6	UserKey	varchar	255	Not Null

Gambar 1. Struktur Tabel Karyawan

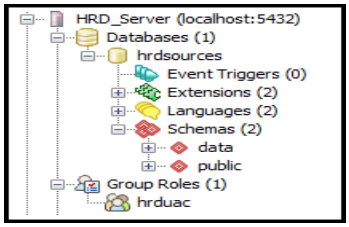
Berdasarkan skema tabel diatas maka akan dirancang sebuah skema untuk membuat penyamaran dari tabel dengan tampilan yang sudah di enkripsi.

Program yang sudah dianalisa selanjutnya akan melewati tahap perancangan. Dengan adanya perancangan diharapkan ketika pembuatan program akan menjadi lebih mudah. Proses perancangan akan meliputi struktur skema dan pengguna. Basis data yang ada pertama kali akan dibagi menjadi skema-skema yang terpisah satu dengan yang lain.

Skema yang pertama adalah skema umum (public) yang sudah tersedia ketika basisdata pertama kali diciptakan. Skema yang kedua adalah skema khusus yang diciptakan untuk super admin yang memiliki akun pengguna dengan level akses yang berbeda.

Untuk skema umum hanya diijinkan atas perintah “INSERT” dan “SELECT” khusus tabel karyawan yang sudah disamarkan. Untuk skema khusus semua perintah terhadap basis data (“INSERT”, “SELECT”, “UPDATE”, “DELETE”, “REFRESH”) akan diijinkan.

Skema khusus dinamai “data”. Pada nanti semua tabel, fungsi akan disimpan didalam skema tersebut. Dan hanya bisa diakses oleh admin yang memiliki hak akses. Untuk pengguna dengan level umum (public) hanya diperbolehkan untuk menambahkan rekaman data (insert) dan pengambilan kembali data (select) pada view yang sudah dienkripsi.



Gambar 2. Skema khusus yang sudah dibuat

Pada program penelitian ini fungsi enkripsi akan menggunakan bahasa python yang nantinya akan dikemas dengan bahasa prosedural dari PostgreSQL. Fungsi yang dibuat pada bahasa python akan dibagi menjadi dua proses utama, yaitu: TextEncryptor() dan CryptoProcessing(). Masing-masing fungsi mempunyai peranan yang berbeda.

Fungsi CryptoProcessing() adalah fungsi yang dibuat sebagai dasar dari pusat enkripsi yang menerapkan algoritma Rijndael. Fungsi ini dibangun dengan sebuah kelas dimana didalam kelas tersebut terdapat fungsi-fungsi utama dari kriptograpi.

Untuk meningkatkan performa dalam pengambilan kembali data, maka data dikemas dalam bentuk materialized view. Sebab dalam mengambil kembali data, terdapat proses subquery dan pemanggilan fungsi enkripsi dimana memiliki unit logika yang komplek dan rumit. Materialized view memiliki keunggulan pada performa, namun mempunyai kelemahan dalam perubahan data yang bersifat terus menerus (real-time).

Proses enkripsi data akan dilakukan pada materialized view. Hal ini akan membuat pengguna umum (public-user) ketika melakukan query select pada tabel karyawan hanya menampilkan tabel yang sudah disamarkan dengan rekaman data yang telah dienkripsi. Berikut query yang rancang sebagai materialized view untuk proses pengambilan kembali data. Fungsi enkripsi yang telah dibahas, akan dikemas kedalam fungsi prosedural pada PostgreSQL. Dan fungsi-fungsi tersebut akan dieksekusi pada materialized view.

Perlu ditekankan proses enkripsi bukan terjadi pada data fisik. Dalam artian rekaman data yang asli yang tersimpan di media penyimpanan (HDD) tidak terenkripsi. Sehingga jika terjadi kerusakan logic pada sistem operasi atau dilakukannya migrasi data, rekaman data masih dapat dibaca dan dipahami. Proses enkripsi yang terjadi ketika pengambilan kembali data dilakukan oleh pengguna umum yang tidak memiliki level akses super admin.

Untuk tahapan uji coba akan melewati beberapa tahapan diantaranya: Pengujian pada hasil enkripsi terhadap tabel. Pengujian terhadap fungsi deskripsi data tunggal. Dan pengujian terhadap performa enkripsi.

Tahapan akan menguji data dari tabel sebelum di enkripsi dan hasil setelah dienkripsi. Data yang dilihat ketika melakukan eksekusi dengan query "SELECT * FROM employee" akan dialihkan ke dalam skema data. Hal ini akan menyembunyikan tabel asli. Berikut tampilan dari tabel asli:

	empid character(4)	name character varying(50)	gender character(1)	useracc character varying(30)	userkey character varying(255)
1	0101	Fai Filza	L	fai	fai234
2	0001	Karsono	L	arso	kar123
3	0002	Endang Susinurwati	P	sinur	end234
4	0003	Narwoto	L	arwe	nar123
5	0004	Ashar Agus Hidayat	L	ash	ash456

Gambar 3. Bentuk dari tabel asli

Setelah itu dibuatkan sebuah Materialized view yang diletakkan kedalam skema public untuk mengelabui tabel yang asli. View ini sudah dienkripsi. Sehingga dengan dieksekusi query "SELECT * FROM employee" maka akan dihasilkan tabel seperti gambar dibawah ini:

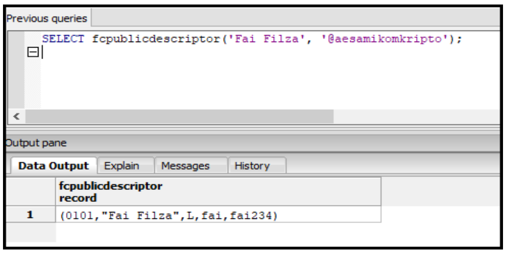
	empid text	name character varying(50)	gender character(1)	useracc text	userkey text
1	□[æDßeT\)õqÐò	Fai Filza	L	8«°□bæ]924-ë	ðÑì□úÐÆ÷ □□Ý´°
2	ó£]Ô0ºü©É^]h	Karsono	L	Éå''C$□□ûÖ-□ßø	□W
3	□À¢}FÎ □üëá(§a	Endang Susinurwati	P	(¯ÊùÖp-hÕ®úÒp¯	vçz□¢(b³ou7iÂÂ
4	□;ÍÔçsyc	Narwoto	L	hKJ®□□iM/Ý*W9¬õ	ôÅ&3ïÜ`ÂWû□t□Õ"
5	□□1/	Ashar Agus Hidayat	L	5âSújàH×çâV¾ø	h□@ùonÐKm*ÖýËã{

Gambar 4. Bentuk dari tabel yang telah disamarkan

Akan terlihat rekaman data yang dianggap bernilai akan disandikan dan ditampilkan menjadi karakter yang tak terwujud dan tidak dapat dibaca (Human-Unreadable).

Tahapan selanjutnya adalah melakukan pengecekan kembali pada fungsi deskriptor untuk memastikan data yang disamarkan dapat dibaca kembali secara spesifik. Yaitu dengan menggunakan sebuah fungsi yang dibuat khusus. Fungsi ini akan meminta dua buah argumen. Yaitu sebuah teks yang diambil dari salah satu nama yang ada didata, dan sebuah kunci yang telah diketahui oleh super admin. (sejenis password).

Jika kata sandi yang tepat dimasukkan, maka akan menampilkan data asli yang tidak disandikan seperti yang tampak pada gamar dibawah ini:



Gambar 5. Fungsi public deskriptor

Hasil dari fungsi publik deskriptor adalah kumpulan data yang tidak tersandikan yang dipilih spesifik berdasakan nama karyawan. Pengujian berikut adalah pengujian yang paling penting. Sebab rangkaian pengujian ini akan menghitung seberapa dekat waktu yang diberikan oleh view terhadap tabel aslinya. Pengujian dilakukan sebanyak lima kali untuk tiga objek yaitu: Tabel, View Standar dan Materialized view. Pengujian melibatkan Planning variabel dan Execution variabel, yang merupakan variabel bawaan. Pengujian akan menggunakan fasilitas dari PosgreSQL yang bernama "Explain Select". Setelah pengujian maka akan ditarik nilai rata-rata dari setiap variabel, lalu dibandingkan dengan kurva grafik. Berikut hasil pengujian.

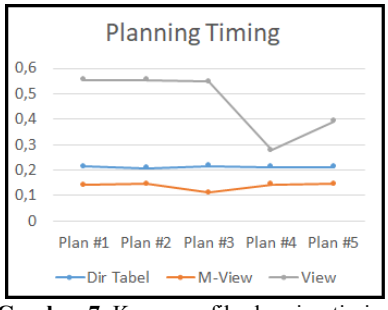
```
#1: From direct table (*ms)
Planning   : 0.215, 0.207, 0.217, 0.213, 0.211
Execution : 0.108, 0.092, 0.092, 0.091, 0.094
Result Planning  : 0.2126 ms
Result Execution : 0.0954 ms

#2: From materialized view (*ms)
Planing   : 0.142, 0.145, 0.112, 0.144, 0.145
Execution : 0.090, 0.092, 0.057, 0.094, 0.090
Result Planning  : 0.1376 ms
Result Execution : 0.0846 ms

#3: From view (*ms)
Planing   : 0.555, 0.554, 0.547, 0.278, 0.392
Execution : 1762.554, 1988.391, 1781.284, 1785.553, 1806.876
Result Planning  : 0.4652 ms
Result Execution : 1824.9316 ms
```

Gambar 6. Hasil pengujian

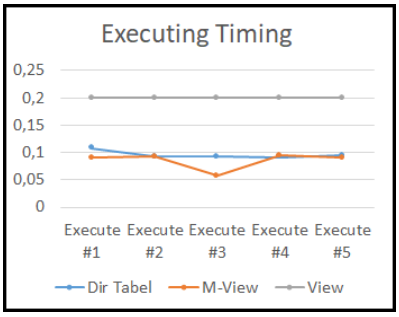
Dapat dilihat hasil dari pengujian langsung terhadap tabel adalah sebagai berikut: Planing variabel menujukan rata-rata (dalam milisecond) 0.2126 ms. Sedangkan Eksekusi variabel menunjukan rata-rata 0.0954 ms. Dengan kata lain view yang diakses harus mendekati dengan nilai rata-rata pada tabel asli.

Pada pengujian selanjutnya dilakukan terhadap standar view. Planing variabel menujukan rata-rata 0.4652 ms. Sedangkan Eksekusi variabel menunjukan rata-rata 1824.9316 ms (melebihi 1 detik). Dapat disimpulkan proses enkripsi dapat memakan waktu yang cukup berat. Maka disimpulkan view biasa akan dirasa kurang apabila dijadikan acuan dalam performa.

Pada pengujian selanjutnya dilakukan terhadap materialized view. Planing variabel menujukan rata-rata 0.1376 ms. Sedangkan Eksekusi variabel menunjukan rata-rata 0.0846 ms. Dapat disimpulkan proses enkripsi pada view ini dapat mendekati tabel asli dengan catatan selisih waktu di planing variabel adalah 0.2126 - 0.1376 sama dengan 0.075 ms lebih cepat materialized view. Dan catatan selisih waktu di eksekusi variabel adalah 0.0954 - 0.0846 sama dengan 0.108 ms lebih cepat materialized view. Untuk lebih lengkap perhatikan grafik yang dibuat.



Gambar 7. Kurva grafik planning timing



Gambar 8. Kurva grafik eksekusi timing

IV. KESIMPULAN

Kesimpulan dari penelitian ini adalah. Terwujudnya sebuah rancangan sistem keamanan pada basis data dengan cara penyandian (encoding) melewati bidang keilmuan kriptografi.

Hal ini telah dibuktikan dengan penerapan fungsi-fungsi yang ditanamkan didalam prosedur yang ada di basis data. Dimana prosedur tersebut akan melakukan serangkaian enkripsi dan dekrispsi terhadap setiap data yang ada. Data asli yang tersimpan tidak tersandikan. Proses penyandian hanya dipicu ketika pengguna ini melakukan aksi pengambilan kembali data.

Dari hasil yang telah dilakukan, maka dapat dinyatakan penyamaran tabel dengan view yang sudah disandikan memiliki waktu yang stabil bahkan lebih cepat meskipun jarak selisih angka tidak jauh. Sehingga penyamaran ini seharusnya tidak akan mengganggu performa dibandingkan dengan tabel aslinya.

UCAPAN TERIMA KASIH

Terima kasih diucapkan sebesar-besarnya kepada tim pendukung yang telah banyak membantu dalam proses penelitian. Terima kasih juga diucapkan kepada Lembaga Penelitian Amikom atas dukungan fasilitas dan pendanaan sehingga penelitian berhasil diselesaikan. Serta yang terakhir ucapa terima kasih kepada editor dan reviewer yang telah membantu menyeleksi tulisan ini menjadi lebih baik.

REFERENSI

[1] Privacy Rights Clearing House. 2010. Chronology of data breaches. http://www.privacyrights.org/, diakses 2 juli 2017.

[2] Shelly Rohilla, Pradeep Kumar Mittal, 2013, Database Security: Threats and Challenges, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5.

[3] Yuliandaru, Adam Rotal, 2016, Teknik Kriptografi Hill Cipher Menggunakan Matriks, Makalah IF2123 Aljabar Geometri – Informatika ITB Tahun 2015/2016, Bandung.M. Wegmuller, J. P. von der Weid,

[4] Malik, Mubina & Patel, Trisha, 2016, Database Security - Attacks and Control Methods, International Journal of Information Sciences and Techniques (IJIST) Vol.6, No.1/2, March 2016, CMPICA, Charotar University of Science & Technology (CHARUSAT), Changa.

[5] Leohani, Ricky Antonius & Agus, Imaludin, 2016, Proses Enkripsi dan Dekripsi Email menggunakan Algoritma Advanced Encryption Standard (AES), Seminar Nasional Matematika dan Pendidikan Matematika UNY 2016, Yogyakarta.

[6] Setyawan, Rendi, 2009, Jenis-jenis Enkripsi, https://id.scribd.com/doc/283280835/jenis-jenis-enkripsi.pdf diakses pada tanggal 2 juli 2017.

[7] "PostgreSQL: History". PostgreSQL Global Development Group. Retrieved 27 August 2016.

[8] *"Procedural Languages". postgresql.org. 2016-03-31. Retrieved 2016-04-07.*