Operasi Komputasi pada Basis Data Relasional yang Terenkripsi Memanfaatkan Sifat Homomorfik Algoritma Pailier

Untuk Operasi CRUD dan Penjumlahan

Muhtar Hartopo / 13513068

Teknik Informatika

Institut Teknologi Bandung (ITB)

Jalan Ganesa 10 Bandung, Indonesia

muhtarhartopo@gmail.com

***Abstract*—Saat ini kemanan data menjadi salah satu masalah terutama pada data yang disimpan di server yang bukan milik sendiri. Enkripsi menjadi cara untuk menjaga data tersebut terap aman. Namun salah satu kekurangan pada data yang terenkripsi adalah sulitnya melakukan komputasi terhadap data yang disimpan sebab data harus didekripsi terlebih dahulu. Algortima pailier merupakan salah satu algortima kriptografi kunci publik yang memiliki sifat homomorfik yang memungkinkan dilakukannya oprasi pada data terenkripsi tanpa melakukan dekripsi terlebih dahulu. Pada makalah ini akan dibahas bagaimana menerapkan algortima pailier pada enkripsi basis data relasional yang memungkinakan untuk dilakukan operasi pada basis data tersebut.**

Keywords—pailier;basis data; operasi; enkripsi; dekripsi.

I. PENDAHULUAN

Era digital memungkinkan manusia menyimpan berkas bukan dalam bentuk fisik lagi melainkan dalam bentuk digital. Berkas tersebut dapat disimpan di media penyimpanan seperti *Hard disk*, *CD*, *DVD*, *Flash drive* maupun di *cloud*. Selain mudah penyimpanan data dalam bentuk digital juga murah dibandingkan dengan penyimpanan dalam bentuk fisik.

Walaupun memiliki banyak keuntungan, penyimpanan dalam bentuk digital tentu akan memiliki risiko keamanan seperti pencurian data dan risiko integritas data. Solusi untuk mengatasi permasalagan tersebut adalah dengan menggunanak enkripsi pada data. Lalu bagaimana jika kita memiliki data penting yang terenkripsi kemudian disimpan di sebuah server, lalu kita ingan melakukan komputasi pada data tersebut.

Sebagai contoh sebuah perusahaan memiliki data penjualan bulanan yang disimpan di sebuah server yang disewa dan data tersebut merupakan data yang sangat rahasia sehingga data tersebut dienkripsi. Kemudian pada suatu saat perusahaan tersebut ingin mengetahui total penjulana perusahaannya selama 12 bulan terakhir untuk analisis bisnisnya. Perusahaan tersebut harus mendekripsi data yang ada di server dengan mengrimkan kunci terlebih dahulu agar bisa didekripsi kemudian baru bisa menghitung total penjualannya. Tentu hal ini memiliki risiko yang besar karena bisa saja ada pihak tertentu yang menyadap pengiriman kunci perusahaan.

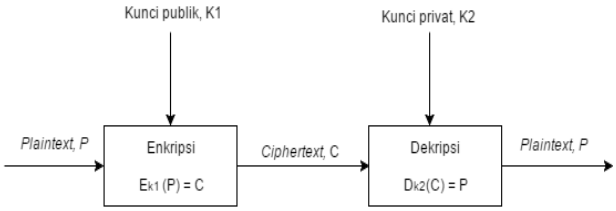
Solusi untuk permasalahan tersebut salah satunya adalah dengan melakukan operasi pada data yang terenkripsi agar tidak ada pengiriman kunci yang dilakukan ke server. Total penjualan perusahaan akan dihitung dalam bentuk terenkripsi. Total penjualan terenkripsi tersebut kemudian akan dikirimkan dari server ke client kemudian di dekripsi di sisi client. Cara ini dimungkinkan dengan menggunanak algoritma pailier karena algoritma pailier memiliki sifat homomorfik *additive*.

Pada makalah ini akan dibahas mengenai cara menerapkan sifat homomorfik pada algoritma pailier untuk melakukan operasi komputasi pada basis data yang terenkripsi. Operasi tersebut adalah operasi *read, insert, delete, update* dan operasi penjumlahan.

II. DASAR TEORI

A. Algotritma Kritografi kunci publik

Algortima kriptografi kunci publik merupkan algoritma kriptografi yang menggunakan kunci yang berbeda untuk melakukan enkripsi dan dekripsi pesan. Kunci publik digunakan untuk mengenkripsi pesan dan kunci privat digunakan untuk mendekripsi pesan. Skema kriptografi kunci publik dapat dilihat pada Gambar 1.



Gambar 1 Skema algoritma kriptografi kunci publik

B. Algortima Pailier

Algortima Pailier merupakan algoritma kriptografi kunci publik yang menggunakan *probabilistic symetric algoritm*. Algoritma ini ditemukan oleh Pascal Pailier pada tahun 1999. Algoritma Pailier didasarkan pada sulitnya menghitung residu kelas ke-n atau disebut *composite residuosity problem*. Suatu bilangan bulat z dikatakan sebagai residu ke-n modulo n^2 jika terdapat bilangan bulat y sehingga $z = y^n \ mod \ n^2$ [1].

Untuk membangkitkan pasangan kunci pada algoritma Pailier, berikut ini langkah-langkah yang harus dilakukan :

1. Pilih dua buah bilangan prima p dan q yang memenuhi syarat $gdc\big(pq, (p-1)(q-1)\big) = 1$
2. Hitunng $n = pq$ dan $\lambda = lcm\ (p-1, q-1)$
3. Pilih sembarang bilangan bulat g, dengan $g < n^2$
4. Hitung $\mu = (L\big(g^\lambda \ mod \ n^2\big))^{-1} \ mod \ n$
   Dengan fungsi L adalah $L(x) = \frac{x-1}{n}$

Hasil dari langkah-langkah diatas adalah kunci publik yang berupa pasangan g, n dan kunci privat yang berupa pasangan λ, μ.

Proses enkripsi dengan algoritma Pailier dilakukan dengan cara :

1. Bagi *plaintext* menjadi blok-blok sehingga nilai setiap blok *plaintext* lebih kecil dari n.
2. Pilih bilangan bulat r dimana r < n.
3. Enkripsi setiap blok dengan rumus
   $c = \ g^m \ r^n \ mod \ n^2$,
   dengan m adalah blok *plaintext*.

Kemudian proses dekripsi pada algoritma Pailier dilakukan dengan cara :

1. Dekripsi tiap blok *ciphertext* dengan rumus
   $m = \frac{L(c^\lambda \ mod \ n^2)}{L(g^\lambda \ mod \ n^2)} \ mod \ n$, atau
   $m = \ L\big(c^\lambda \ mod \ n^2\big).\mu \ mod \ n$
2. Gabungkan blok-blok *plaintext* menjadi pesan utuh.

C. Sifat Homomorfik Pailier untuk Penjumlahan

Algoritma pailier merupakan algoritma kriptografi yang memiliki sifat homomorfik dan termasuk salah satu algortima enkripsi homomorfik [2].

Enkripsi Homomorfik (*homomorphic encryption*) merupakan suatu bentuk enkripsi yang memungkinkan dilakukannya komputasi pada pada *ciphertext* tanpa mendekripsi terlebih dahulu *ciphertext* tersebut. Operasi yang dilakukan pada *ciphertext* yang menggunakan enkripsi homomorfik akan menghasilkan *ciphertext* yang jika didekripsi akan menghasilkan hasil yang sama dengan operasi serupa pada *plaintext* [4].

Secara matematis, *homomorphic cryptosystem* adalah sebuah *cryptosystem* yang menggunakan fungsi enkripsi yang bersifat homomorfik dan memungkinkan dilakukannya operasi pada *ciphertext*. Terdapat dua jenis operasi utama yaitu penjumlahan dan perkalian.

Suatu kriptosistem dikatakan bersifat additif jika dan hanya jika :

$$\exists \Delta: \ \varepsilon(x1) \ \Delta \ \varepsilon(x2) = \ \varepsilon(x1 + x2)$$

Dengan $x1$ dan $x2$ adalah *plaintext*, ε adalah fungsi enkripsi dan Δ adalah suatu operasi yang bergantung pada sifat algoritma enkripsi yang digunakan. Kemudian suatu kriptosistem dikatakan bersifat multiplikatif jika dan hanya jika:

$$\exists \Delta: \ \varepsilon(x1) \ \Delta \ \varepsilon(x2) = \ \varepsilon(x1.x2)$$

Pailier merupakan algoritma yang memiliki sifat *additive* (penjumlahan). Dengan mengalikan dua buah *ciphertext* Pailier maka hasil dekripsinya akan ekuivalen dengan penjumlahan dua buah nilai *plaintext*-nya [2].

Sifat *additive* algoritma pailier dapat dibuktikan sebagai berikut. Misalkan a dan b adalah dua buah *plaintext*, kemudian $c1$ dan $c2$ adalah *ciphertext* dari masing-masing a dan b dengan enkripsi ElGamal dan g, y adalah pasangan kunci publik maka

$$c1 = \ g^a r_1^n \ mod \ n^2$$
$$c2 = \ g^b r_2^n \ mod \ n^2$$

Dengan mengalikan c1 dan c2 maka akan diperoleh hasil :

$$c1.c2 = \ g^a r_1^n . g^b r_2^n \ mod \ n^2$$
$$= \ g^{a+b} (r_1 r_2)^n \ mod \ n^2$$

Ketika dilakukan dekripsi maka :

$$d(g^{a+b}(r_1 r_2)^n) = a + b$$

Jadi $d(c1.c2) \ = \ a \ + \ b$.

D. Basis Data Relasional

Model Data Relasional adalah suatu model basis data yang menggunakan tabel dua dimensi, yang terdiri atas baris dan kolom untuk menggambarkan sebuah berkas data. Model ini menunjukkan cara mengelola/mengorganisasikan data secara fisik dalam memory sekunder, yang akan berdampak pula pada bagaimana kita mengelompokkan data dan membentuk keseluruhan data yang terkait dalam sistem yang kita buat.

Dalam basis data relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, dan antara tabel satu dengan tabel lainnya terdapat hubungan atau relationship sehingga dapat di simpulkan, basis data adalah kumpulan dari sejumlah tabel yang saling hubungan atau saling keterkaitan. Kumpulan dari data yang diorganisasikan sebagai tabel tadi disimpan dalam bentuk data elektronik di dalam harddisk komputer dan dikelompokan secara logis berdasarkan *schema* user [3].

III. Rancangan Implementasi

Pada pengimplementasian basis data terenkripsi ini, hanya nilai pada record basis data yang dienkripsi, skema dan nama kolom pada basis data tidak terenkripsi. Enkripsi dapat dilakukan pada record-record pada tiap kolom atau hanya pada kolom tertentu saja. Perlu diingat bahwa seluruh proses enkrisi dan dekripsi hanya dilakukan di sisi *client*, tidak ada proses enkripsi ataupun dekripsi yang dilakukan disisi server. Hal ini dilakukan untuk menjamin keamana data saat pengiriman dan saat berada pada server.

Terdapat lima operasi terhadap basis data yang akan diimplementasikan yaitu operasi *read*, *insert*, *update*, *delete* dan operasi penjumlahan.

A. Operasi Read

Operasi ini adalah operasi untuk membaca record dari basis data. Karena record dari basis data disimpan dalam bentuk terenkripsi, maka untuk mebaca basis data maka perlu dilakukan pembacaan data menggunakan perintah *select* pada SQL terlebih dahulu. Setelah dilakukan dekripsi terhadap data yang terenkripsi.

B. Operasi Insert (Create)

Operasi *insert* digunakan untuk menyisipkan sebuah record ke dalam basis data. Karena data yang disimpan pada record basis data bentuknya terenkripsi maka data terlebih dahulu harus dienkripsi menggunakan kunci publik lalu dilakukan penyisipan dengan perintah instert menggunakan SQL.

C. Operasi Update

Operasi *update* yaitu operasi yang mengubah data pada suatu record. Operasi ini dilakukan hampir sama dengan *insert* yaitu dengan mengenkripsi data terlebih dahulu kemudian dilakukan *update* dengan perintah SQL biasa.

D. Operasi Delete

Operasi *delete* adalah operasi yang menghapus record tertentu pada sebuah tabel. Operasi ini bisa dilakukan tanpa melakukan enkripsi data terlebih dahulu jika kondisi untuk melakukan *delete* tidak dipengaruhi oleh data yang terenkripsi. Tetapi jika kondisi untuk melakukan *delete* dipengaruhi oleh data yang terenkripsi maka data harus dienkripsi terlebih dahulu kemudian dilakukan *delete* dengan perintah SQL biasa.

E. Operasi Jumlah

Operasi ini melakukan penjumlah nilai pada kolom yang terenkripsi. Operasi ini dilakukan dengan memanfaatkan sifat homomorfik dari algoritma pailier. Penjumlahan dilakukan sesuai dengan sifat homomorfik algoritma pailiert yaitu dengan mengalikan nilai dari dua buah data yang terenkripsi. Operasi ini dapat dilakukan disisi server karena tidak melibatkan proses enkripsi dan dekripsi.

IV. Implementasi Analisis Hasil

Bersamaan dengan penulisan makalah ini dilakukan implementasi program sederhana untuk melakukan operasi CRUD dan penjumlahan pada basis data sederhana. Basis data memiliki beberapa tabel, namun pada percobaan yang dilakukan, enkripsi yang dilakukan hanya pada satu tabel saya yaitu tabel penjualan. Tabel penjualan sendiri memiliki beberapa kolom yaitu kolom id, kolom tanggal dan kolom penjualan.

Percobaan dilakukan dengan panjang kunci yang berbeda yaitu 32 bit, 64 bit, 128 bit, 256 bit, 512 bit dan 1024 bit. Percobaan pertama dilakukan untuk melihat kecepatan enkripsi dan dekripsi pesan berdasarkan kunci tertentu. Hasil percobaan tersebut dapat dilihat pada Tabel 1. Berdasarkan hasil percobaan diperoleh hasil bahwa waktu enkripsi dan dekripsi pesan sebanding dengan panjang kunci yang digunakan. Kemudian pertamahan panjang kunci menjadi dua kali lipat menyebabkan waktu komputasi menjadi empat kali lipat.

Tabel 1 Hasil eksperimen enkripsi dan dekripsi

No	Ukuran Kunci	Waktu enkripsi (ms)	Waktu dekripsi (ms)
1	32	0,16	0,14
2	64	0,47	0,23
3	128	0,8	1
4	256	1,3	1,1
5	512	4,8	4,2
6	1024	26,9	14,9

Pada percobaan operasi penjumlahan dilakukan penjumlahan *ciphertext* berdasarkan sifat homomorfik algoritma pailier (*ciphertext* dikalikan). Operasi ini dilakukan pada 100 data pada kolom penjualan dengan kisaran nilai *plaintext* 20 hingga 1000. Operasi tersebut diulangi untuk panjang kunci yang berbeda. Hasil percobaan tersebut dapat dilihat pada Tabel 2.

Berdasarkan percoabaan tersebut dapat dilihat bahwa dekripsi dari hasil penjumlahan homomorfik pada *ciphertext* menghasilkan hasil yang sama dengan penjumlahan data yang tidak terenkripsi. Sementara itu pada performa operasi menunjukkan bahwa waktu operasi yang dibutuhkan sebanding dengan ukuran kunci yang digunakan. Hal ini disebabkan karena ukuran ukuran *ciphertext* akan menjadi dua kali lipat juga sehingga waktu operasi juga akan meningkan hampir dua kali lipat.

Tabel 2 Hasil eksperimen operasi penjumlahan

No	Ukuran Kunci	Waktu (ms)	Jumlah (Hasil dari program)	Jumlah sebenarnya	Ket
1	32	5	6229	6229	Berhasil
2	64	5	6229	6229	Berhasil
3	128	6	6229	6229	Berhasil
4	256	10	6229	6229	Berhasil
5	512	18	6229	6229	Berhasil
6	1024	38	6229	6229	Berhasil

Pada operasi *read*, *insert* dan *update* kecepatan operasi dipengaruhi oleh panjang kunci yang digunakan. Hasil percobaan operasi *read*, *insert* dan *update* berturut-turut dapat dilihat pada Tabel 2, Tabel 4 dan Tabel 5. Dari tabel-tabel tersebut dapat dilihat bahwa semakin panjang kunci yang digunakan maka waktu untuk melakukan operasi juga akan semakin lama. Hal ini disebabkan karena ukuran *ciphertext* juga akan semakin panjang atau ukuran data semakin besar sehingga waktu untuk melakukan query *select*, *insert* dan *update* ke basis data akan semakin lama. Kemudian operasi *read*, *insert* dan *update* juga dipengaruhi olehh waktu untuk melakukan enkripsi dan dekripsi data.

Tabel 3 Hasil eksperimen operasi read

No	Ukuran Kunci	Waktu (ms)	Waktu + decrypt (ms)
1	32	1	1,93
2	64	1	1,95
3	128	1	2,11
4	256	1	2,8
5	512	1	5,8
6	1024	1	17,8

Tabel 4 Hasil eksperimen operasi insert

No	Ukuran Kunci	Waktu (ms)	Waktu + encript (ms)
1	32	30	30,13
2	64	33	33,15
3	128	34	34,31
4	256	55	56
5	512	53	57
6	1024	60	76

Tabel 5 Hasil eksperimen operasi update

No	Ukuran Kunci	Waktu (ms)	Waktu + encrypt (ms)
1	32	1	1,13
2	64	1	1,15
3	128	1	1,31
4	256	2	3
5	512	5	9
6	1024	9	25

Pada operasi *delete* tidak dilakukan proses enkripsi ataupun dekripsi sehingga kecepatan operasi hanya dipengaruhi oleh kecepatan untuk melakukan *delete* ke basis data. Hasil percobaan tersbut dapat dilihat pada Table 6. Pada Tabel 6 dapat dilihat bahwa kecepatan operasi *delete* tidak dipengaruhi oleh panjang kunci yang digunakan. Hal ini disebabkan karena pada proses penghapusan, basis data hanya perlu mengetahui lokasi penyimpanan record yang akan dihapus lalu menghapus pointer ke record tersebut..

Tabel 6 Hasil ekperimen operasi delete

No	Ukuran Kunci	Waktu (ms)
1	32	39
2	64	60
3	128	54
4	256	39
5	512	41
6	1024	31

V. Analisis Keamanan

Keamanan pada operasi basis data yang memanfaatkan algoritma pailier bergantung sepenuhnya pada keamanan algoritma pailier tersebut. Algortima pailier didasarkan pada masalah *Composite Residuosity Class Problem* atau permasalahan mencari residu komposit pada suatu kelas. Permasalahan tersebut adalah permasalahan dengan kompleksitas algortima yang eksponensial bergantung pada panjang kunci yang digunakan. Hingga saat ini belum ada algoritma yang efisien untuk menyelesaikan permasalahan tersebut.

Keamanan dalam melakukan operasi pada basis data terenkripsi masih terjamin selama algoritma pailier masih terjamin keamanannya. Hanya saja perlu diingat bahwa untuk menjamin algoritma pailier tetap aman adalah dengan menggunakan kunci yang panjang. Semakin panjang kunci yang digunakan maka akan semakin aman. Namun akibatnya operasi komputasi pada basis data yang terenkripsi akan semakin lama.

VI. Kesimpulan dan Saran

Penggunaan enkripsi dengan algoritma pailier pada basis data dapat meningkatkan keamanan basis data tersebut. Penggunaan enkripsi pailier juga memungkinan dilakukannya beberapa operasi pada basis data tersebut seperti *read*, *insert*, *update*, *delete* dan operasi penjumlahan. Penggunaan kunci yang panjang akan meningkatkan keamanan data namun akibatnya waktu untuk melakukan operasi pada data juga akan semakin lama yang disebabkan karena ukuran *ciphertext* yang juga semakin besar.

Untuk pengembangan selanjutnya perlu dilakukan pengembangan pada cara untuk melakukan operasi pada data agar waktu komputasi dapat berkurang.

ACKNOWLEDGMENT

Program untuk melakukuan eksperimen dapat dilihat pada https://github.com/mhartopo/jpalilier.

REFERENCES

[1] Pailier Pascal ,"Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", Gemplus Card International, 1999

[2] Morris Lieam, "*Analysis of Partial and Fully Homomorphic Encription"*, Rochester Institute of Technology, 2013

[3] Silberschatz Abramam dkk, "Database System Concept 6th edition", McGraw Hill, 2011.

[4] Potzelsberger, "KV Web Security: Application of Homomorphic Encryption". 2013
http://www.fim.uni-linz.ac.at/lva/Web_Security/Abgaben/Poetzelsberger-Homomorphic.pdf . Diakses pada 16 Desember 2016

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Desember 2016

Muhtar Hartopo
13513068