STMIK Amik Riau

SATIN – Sains dan Teknologi Informasi

journal homepage : http://jurnal.stmik-amik-riau.ac.id

Penerapan Keamanan Penggunaan Data pada Database Kepegawaian Menggunakan Teknik *Transparent Data Encryption* (Studi Kasus Sekolah Tinggi Teknologi Payakumbuh)

Arif Budiman
Sekolah Tinggi Teknologi Payakumbuh
budiman024@gmail.com

Noviardi
Sekolah Tinggi Teknologi Payakumbuh
maha_rajo@yahoo.co.id

Abstrak

Database kepegawaian yang baik adalah database yang terlindung dari akses orang yang tidak berkepentingan. Meskipun database sudah terlindungi dengan password, belum menjamin amannya sebuah database. Apalagi data tersebut berupa data sensitif atau data yang hanya boleh diketahui oleh orang-orang tertentu, namun ada saja cara yang bisa digunakan untuk mengambil data dan memanipulasinya. Melindungi data dengan teknik Transparent Data Encryption (TDE) mampu mengamankan data semaksimal mungkin. Meskipun data tersebut di copy atau dicuri secara fisik, data tersebut tetap tidak akan bisa dibuka tanpa proses dekripsi TDE. Penelitian ini bertujuan untuk menerapkan sistem keamanan pada database kepegawaian Sekolah Tinggi Teknologi Payakumbuh dengan teknik TDE. Penelitian ini dilakukan dengan cara memisahkan data sensitif dengan data non sensitif, kemudian data sensitif tersebut akan di enkripsi menggunakan algoritma yang disediakan oleh TDE, sehingga keamanan terhadap data sensitif dapat diimpelentasikan dengan baik

***Kata Kunci** : Database, TDE, Enkripsi*

1. Pendahuluan

Database yang baik adalah *database* yang terjamin keamannya, mudah dalam penggunaan dan dapat dipertanggungjawabkan. Keamanan *database* sekarang ini mengkhawatirkan, mengingat banyaknya kejadian kehilangan data, penyalah gunaan data dan pencurian. Perkembangan penggunaan *database* berbarengan pula dengan keharusan pemahaman akan keamanan *database*. (Murray, 2010) Masih banyak lembaga atau instansi yang menyepelekan keamanan *database*, sehingga data yang ada bisa diakses oleh orang yang tidak berhak mendapatkannya. Banyak cara yang bisa dilakukan untuk bisa mengamankan sekumpulan data tersebut. Secara fisik, bisa disimpan dalam tempat yang terkunci rapat, dan dalam ruangan yang keamanannya sangat tinggi. Bisa juga disimpan dalam sebuah bangker yang dalam sehingga tidak sembarang orang yang bisa menemukan fisik *database* tersebut. Kemudian *database* juga bisa diamankan dengan *software*, baik itu dengan penggunaan *password*, maupun dengan aplikasi khusus. Namun masih tetap saja akan bisa diambil, di kopi *paste*, atau dicuri fisiknya dan dimanipulasi. Menerapkan enkripsi pada sebuah data adalah salah satu cara yang bisa bisa digunakan untuk melindungi data karena data ini akan dienkripsi dan dirubah formatnya dari plaintext kepada *chipertext*. (Wibowo, Susanto, & Karel, 2011).

Database kepegawaian pada Sekolah Tinggi Teknologi Payakumbuh (STTP) merupakan *database* yang sensitif, yang menyimpan semua data karyawan dan dosen dilingkungan STTP. Data ini sangat bersifat rahasia, tidak semua orang bisa mengakses dan menggunakannya. Karena pada *database* kepegawaian tersimpan informasi mengenai identitas, besaran gaji, nomor rekening dan informasi penting lainnya, yang apabila disalah gunakan, dapat berakibat fatal bagi instansi maupun orang pribadi di lingkungan STTP. Menerapkan TDE pada *database* kepegawaian, bertujuan untuk mengamankan data secara maksimal, namun fleksibel dalam penggunaan. Setiap orang yang mengakses database, akan mendapatkan informasi

yang relevan sesuai dengan hak akses orang tersebut terhadap *database*. Dengan TDE data akan diekripsi menggunakan sebuah *master key*, dan *master key* ini akan disimpan dalam sebuah *wallet* didalam komputer. Sehingga walaupun kita mengetahui *master key* nya, namun *key* untuk *wallet* tidak, tetap tidak akan bisa membuka data yang terenkripsi. Sehingga apabila terjadi pencurian fisik data sekalipun, tetap tidak akan bisa di dekripsi tanpa *key* diatas.

2. Tinjauan Pustaka

2.1 *Encryption* / Enkripsi

Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan dan media yang khusus. Enkripsi juga bisa disebut sebagai proses yang paling efektif untuk mencapai keamanan data yang baik, karena proses enkripsi ini adalah proses menyembunyikan data dan hanya bisa dibuka lagi melalui proses dekripsi (Becker, 2007). Sehingga proses enkripsi ini sangat bagus untuk mengamankan data yang sensitif dari gangguan akses yang tidak seharusnya.

2.2 *Transparent Data Encryption*

Transparent Data Encryption(TDE) menyediakan kriptografi yang transparan pada user yang sah tetapi tidak pada penyusup dari luar maupun dari dalam. Enkripsi ini digunakan untuk kasus apabila terjadi pencurian *hardware* atau media backup atau *unauthorizedaccess* pada data yang sensitif di level sistem operasi. Salah satu cara untuk mengatasi pencurian media adalah mengenkripsi data yang sensitif di dalam *database* dan menyimpan *encryption key* nya di lokasi yang terpisah. Karena memang tujuan utamanya dari TDE ini adalah untuk memberikan keamanan data sampai kelevel baris didalam sebuah *database*. (Pasha & Gafoor, 2011) Tetapi harus dipertimbangkan keseimbangan antara dua konsep yang bertentangan dan kemudahan dimana aplikasi bisa mengakses *encryption keys* dan keamanan yang diperlukan bila terjadi pencurian *key*. (Sudrajat, 2006)

2.3 Konsep *Transparent Data Encryption*

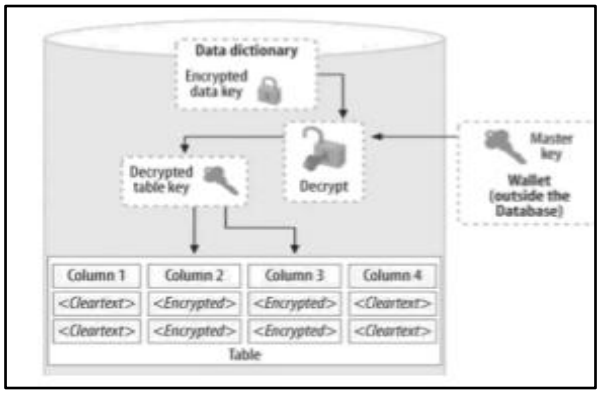
Dalam menciptakan TDE pada sebuah *database*, perlu dipahami beberapa konsep berikut :

2.3.1 *Master Encryption Key*. *Master Encryption Key* adalah bagian dari arsitekur pengamanan dua tingkat yang melindungi kunci enkripsi. *Master Encryption Key* disimpan pada sebuah lokasi dalam *database* yang disebut dengan *wallet*, sedangkan untuk membuka *wallet* diperlukan sebuah kunci / *walletkey*. Arsitekur seperti ini memungkin untuk memberikan kinerja keamana tingkat tinggi.

2.3.2 *Wallet.* *Wallet* adalah sebuah lokasi yang disediakan didalam *database* yang digunakan untuk menyimpan *Master key*. *Wallet* sendiri mempunyai *Wallet* key yang dibutuhkan untuk membuka *wallet*. *Master key* tidak akan bisa diambil bila *wallet* key tidak ada. *Wallet* secara default akan tersimpan pada direktori *wallet*. Dan bisa dirubah kelokasi yang diinginkan.

2.3.3 *Advanced Encryption Standard (AES).* Merupakan algoritma standart yang dipakai untuk melakukan enkripsi. Algoritma ini teridiri dari bebarapa jenis diantaranya AES256, AES192, AES128. Digunakan sesuai kebutuhan enkripsi pada database. Pengelompokan jenis AES ini adalah berdasarkan panjang kunci yang digunakan. Angka-angka dibelakang kata AES menggambarkan panjang kunci yang digunakan pada tiap tiap AES, karena algoritma AES ini baik digunakan untuk enkripsi file, teks, ataupun *database*. (Silva, 2013).

2.3.4 Cara Kerja TDE. TDE dapat diimplementasikan pada *object table* dan juga bisa di terapkan secara langsung pada *tablespace*. Pada saat sebuah *table* (bisa jadi satu kolom atau semua kolom) telah dienkripsi kemudian seorang user menginputkan data pada sebuah kolom tersebut, sistem *database* mengambil *master key* dari *wallet*, mendekripsi *encryption key* untuk *table* tersebut dari *data dictionary*, menggunakan *encryption key* pada nilai *input* dan menyimpan data yang dienkripsi pada *database*. Gambar 1, menjelaskan bagaimana proses enkripsi pada sebuah tabel.



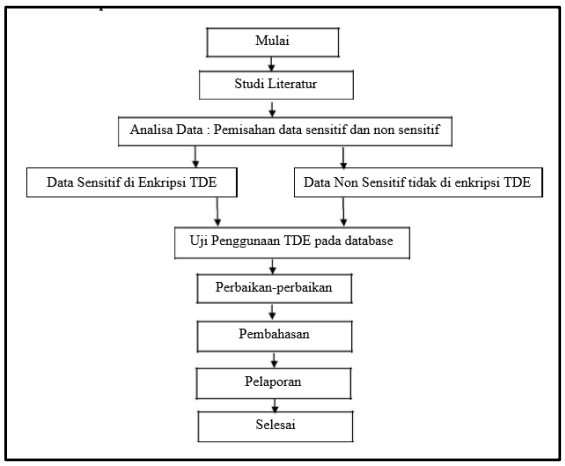
Gambar 1 : Alur proses *tranparent data encryption*

3. Metode Penelitian

Penelitian ini dilakukan dengan cara mengamati langsung penggunaan akses *database*, kemudian memisahkan elemen data yang sensitif dan *non*

sensitif, kemudian menerapkan enkripsi pada masing-masing data. Data yang sudah dienkripsi diharapkan tidak akan mengganggu kinerja akses database. Kemudian apabila data tersebut dibutuhkan dapat di dekripsi kembali menggunakan proses dekripsi TDE. Pada penelitian ini akan menggunakan semua data kepegawaian, dan mengamati efektifitas penggunaan teknik enkripsi ini, kemudian memberikan perbaikan jika ada.

Langkah Kerja Pelaksanaan Penelitian

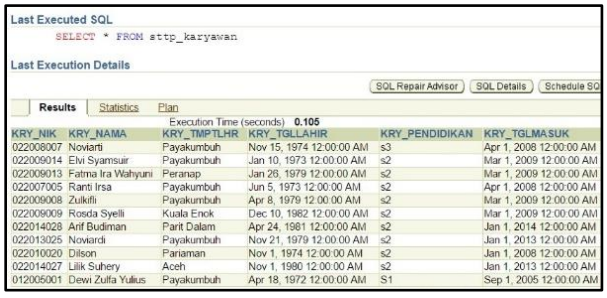


Gambar 2 : Tahapan Penelitian

4. Hasil dan Pembahasan

4.1Memilih Data Sensitif dan Non Sensitif

Data non sensitif tidak akan dienkripsi dengan TDE, data ini dapat dilihat oleh siapapun. Berikut adalah data non sensitif.



Last Executed SQL

```
SELECT * FROM sttp_karyawan
```

Last Execution Details

SQL Repair Advisor | SQL Details | Schedule SQ

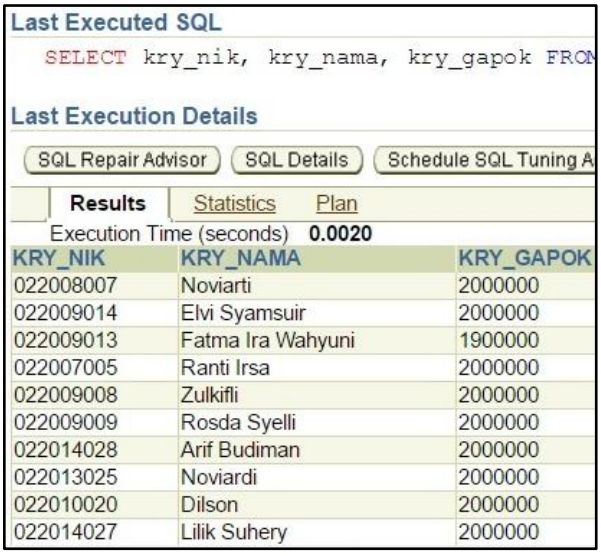
Results | Statistics | Plan

Execution Time (seconds) **0.105**

KRY_NIK	KRY_NAMA	KRY_TMPTLHR	KRY_TGLLAHIR	KRY_PENDIDIKAN	KRY_TGLMASUK
022008007	Noviarti	Payakumbuh	Nov 15, 1974 12:00:00 AM	s3	Apr 1, 2008 12:00:00 AM
022009014	Elvi Syamsuir	Payakumbuh	Jan 10, 1973 12:00:00 AM	s2	Mar 1, 2009 12:00:00 AM
022009013	Fatma Ira Wahyuni	Peranap	Jan 26, 1979 12:00:00 AM	s2	Mar 1, 2009 12:00:00 AM
022007005	Ranti Irsa	Payakumbuh	Jun 5, 1973 12:00:00 AM	s2	Apr 1, 2008 12:00:00 AM
022009008	Zulkifli	Payakumbuh	Apr 8, 1979 12:00:00 AM	s2	Mar 1, 2009 12:00:00 AM
022009009	Rosda Syelli	Kuala Enok	Dec 10, 1982 12:00:00 AM	s2	Mar 1, 2009 12:00:00 AM
022014028	Arif Budiman	Parit Dalam	Apr 24, 1981 12:00:00 AM	s2	Jan 1, 2014 12:00:00 AM
022013025	Noviardi	Payakumbuh	Nov 21, 1979 12:00:00 AM	s2	Jan 1, 2013 12:00:00 AM
022010020	Dilson	Pariaman	Nov 1, 1974 12:00:00 AM	s2	Jan 1, 2008 12:00:00 AM
022014027	Lilik Suhery	Aceh	Nov 1, 1980 12:00:00 AM	s2	Jan 1, 2013 12:00:00 AM
012005001	Dewi Zulfa Yulius	Payakumbuh	Apr 18, 1972 12:00:00 AM	S1	Sep 1, 2005 12:00:00 AM

Gambar 3 : Data non sensitif

Setelah mendapatkan non sensitif, maka yang menjadi data sensitif adalah data gaji seperti gambar dibawah :



Last Executed SQL

```
SELECT kry_nik, kry_nama, kry_gapok FRON
```

Last Execution Details

SQL Repair Advisor | SQL Details | Schedule SQL Tuning A

Results | Statistics | Plan

Execution Time (seconds) **0.0020**

KRY_NIK	KRY_NAMA	KRY_GAPOK
022008007	Noviarti	2000000
022009014	Elvi Syamsuir	2000000
022009013	Fatma Ira Wahyuni	1900000
022007005	Ranti Irsa	2000000
022009008	Zulkifli	2000000
022009009	Rosda Syelli	2000000
022014028	Arif Budiman	2000000
022013025	Noviardi	2000000
022010020	Dilson	2000000
022014027	Lilik Suhery	2000000

Gambar 4 : Data sensitif

4.2Menerapkan TDE

Dalam proses ini yang pertama sekali dilakukan adalah menyiapkan lokasi *wallet* pada database kepegawaian menggunakan Enterprise Management Console seperti berikut :

ENCRYPTION_WALLET_LOCATION =
(SOURCE= (METHOD=file)
(METHOD_DATA=DIRECTORY=/myWallet))

Membuat *Wallet*

Alter system set Encryption Key authenticated by "wall5TTP";

Perintah diatas adalah perintah untuk membuat sebuah *wallet* dengan pasword "wall5TTP" dan sekaligus membuka *wallet* untuk TDE yang berguna untuk menyimpan dan mengambil kembali *master key*.

4.3Membuka *Wallet*

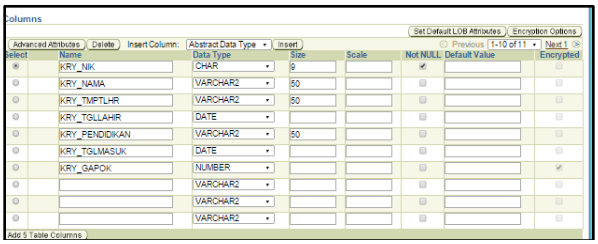
Saat menjalankan *instance database*, sebenarnya secara otomatis proses membuka *wallet* juga sedang terjadi. *Wallet* yang sedang terbuka akan memungkinkan user untuk mengakses dan melihat *database*. Namun setelah *wallet* ditutup, otomatis semua data yang dienkripsi tidak dapat diakses sama sekali. Berikut adalah sintak untuk membuka *wallet*

Alter system set encryption wallet open authenticated by "wall5TTP";

Saat *wallet* terbuka, semua data tidak akan di enkripsi, untuk meng-enkripsi data tutup dulu *wallet*

Alter system set wallet close;

Pada saat *wallet* terbuka, user baru bisa menerapkan proses TDE. Proses TDE dapat diterapkan secara langsung dengan *tools* yang tersedia.



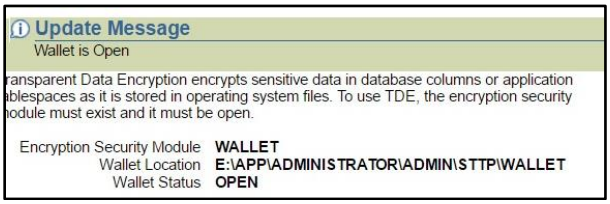
Gambar 5 : Memberikan enkripsi untuk *field* gaji pokok

Proses diatas adalah memberikan enkripsi menggunakan algoritma AES192 pada field yang dianggap sensitif. Algoritma yang digunakankan merupakan algoritma standar TDE.

4.4 Menguji TDE yang sudah diterapkan

Untuk menguji TDE yang sudah diterapkan pada database kepegawaian, *wallet* terlebih dahulu harus ditutup, ini dilakukan untuk membuktikan apakah TDE bekerja.

Alter system set encryption wallet open authenticated by "wall5TTP";



Gambar 6 : Membuka *wallet* setelah dienkripsi

Pada saat *wallet* terbuka, semua perintah untuk pemanggilan data pada tabel yang field nya sudah dienkripsi dapat dilakukan seperti biasa.

*Select * from pegawai;*

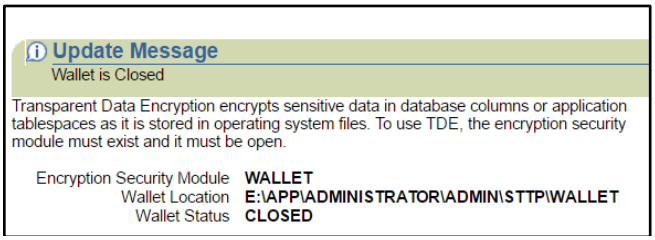
Gambar 7 : Hasil *query* sebelum enkripsi

Jika TDE tidak diaktifkan, maka akan ditampilkan informasi seperti pada gambar 7.

4.5 Menguji TDE dengan menutup *wallet*

Untuk menguji TDE yang telah diterapkan, terlebih dahulu harus menutup *wallet* atau menjalankan proses enkripsi

Alter system set encryption wallet close;



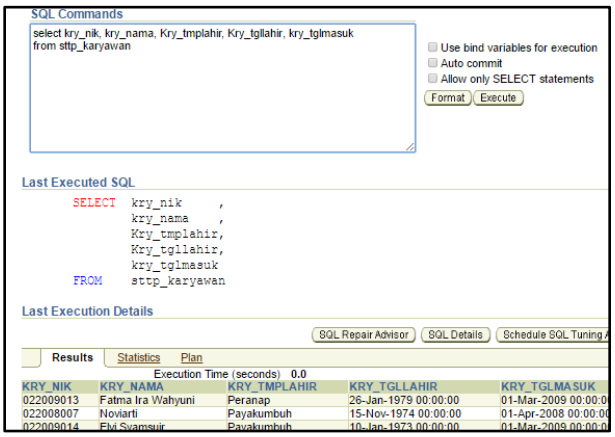
Gambar 8 : Menutup *Wallet*

Setelah *wallet* ditutup, pengujian dilakukan kembali dengan cara memanggil data dengan perintah berikut :

"*select * from sttp_karyawan"*

Gambar 9 : Hasil jika *wallet* telah ditutup

Data tidak bisa ditampilkan, karena proses enkripsi sedang berjalan. Semua field yang diekripsi tidak akan tampil yang menyebabkan keseluruhan data tidak bisa ditampilkan juga. Namun jika TDE tidak bekerja, data akan menampilkan semua data yang ada, baik itu data sensitif, maupun data nonsensitif, sedangkan untuk pemanggilan terpilih atau pemanggilan untuk data non sensitif pada saat TDE bekerja dapat tetap dilakukan.



Gambar 10. Proses pemanggilan data non sensitif tetap bisa dilakukan

Pada saat proses enkripsi TDE berjalan, sebenarnya hanya data sensitif saja yang dilindungi, sedangkan data non sensitif dapat tetap ditampilkan dengan pemanggilan field-field yang tidak diekripsi.

5. Simpulan

TDE yang diterapkan pada *database* kepegawaian khususnya pada table pegawai dapat memberikan alternatif keamanan yang kuat. Dengan adanya proses ini, secara otomatis, administrator database dapat mengontrol akses data terhadapat data yang tidak seharusnya dapat dilihat oleh semua user. *Administrator* hanya perlu menutup *wallet* jika akan menutup akses dan membuka *wallet* jika akan melihat data yang dianggap sensitif. Penerapan TDE yang dilakukan hanya pada table pegawai, diharapkan tabel lain yang juga memiliki data sensitif dapat segera diamankan dengan teknik TDE ini.

6. Ucapan Terima Kasih

[1]. Terimakasih penulis ucapkan kepada Kementrian Riset dan Pendidikan Tinggi Republik Indonesia yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian ini yang penulis dapatkan dari skema Penelitian Dosen Pemula (PDP). Semoga dengan adanya kesempatan yang diberikan oleh KEMENRISTEK DIKTI ini dapat mendorong penulis dan sejawat dalam meningkatkan kemampuan dalam melakukan penelitian.

[2]. Terimakasih penulis ucapakan kepada institusi tempat penulis mengabdi yaitunya Sekolah Tinggi Teknologi Payakumbuh, yang telah bersedia memberikan penulis data yang diperlukan dalam melalukan penelitian ini.

7. Referensi

Becker, A. (2007). *Transparent Data Encryption. DOAG SAP Special Interest Day 27th June 2007*.

Murray, M. C. (2010). Database Security: What Students Need to Know. *Journal of Information Technology Education*, *9*, 61–77. Retrieved from http://www.eric.ed.gov/ERICWebPortal/detail?accno=EJ895859

Pasha, A., & Gafoor, A. (2011). Transparent Data Encryption- Solution for Security of Database Contents. *International Journal of Advanced Computer Science and Applications*, *2*(3), 25–28.

Silva, L. Da. (2013). Aplikasi Enkripsi Dan Dekripsi File Dengan Menggunakan Aes (Advanced Encryption Standard) Algoritma Rijndael Pada Sistem Operasi Android. *Telematika*, *10*(1), 33–42.

Sudrajat, A. W. (2006). Implementasi Enkripsi Database Menggunakan Transparent Data Encryption Pada Database Engine Oracle. *Algoritma*, *2*, 14–19.

Wibowo, I., Susanto, B., & Karel, J. (2011). Penerapan algoritma kriptografi asimetris rsa untuk keamanan data di oracle. *Jurnal Informatika*, (1). Retrieved from http://labti.ukdw.ac.id/ojs/index.php/informatika/article/viewFile/68/32