TUGAS AKHIR – KI141502

RANCANG BANGUN SISTEM KEAMANAN PENYIMPANAN DATA PADA SISTEM E-VOTING

ASTANDRO KOESRIPUTRANTO
NRP 5111100017

Dosen Pembimbing I
Prof. Ir. Supeno Djanali, M.Sc, Ph.D.

Dosen Pembimbing II
Baskoro Adi Pratomo, S.Kom, M.Kom.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2015

UNDERGRADUATE THESES – KI141502

SECURITY SYSTEM DESIGN OF E-VOTING SYSTEM DATA STORAGE

ASTANDRO KOESRIPUTRANTO
NRP 5111100017

Supervisor I
Prof. Ir. Supeno Djanali, M.Sc, Ph.D.

Supervisor II
Baskoro Adi Pratomo, S.Kom, M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2015

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM KEAMANAN PENYIMPANAN DATA PADA SISTEM E-VOTING

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
ASTANDRO KOESRIPUTRANTO
NRP : 5111 100 017

Disetujui oleh Dosen Pembimbing Tugas Akhir:

1. Prof. Ir. Supeno Djanali, M.Sc, Ph.D. .....
   NIP: 19480619 197301 1 001 (Pembimbing 1)

2. Baskoro Adi Pratomo, S.Kom, M.Kom. .....
   NIP: 19870218 201404 1 001 (Pembimbing 2)

SURABAYA
JULI, 2015

RANCANG BANGUN SISTEM KEAMANAN PENYIMPANAN DATA PADA SISTEM E-VOTING

Nama	: ASTANDRO KOESRIPUTRANTO
NRP	: 5111100017
Jurusan	: Teknik Informatika, FTIf-ITS
Dosen Pembimbing 1	: Prof. Ir. Supeno Djanali, M.Sc, Ph.D.
Dosen Pembimbing 2	: Baskoro Adi Pratomo, S.Kom, M.Kom.

Abstrak

Dewasa ini, perkembangan teknologi informasi telah merambah dunia politik salah satunya dalam hal pemilihan umum Beberapa negara telah menerapkan sistem pemilihan umum dengan memanfaatkan teknologi informasi yang kemudian dikenal dengan sebutan e-voting.

Dilihat dari tingginya nilai kerahasiaan dan keamanan informasi dalam sebuah pemilu, maka dibutuhkan sistem pemilihan umum elektronik (e-voting) yang aman baik dari segi teknis maupun penyimpanan data.

Tugas akhir ini menggabungkan beberapa metode pengamanan data seperti penggunaan fungsi hash SHA256, digital signature, dan algoritma enkripsi asimetris RSA yang diterapkan pada basis data serta protokol pengiriman data dalam sistem e-voting. Fokus yang dikerjakan lebih terarah pada aplikasi "back-end" dari sistem yaitu dengan menggunakan web service sebagai aplikasi utama yang menjembatani pengiriman data suara baik dari TPS ke server maupun server ke server sekaligus menjamin keamanan, kerahasiaan, serta integritas data suara yang dikirimkan..

Dari uji coba metode pengamanan data di atas, didapatkan hasil bahwa sistem e-voting ini mampu menangani beberapa ancaman keamanan sistem utamanya pada basis data

dan pengiriman data dari serangan replay attack, packet sniffing, pemalsuan data suara, serta mampu menjamin privasi pemilih dan integritas data yang dapat dijadikan salah satu alternatif pengamanan basis data dalam berbagai jenis bentuk front-end sistem e-voting.

Kata kunci: Fungsi Hash SHA256 , Digital Signature, Enkripsi Asimetris RSA, E-Voting, Keamanan, Basis Data.

SECURITY SYSTEM DESIGN OF E-VOTING SYSTEM DATA STORAGE

Student's Name	:	ASTANDRO KOESRIPUTRANTO
Student's ID	:	5111100017
Department	:	Teknik Informatika FTIF-ITS
First Advisor	:	Prof. Ir. Supeno Djanali, M.Sc, Ph.D.
Second Advisor	:	Baskoro Adi Pratomo, S.Kom, M.Kom.

Abstract

Nowadays, information technology development finally reaches politics field for example information technology in government election. Some of major country have been using an election system applying information technology in its core known as e-voting.

In an election, the needs of security and secrecy is really high. We can conclude that we need an e-voting system that can provide both security in technics and the data storage.

This theses uses some kind of data security methods such as hash function SHA256, Digital Signature, and Asymmetric Encryption Algorithm RS. Those methods are applied at the database of e-vote system and the protocol used for sending data in the entire system. The focus of this theses is the "back-end" application of the entire e-vote system. Web services used as the middleware to send data between TPS to server or server to server also assuring the secrecy, security, and integrity of data sent.

From the tests of data security methods in e-vote system, the results say that this design can offers a good security against some security threat. The threat is the ones that is focused on database and sending data. Examples of the threats are replay attack, man in the middle attack, fake voting data. The test also conclude that the system can provide the privacy of voter and data

integrity. This theses be used as an alternative way to securing database on some kind e-vote system front-end.

Keywords: Hash Function SHA256, Digital Signature, Asymetric Encryption RSA, E-Voting, Security, Database.

KATA PENGANTAR

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **"RANCANG BANGUN SISTEM KEAMANAN PENYIMPANAN DATA PADA SISTEM E-VOTING".**

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan Tugas Akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan Tugas Akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Bapak, Ibu, Adek, serta saudara kembar saya Andri yang telah memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan Tugas Akhir ini.
3. Bapak Prof. Ir. Supeno Djanali, M.Sc, Ph.D. selaku pembimbing I yang telah membantu, membimbing, dan memotivasi penulis dalam menyelesaikan Tugas Akhir ini dengan sabar.
4. Bapak Baskoro Adi Pratomo, S.Kom, M.Kom. selaku pembimbing II yang juga telah membantu, membimbing, dan memotivasi kepada penulis dalam mengerjakan Tugas Akhir ini.

5. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Teman-teman setopik tugas akhir e-voting : Puguh, Ishom, dan Danang yang telah memberikan motivasi dan masukan-masukan serta bantuan berharga dalam diskusi dan pengerjaan tugas akhir ini.
7. Teman-teman seperjuangan di laboratorium Algotitma dan Pemrograman: Mahen, Yunus, Bustan, dan Novandi yang telah berjuang bersama penulis untuk menyelesaikan tugas akhir.
8. Rekan satu tim Informatics Basketball ITS yang senantiasa memberikan semangat dan dukungan dan hiburan ketika bermain basket di sela-sela pengerjaan tugas akhir ini.
9. Serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juli 2015

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR KODE SUMBER

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Penyimpanan data pada sistem pemungutan suara secara digital/elektronik (e-voting) menjadi salah satu hal yang sangat krusial dalam implementasi nyata. Hal ini dikarenakan penyimpanan data merupakan kunci untuk permasalahan privasi dari pemilih, keaslian data, dan kebenaran jumlah suara dalam pemungutan suara. Privasi dari pemilih yang dimaksud adalah bahwa tidak ada seorangpun yang boleh mengetahui calon mana yang dipilih oleh pemilih. Namun, suara tersebut juga harus dapat dipastikan keasliannya bahwa suara tersebut benar-benar berasal dari pemilih yang sah. Jumlah suara juga perlu dijaga kebenarannya untuk mencegah kecurangan seperti suara ganda atau penambahan suara untuk salah satu calon. Oleh karena itu, desain basis data yang aman dibutuhkan untuk sistem e-voting.

Untuk menangani masalah privasi dalam hal ini diperlukan suatu metode penyimpanan data yang melibatkan identitas pemilih pada basis data e-voting namun harus dipastikan bahwa tidak ada pihak manapun yang dapat memetakan bahwa pemilih A memilih calon x. Oleh karena itu untuk mengatasi masalah tersebut digunakanlah *hash function* pada identitas pemilih.

Keaslian atau integritas data suara yang disimpan juga perlu dipastikan untuk menghindari adanya suara palsu atau suara dari pemilih yang tidak sah. Untuk memastikan keaslian data suara yang dikirim, maka digunakan *digital signature* yang dapat memastikan keaslian data yang dikirimkan.

Yang terakhir adalah masalah penyimpanan data pada basis data. Dalam kasus ini yang akan disimpan identitas pemilih bersama dengan informasi TPS tempat pemilih melakukan pemungutan suara serta data calon yang dipilih. Untuk menjaga kerahasiaan data suara, maka dilakukan enkripsi terhadap data

calon yang dipilih. Untuk mengurangi beban terhadap server jika data langsung dikirimkan ke server pusat maka data-data tersebut terlebih dahulu disimpan di server lokal. Data yang dikirimkan dari server lokal ke server pusat nantinya juga harus dipastikan tidak mengalami perubahan. Untuk mengatasi hal tersebut maka akan digunakan pula *digital signature*. Enkripsi yang digunakan adalah RSA *Encryption* dengan memanfaatkan *public encryption, private decryption* dan *public decryption* yang digunakan untuk dekripsi data suara dan *digital signature* pada data suara.

Dengan menerapkan *hash function, digital signature,* dan enkripsi pada data dan sistem e-voting ini, diharapkan mampu menghasilkan sebuah sistem e-voting yang aman dan terjamin privasi pemilih, keaslian data, dan kebenaran hasil penghitungan suaranya.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membangun sistem e-voting yang menjaga privasi pemilih?
2. Bagaimana memastikan integritas data yang dikirimkan dalam sistem e-voting?
3. Bagaimana membangun sistem penyimpanan data yang aman untuk sistem e-voting?
4. Bagaimana menjaga kerahasiaan data suara sistem e-voting?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Aplikasi ini berbasis desktop dan web dengan bahasa pemrograman C# dengan framework .Net dan database MySQL.
2. Data yang disimpan dalam basis data adalah identitas pemilih (ID), nomor ID TPS, serta data calon yang dipilih

dengan tambahan kolom lainnya untuk menunjang keamanan sistem.
3. Pemilih dalam sistem ini diasumsikan telah terautentikasi.
4. Pemilihan umum yang digunakan sebagi acuan adalah pemilu legislatif di indonesia terkhususkan pada pemilihan calon DPRD tingkat Kota di Surabaya.
5. Aturan sistem E-voting yang dibuat mengikuti aturan pemilihan umum pada poin 4 untuk periode sebelumnya.

1.4 Tujuan

Tugas Akhir ini mempunyai beberapa tujuan, yaitu sebagai berikut:

1. Membangun sistem penyimpanan data yang aman pada e-voting dan dapat menjaga privasi pemilih serta integritas data yang disimpan.
2. Memberikan alternatif rancangan sistem penyimpanan basis data yang aman untuk berbagai macam sistem e-voting.

1.5 Metodologi

Tahap yang dilakukan untuk menyelesaikan Tugas Akhir ini adalah sebagai berikut:

1. **Penyusunan proposal tugas akhir**

   Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

2. **Studi literatur**

   Pada studi literatur ini, dipelajari sejumlah referensi yang diperlukan dalam perancangan sistem yaitu mengenai Hash Function, Digital Signature, RSA *Encryption, Web Service*, IIS Server, ASP .NET MVC 4 dengan Razor, serta koneksi basis data MySQL dengan .Net.

3. **Implementasi**

   Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. **Uji Coba dan evaluasi**

   Pada tahapan ini dilakukan uji coba pada data yang telah dikumpulkan. Pengujian dan evaluasi akan dilakukan dengan menggunakan bahasa Java. Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data dan program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

5. **Penyusunan Buku Tugas Akhir**

   Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.6 Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan

Bab pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan Tugas Akhir.

2. Bab II. Tinjauan Pustaka

   Bab tinjauan pustakan berisi penjelasan mengenai dasar teori yang mendukung pengerjaan Tugas Akhir.

3. Bab III. Analisis dan Perancangan

   Bab analisis dan perancangan berisi penjelasan mengenai analisis kebutuhan, perancangan sistem dan perangkat yang digunakan dalam pengerjaan Tugas Akhir serta urutan pelaksanaan proses.

4. Bab IV. Implementasi

   Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *code* yang digunakan untuk proses implementasi.

5. Bab V. Uji Coba dan Evaluasi

   Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

6. Bab VI. Kesimpulan dan Saran

   Bab kesimpulan dan saran berisi kesimpulan hasil penelitian. Selain itu, bagian ini berisi saran untuk pengerjaan lebih lanjut atau permasalahan yang dialami dalam proses pengerjaan Tugas Akhir.

BAB II
TINJAUAN PUSTAKA

Bab tinjauan pustaka berisi penjelasan teori yang berkaitan dengan implementasi perangkat lunak. Penjelasan tersebut bertujuan untuk memberikan gambaran mengenai sistem yang akan dibangun dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Hash Function

Hash function atau fungsi hash merupakan fungsi matematik yang dapat melakukan transformasi dari suatu nilai input dengan ukuran yang berbeda beda ke bentuk string dengan ukuran yang tetap. Hasil dari proses transformasi oleh fungsi hash disebut *hash value*, *hash sums*, *hash code*, atau hanya hash saja. Fungsi hash untuk data x dapat dinotasikan dengan H(x).

Properti dasar dari sebuah fungsi hash adalah :

1. Input data yang dengan panjang yang dapat beragam
2. Output data dengan panjang yang tetap,
3. H(x) cukup mudah untuk dihitung untuk apapun nilai x ,
4. H(x) merupakan "énkripsi" searah.
5. H(x) *collision-free*.

Jika terdapat suatu pesan x, yang kemudian dihitung nilai H(x), dan memungkinkan untuk dicari nilai y yang mana nilai H(x) = H(y) maka fungsi hash tersebut disebut *weak collission-free*. Sebaliknya, fungsi hash yang baik adalah dimana tidak memungkinkan untuk mencari suatu nilai x dan y di mana nilai H(x) = H(y), dan x!=y. Contoh fungsi hash yang banyak dikenal dan digunakan adalah MD2, MD5, dan SHA.

Salah satu contoh algoritma hash yang aman dan sering digunakan adalah SHA. Terdapat beberapa macam algoritma SHA yang sering digunakan yang dibedakan berdasarkan panjang bit *message digest*-nya. Algoritma SHA-256 merupakan salah satu algoritma fungsi *hash* dengan panjang *digest* 256 bit. Setiap pesan

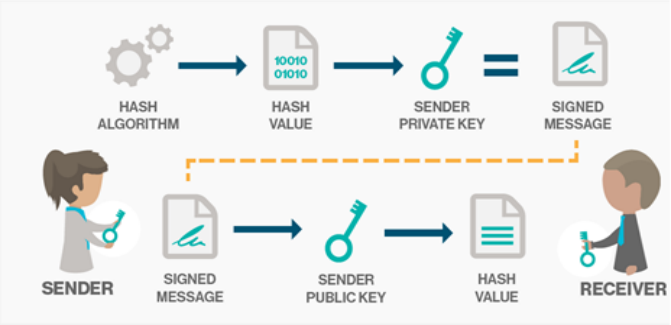
diproses dengan 512 = 16 x 32 bit *blocks* yang masing-masing terdiri dari 64 *round*. Operasi dasar yang digunakan dalam algoritma ini adalah *boolean* AND, OR, dan XOR. SHA-256 dianggap lebih aman dibanding SHA-1 karena tingkat kompleksitas yang lebih tinggi dan *message digest* yang dihasilkan lebih panjang. Sedangkan pada kasus algoritma SHA-1, terdapat hash collision. Pada SHA-1 kelemahannya terletak pada kurangnya kompleksitas algoritma.

2.2 *Digital Signature*

Digital signature atau tanda tangan digital merupakan metode yang digunakan untuk autentikasi dan atau untuk mengecek integritas data dalam proses pengiriman pesan. Serupa dengan tanda tangan/stempel/segel namun secara digital, tapi digital siganture menawarkan keamanan yang lebih dalam, tanda tangan digital dimaksudkan untuk memecahkan masalah gangguan dan peniruan dalam komunikasi digital. Tanda tangan digital dapat memberikan jaminan tambahan bukti asal, identitas dan status dokumen elektronik, transaksi atau pesan.

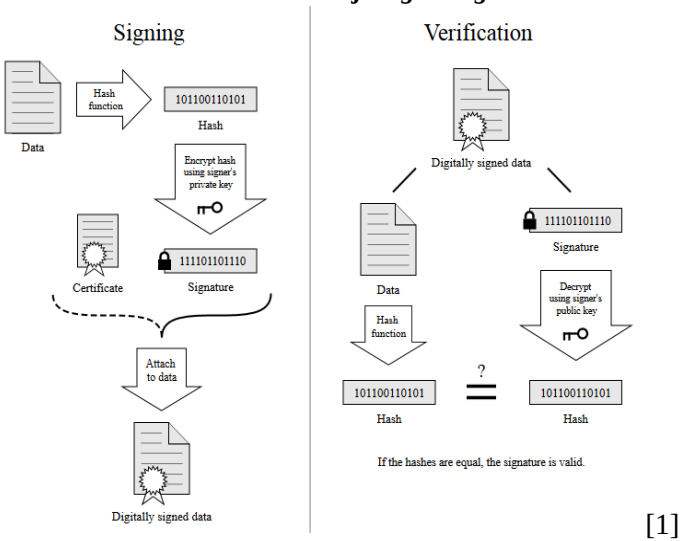
Di banyak negara, termasuk Amerika Serikat, tanda tangan digital memiliki makna hukum yang sama dengan bentuk-bentuk yang lebih tradisional semacam dokumen yang ditandatangani. Kantor Percetakan Pemerintah Amerika Serikat juga menerbitkan versi elektronik dari anggaran, hukum publik dan swasta, dan tagihan kongres dengan tanda tangan digital.

Digital signature bekerja berdasarkan kriptografi asimetris. Menggunakan algoritma enkripsi asimetris seperti RSA yang memiliki 2 macam key untuk proses enkripsinya yaitu public key da private key. Private key inilah yang akan digunakan untuk mengenkripsi hash value dari data yang akan dikirim. Proses penandatanganan digital untuk memastikan integritas pesan dapat dilihat pada gambar 2.1.



[1]

Gambar 0.1. Alur kerja *digital signature*



[1]

Gambar 0.2. Proses verifikasi dengan *digital signature*

2.3 Algoritma Enkripsi RSA

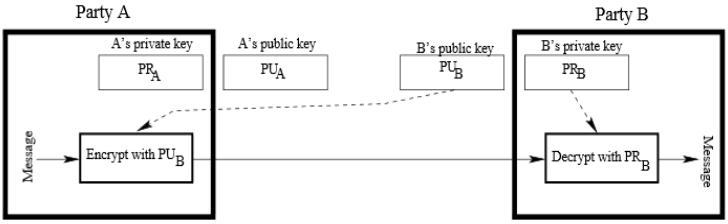
RSA merupakan teknik kriptografi untuk enkripsi dengan public key, dan banyak digunakan untuk mengamankan data yang sensitif terutama ketika akan dikirimkan melalui jaringan. RSA pertama kali dideskripsikan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman dari Massachusetts Institute of Technology.

Teknik kriptografi ini menggunakan dua macam kunci untuk enkripsinya. Dikenal dengan enkripsi asimetris. Dua macam kunci yang digunakan yaitu *public* dan *private* secara nilai berbeda namun saling berhubungan secara matematis. Kedua kunci tersebut dapat digunakan untuk proses enkripsi yang mana kunci pasangan yang digunakan untuk mengenkripsi pesan sebelumnya yang dapat digunakan untuk melakukan dekripsi. Atribut inilah yang membuat RSA banyak digunakan. Hal ini karena algoritma ini mampu dimanfaatkan dalam metode untuk menjaga kerahasiaan dan integritas data.

Banyak protokol yang menggunakan RSA untuk enkripsinya seperti SSH, OpenPGP, S/MIME, dan SSL/TLS. Selain itu banyak pula digunakan pada program aplikasi, web browser, dan aplikasi lain yang membutuhkan koneksi yang aman melalui jaringan dan atau memerlukan proses validasi seperti *digital signature*.

2.3.1 *Public Encryption* dan *Private Decryption*

Ketika dibutuhkan kerahasiaan (*confidentiality*) data dibutuhkan, maka jenis enkripsi dan dekripsi yang digunakan dalam algoritma RSA adalah enkripsi dengan *public key* penerima/tujuan dan dekripsi dengan *private key* penerima. Tujuan dari metode ini adalah supaya yang berhak untuk membuka atau membaca data yang dirahasiakan hanyalah pihak yang memiliki *public key* yang bersangkutan. Mekanisme enkripsi dan dekripsi ini dapat dilihat pada gambar 2.3.

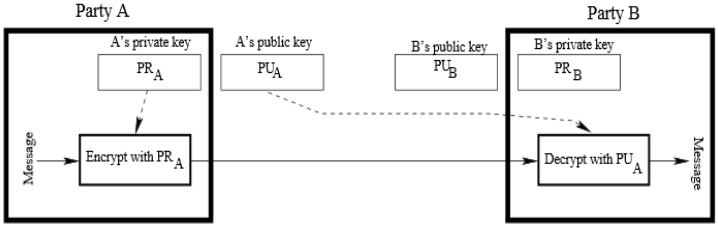


[2]

Gambar 0.3. Mekanisme *public encryption* dan *private decryption*

2.3.2 *Private Encryption* dan *Private Decryption*

Jika yang dibutuhkan hanyalah autentikasi saja, yaitu untuk memastikan bahwa pengirim merupakan "orang" yang tepat, maka mekanisme dalam RSA yang digunakan adalah *private encryption* dan *public decryption*. Mekanisme ini membutuhkan *private key* dari pengirim yang mana hanya pengirim sendiri yang tahu nilainya. Hasil dari enkripsi tersebut nantinya akan didekripsi dengan menggunakan public key milik pengirim, jika data dapat dibuka/didekripsi, maka dapat dipastikan bahwa pengirim adalah pengirim yang sesungguhnya. Mekanisme ini dapat dilihat pada gambar 2.4.



[2]

Gambar 0.4. Mekanisme *private encryption* dan *public decryption*

2.4 E-Voting

E-Voting adalah suatu sistem pemilihan umum yang memungkinkan voter atau pemilih untuk menyimpan data pemilihan yang bersifat aman dan rahasia secara elektronik. Kelebihan dari e-voting dibanding pemilu konvensional pada umumnya antara lain meminimalisasi kesalahan yang biasa disebabkan adanya *human error*. Selain itu, jika dilaksanakan dengan teknis dan efisiensi yang tepat, e-voting akan memakan biaya yang lebih murah serta cepat dalam pemrosesan atau penghitungan suara dalam pemilu. Prinsip utama e-voting adalah bahwa proses voting harus semirip mungkin dengan regular voting begitu pula dari segi keamanan, minimal harus seaman regular

voting. Oleh karena itu, e-voting haruslah seragam dan bersifat rahasia di mana hanya orang-orang yang berhak yang diperbolehkan mengakses sistem tersebut. [3]

2.5 *Web Service*

Web service adalah sistem software yang dirancang untuk mendukung interopabilitas mesin-ke-mesin yang dapat berinteraksi melalui jaringan. Web service memiliki antarmuka yang dijelaskan dalam format mesin-processable (khusus WSDL). Sistem lain berinteraksi dengan Web service dalam cara ditentukan oleh deskripsi dengan menggunakan pesan SOAP, biasanya disampaikan menggunakan HTTP dengan serialisasi XML dalam hubungannya dengan Web lainnya yang terkait standar.

XML Web Services dapat di definisikan sebagai aplikasi yang diakses oleh aplikasi yang lain. Mungkin orang berpendapat itu semacam web site, tetapi itu bukan demikian. Ada perbedaan – perbedaan yang membedakan dengan web site.

Perbedaan tersebut antara lain :

Web Site :

1. Memiliki web interface
2. Dibuat untuk ber interaksi langsung dengan user
3. Dibuat untuk bekerja pada web browser.

Web Services :

1. Tidak memiliki interface yang bagus
2. Dibuat untuk ber interaksi langsung dengan applikasi yang lain baik beda OS / Konsep sekalipun.
3. Dibuat untuk bekerja pada semua tipe client applikasi / perangkat device

Beberapa karakteristik dari web service adalah:

1. Message-based
2. Standards-based
3. Programming language independent
4. Platform-neutral

Beberapa key standard didalam web service adalah: XML, SOAP, WSDL and UDDI.

SOAP (Simple Object Access Protocol) adalah sebuah XML-based mark-up language untuk pergantian pesan diantara aplikasi-aplikasi. SOAP berguna seperti sebuah amplop yang digunakan untuk pertukaran data object didalam network. SOAP mendefinisikan empat aspek didalam komunikasi: Message envelope, Encoding, RPC call convention, dan bagaimana menyatukan sebuah message didalam protokol transport.

Sebuah SOAP message terdiri dari SOAP Envelop dan bisa terdiri dari attachments atau tidak memiliki attachment. SOAP envelop tersusun dari SOAP header dan SOAP body, sedangkan SOAP attachment membolehkan non-XML data untuk dimasukkan kedalam SOAP message, di-encoded, dan diletakkan kedalam SOAP message dengan menggunakan MIME-multipart.

WSDL (Web Services Description Language) adalah sebuah XML-based language untuk mendeskripsikan XML. WSDL menyediakan service atau layanan yang mendeskripsikan service request dengan menggunakan protokol-protokol yang berbeda dan juga encoding. WSDL memfasilitasi komunikasi antar aplikasi. WSDL akan mendeskripsikan apa yang akan dilakukan oleh web service, bagaimana menemukannya dan bagaimana untuk mengoperasikannya.

2.6 HTTPS

HTTPS adalah penggabungan antara Hypertext Transfer Protocol (HTTP) dengan SSL / TLS protokol. Semua komunikasi yang dilakukan melalui HTTPS akan dienkripsi dengan tujuan untuk keamanan saat terjadi transaksi data di internet. Biasanya para hacker atau peretas internet yang biasa menggunakan tool WireShak sangat mudah untuk mencuri data dari klien yang terhubung ke internet dengan menggunakan HTTP. Berbeda dengan HTTPS, semua akses akan sangat sulit diproses dan menangkap data oleh para pencuri website.

Koneksi internet dengan menggunakan HTTPS lebih aman dibandingkan menggunakan HTTP. Untuk itu banyak perbankan yang membangun layanan onlinenya dengan menggunakan format HTTPS. Hal ini bertujuan untuk menjamin keamanan para nasabah ketika melakukan transaksi online. Tidak hanya transaksi online, di pemerintahan atau sebuah lembaga juga memproteksi keamanan dokumen-dokumen pentingnya dengan menggunakan HTTPS. Dalam pengembangan, biasanya digunakan HTTPS dengan *Self Signed Certificate* yang merupakan sertifikat digital yang digunakan untuk mengirim data pada protokol https, namun tanpa ada adanya verifikasi oleh pihak ketiga yang terpercaya yaitu *Certificate Authority* (CA) seperti Verisign. Meskipun tanpa adanya verifikasi tersebut, sertifikat ini juga tetap menyediakan fitur keamanan untuk enkripsi data yang dikirimkan untuk mencegah data dibaca oleh pihak yang tidak berhak. *Self Signed Certificate,* dapat digunakan untuk testing server untuk pengujian keamanan.

2.7 *Replay Attack*

Merupakan serangan terhadap keamanan jaringan dengan mengirimkan ulang data atau informasi yang sah yang telah direkam sebelumnya yang biasanya digunakan untuk memperoleh akses ke basis data atau aplikasi yang membutuhkan otorisasi khusus. Jenis serangan ini dapat diproteksi dengan menggunakan *nonce* atau jika dikaitkan dengan basis data, dapat menggunakan *identifier* yang unik untuk setiap baris data pada tabel.

2.8 *Packet Sniffing*

Istilah packet scanning atau biasa disebut packet sniffing adalah proses pemindaian paket - paket data di dalam jaringan menggunakan software penyadap paket atau (packet sniffer).

Sniffer paket dapat dimanfaatkan untuk hal-hal berikut:

1. Mengatasi permasalahan pada jaringan komputer.
2. Mendeteksi adanya penyelundup dalam jaringan (Network Intusion).

3. Memonitor penggunaan jaringan dan menyaring isi isi tertentu.
4. Memata-matai pengguna jaringan lain dan mengumpulkan informasi pribadi yang dimilikanya (misalkan password).
5. Dapat digunakan untuk Reverse Engineer pada jaringan.

Tools atau software yang banyak digunakan untuk proses sniffing misalnya Cain & Abel, Wireshark, Etherdetect , dll.

2.9 SQL *Injection*

SQL injection adalah serangan yang memanfaatkan kelalaian dari website yang mengijinkan user untuk menginputkan data tertentu tanpa melakukan filter terhadap malicious character. Inputan tersebut biasanya di masukan pada box search atau bagian-bagian tertentu dari website yang berinteraksi dengan database SQL dari situs tersebut. Perintah yang dimasukan para attacker biasanya adalah sebuah data yang mengandung link tertentu yang mengarahkan para korban ke website khusus yang digunakan para attacker untuk mengambil data pribadi korban.

Contoh sintak SQL dalam PHP :

1. *$SQL = "select * from login where username ="$username" and password = „$password"";* , (dari GET atau POST variable).
2. isikan password dengan string ‟ or ‟‟‟ = ‟.
3. Hasilnya maka SQL akan seperti ini = *"select * from login where username = "$username" and password="pass" or „='";* , (dengan SQL ini hasil selection akan selalu TRUE).

BAB III
DESAIN DAN PERANCANGAN

Pada Bab 3 akan dijelaskan mengenai perancangan sistem perangkat lunak untuk mencapai tujuan dari Tugas Akhir. Perancangan yang akan dijelaskan pada bab ini meliputi perancangan data, perancangan proses dan perancangan antar muka. Selain itu akan dijelaskan juga desain metode secara umum pada sistem.

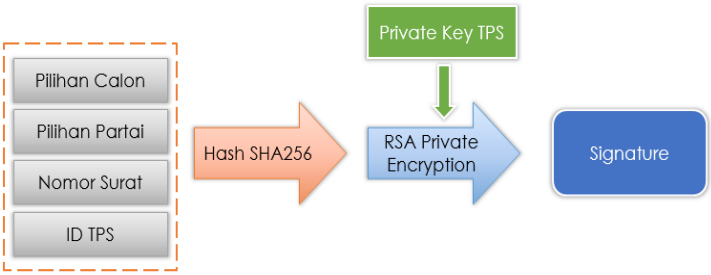
3.1 Desain Pengamanan Basis Data

Desain basis data yang digunakan untuk sistem e-voting ini secara keseluruhan mulai dari basis data yang digunakan pada TPS hingga server pusat adalah sama sebagaimana dapat dilihat pada halaman LAMPIRAN gambar A.2. Untuk mengamankan basis data tersebut, maka diperlukan adanya modifikasi pada beberapa tabel yaitu tabel SUARA, dan tabel TPS. Untuk tabel TPS, hanya perlu ditambahkan satu buah atribut baru *Public Key* untuk menyimpan kunci public dari masing-masing TPS. Selanjutnya, pengamanan pada tabel suara dapat dilihat pada tabel 3.1.

Tabel 0.1. Tabel Pengamanan Basis Data Sistem E-voting

Atribut	Pengamanan	Keterangan
ID Suara	Hash dengan SHA256	H (pilihan calon || pilihan partai || waktu pilih || noSurat || idTPS)
Pilihan Calon	Enkripsi dengan RSA *Public Encryption*	E (Pil. Calon, $K_{public[KPU]}$)

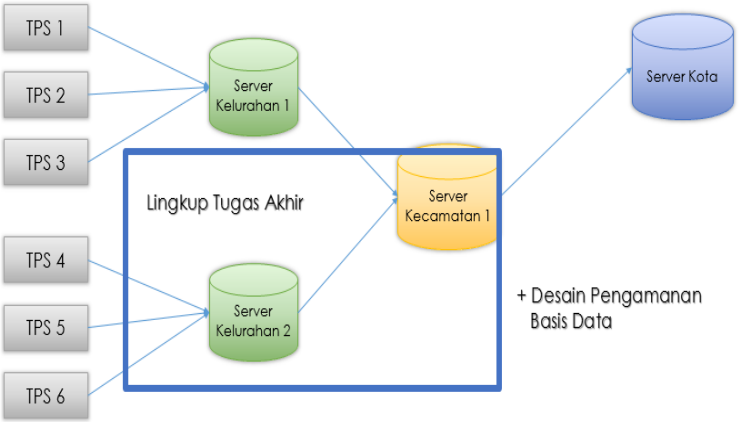
Atribut	Pengamanan	Keterangan
Pilihan Partai	Enkripsi dengan RSA *Public Encryption*	E (Pil. Partai, $K_{public[KPU]}$)
No. Surat	-	-
Hash Suara	Hash dengan SHA256	Untuk kode QR pada kertas suara
Signature	Hash dengan SHA256 dan Enkripsi dengan RSA *Private Encryption*	Untuk validasi keaslian data suara dengan konsep *Digital Signature*.
ID TPS	-	Nomor TPS sesuai dengan kode wilayah TPS
Sender Code	-	Kode pengirim yang digunakan untuk query salah satu fungsi pada *web service*



Gambar 0.1. Mekanisme pembuatan *Signature* pada tabel suara

3.2 Desain Arsitektur Sistem

Arsitektur sistem e-voting pada tugas akhir ini terdiri dari beberapa TPS yang masing-masing memiliki database lokal, server kelurahan, dan juga server kecamatan. Pada setiap server terdapat aplikasi *web service* yang akan digunakan untuk menjalankan fungsional sistem. Gambaran arsitektur sistem e-voting ini dapat dilihat pada gambar 3.2.



Gambar 0.2. Arsitektur sistem e-voting

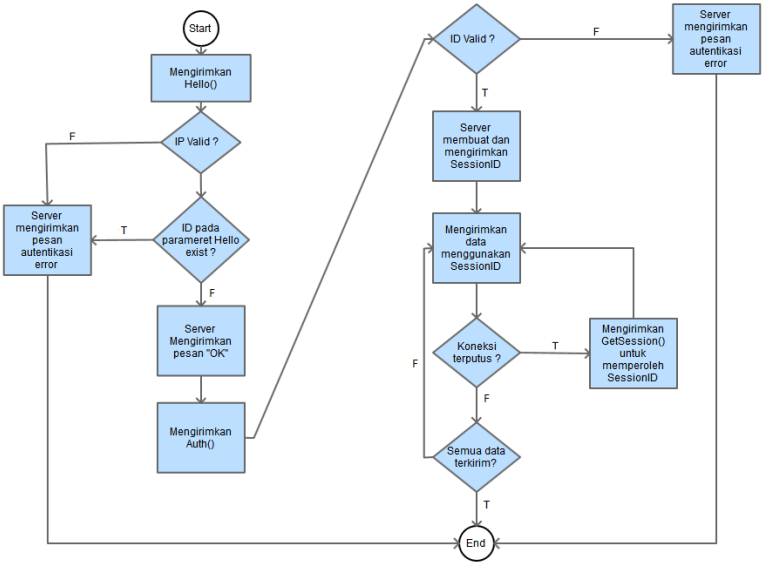
3.3 Desain *Web Service*

Aplikasi *web service* pada tugas akhir ini digunakan untuk proses login admin pada TPS, login admin pada server, serta mengirimkan data dari TPS ke server maupun server ke server secara aman. Untuk memenuhi fungsionalitas tersebut, maka ditentukan beberapa *web method* pada *web service* yaitu :

1. Login
2. Hello

3. Auth
4. SendTotalSuaraPartai
5. SendTotalSuaraCalon
6. SendData
7. GetSession
8. GetAmountSent
9. Close

Kemudian untuk alur pengiriman data menggunakan *web service* dapat dilihat pada gambar 3.3.



Gambar 0.3. Alur pengiriman data menggunakan *web service*

3.3.1 Parameter pada *Web Service*

Terdapat 9 buah *web method* pada *web service* dalam tugas akhir ini. Detail parameter serta *return value* dari setiap parameter dapat dilihat pada tabel 3.2.

Tabel 0.2. Detail Parameter *Web Method* pada *Web Service*

Nama Fungsi	Parameter	*Return Value*
Login	Username, Password	True/False
Hello	H(ID_TPS)	"OK", "Authentication Error"
Auth	H(ID_TPS), E(ID_TPS, $K_{privat[TPS]}$)	SessionID(1), "Authentication Error"
SendTotalSuaraPartai	SessionID, data total suara partai	-
SendTotalSuaraCalon	SessionID, data total suara calon	-
Send	SessionID, Data(2)	-
GetSession	H(ID_TPS)	SessionID
GetAmountSent	SenderCode	AmountSent(3)
Close	H(ID_TPS), SessionID	"BYE"

Keterangan tambahan :

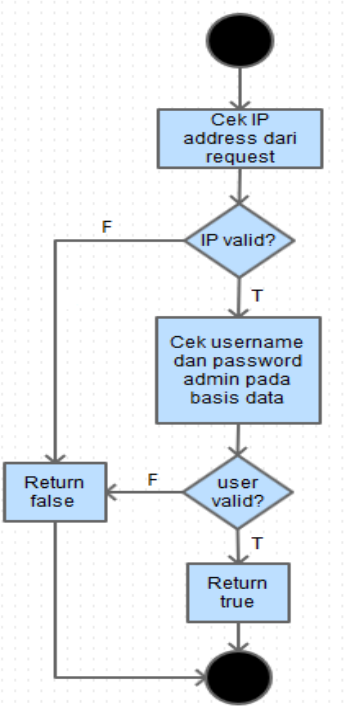
1. SessionID = H(ID_TPS + DateTime.Month + DateTime.Day).
2. Data = Data suara sesuai dengan atribut dalam tabel suara pada basis data e-voting.

3. AmountSent = jumlah data yang telah terkirim oleh TPS dengan SenderCode pada parameter.

3.3.2 Proses pada *Web Method*

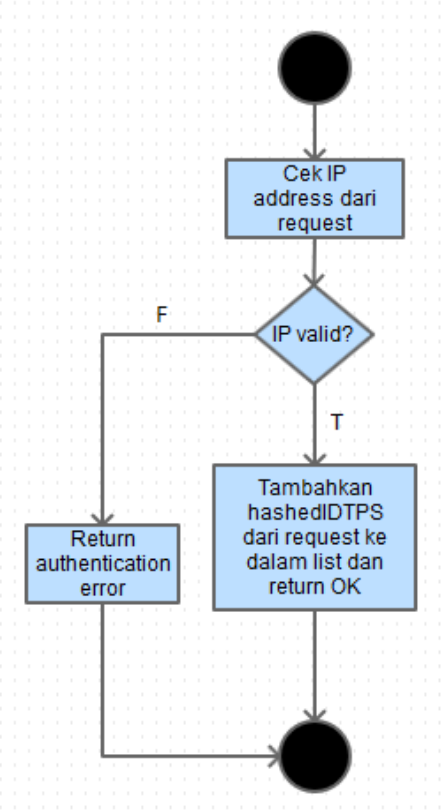
Masing-masing *web method* pada *web service* memiliki proses tersendiri baik secara algoritma maupun akses ke basis data. Detail proses pada masing-masing *web method* antara lain :

1. Login.
   Digunakan untuk proses login admin. Alur *method* ini dapat dilihat pada gambar 3.4.



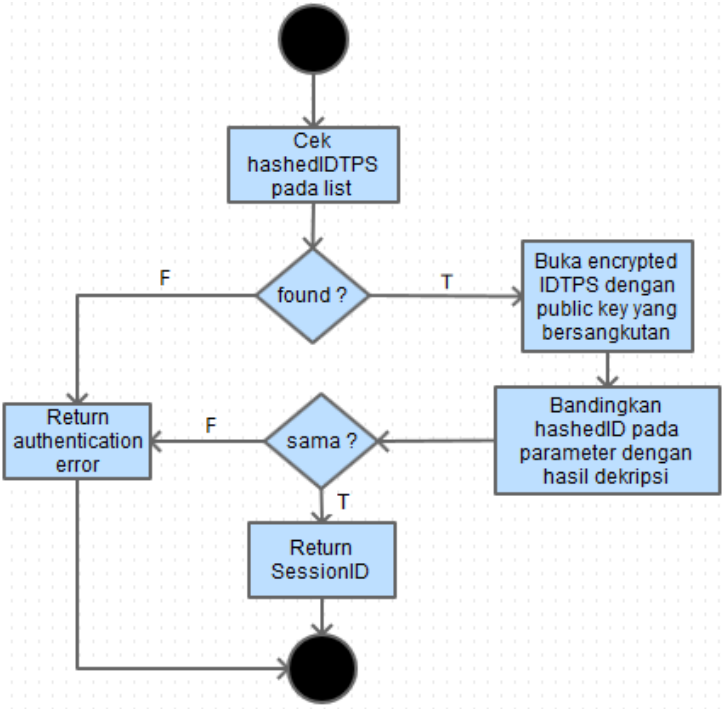
Gambar 0.4. Alur method login pada *web service*

2. Hello.
   Digunakan untuk pengecekan awal "pengirim" yang akan mengirim data mealui *web service.* Alur *method* ini dapat dilihat pada gambar 3.5.



Gambar 0.5. Alur method hello pada *web service*

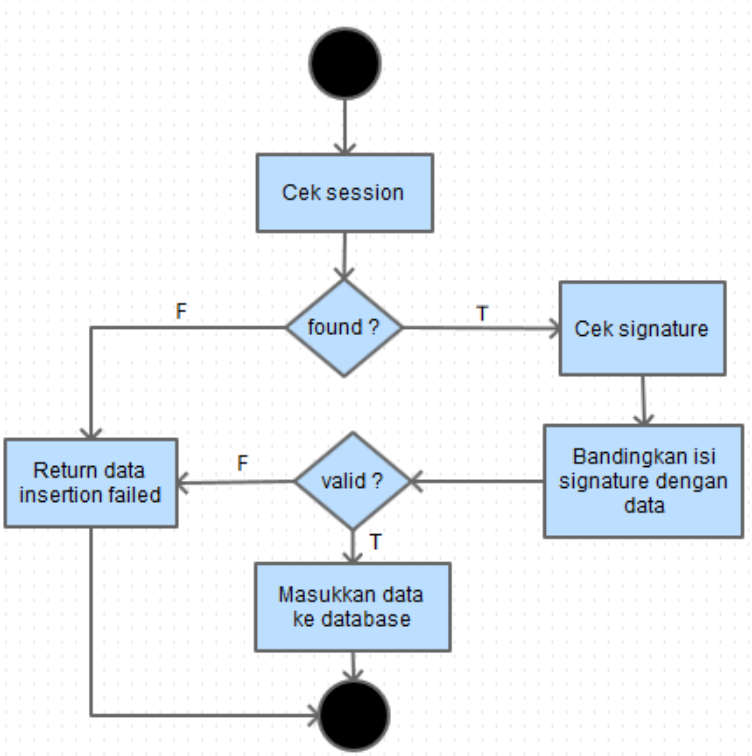
3. Auth.
   Digunakan untuk mengautentikasi *consumer web service*. Alur *method* ini dapat dilihat pada gambar 3.6.



Gambar 0.6. Alur method auth pada *web service*

4. SendTotalSuaraPartai.
   Digunakan untuk mengirimkan nilai perolehan suara sementara masing-masing partai. Method ini bekerja hanya dengan menyimpan data yang diperoleh dari parameter ke dalam basis data pada *web service*.
5. SendTotalSuaraCalon.
   Digunakan untuk mengirimkan nilai perolehan suara sementara masing-masing calon. Method ini bekerja hanya dengan menyimpan data yang diperoleh dari parameter ke dalam basis data pada *web service*.

6. Send.
   Digunakan untuk mengirimkan data suara. Alur method ini dapat dilihat pada gambar 3.7.



Gambar 0.7. Alur method send pada *web service*

7. GetSession.
   Digunakan untuk memperoleh session kembali oleh *consumer* yang sudah pernah melalui tahap autentikasi sebelumnya dengan hanya perlu mengirimkan *request* dengan parameter *hashedIDTPS*.

8. GetAmountSent.
   Digunakan untuk memperoleh jumlah data yang telah dikirim. *Method* ini bekerja dengan melakukan *query* ke basis data jumlah data yang telah dikirim dengan parameter *senderCode*.
9. Close.
   Digunakan untuk mengakhiri proses pengiriman data. Method ini bekerja dengan menghapus *hashedIDTPS* dan *SesssionID* yang ada dalam *list* pada variabel statis *web service* sesuai dengan nilai yang terdapat pada parameter.

3.4 Desain Aplikasi *Web Reporting*

Aplikasi web ini merupakan aplikasi yang terdapat pada tiap server di mana terdapat *web service* sistem e-voting. Fitur utama aplikasi ini antara lain :

1. Melihat rekapitulasi sementara hasil e-voting.
   Halaman yang menampilkan hasil perolehan suara sementara dari partai maupun calon peserta pemilu.
2. Melihat data suara yang masuk ke server.
   Halaman yang menampilkan data suara yang masuk ke basis data milik server yang dikirimkan melalui *web service*. Data ini ditampilkan sesuai dengan kondisi masih terenkripsi.
3. Melihat dan menambahkan tanda tangan panitia/saksi pada TPS/server.
   Satu halaman untuk menambahkan tanda tangan panitia dengan menginputkan informasi panitia beserta *private key* milik panitia tersebut. Halaman ini juga dapat menampilkan tanda tangan panitia yang ada pada TPS atau server tersebut.
4. Mengirimkan data suara yang diterima oleh server ke server selanjutnya.
   Tombol untuk mengirimkan seluruh data suara yang ada pada basis data server ke server selanjutnya dengan menggunakan *web service*.

BAB IV
IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

4.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi adalah IIS 8 yang diinstall pada komputer dengan sistem operasi windows 8 sebagai server. Detail spesifikasi lingkungan implementasi antara lain :

1. Menggunakan basis data MySQL 5.5 pada server.
2. Menggunakan phpmyadmin untuk manajemen basis data.
3. Terhubung dengan jaringan lokal ITS Tekni Informatika dengan alamat IP server 10.151.32.55
4. Menggunakan protokol https (*Self Signed Certificate*) untuk mengakses server dan phpmyadmin.

4.2 Implementasi Proses

Pada subbab ini akan dijelaskan implementasi setiap subbab yang terdapat pada bab sebelumnya yaitu bab perancangan program. Pada bagian implementasi ini juga akan dijelaskan mengenai fungsi-fungsi yang digunakan dalam program tugas akhir ini dan disertai dengan kode sumber masing-masing fungsi.

4.2.1 Implementasi Basis Data

Basis data sistem e-voting ada tugas akhir ini menggunakan MySQL dengan phpmyadmin untuk manajemen basis datanya. Struktur basis data pada sistem ini dibuat dengan menggunakan aplikasi Power Designer. Untuk pengamanan basis data, maka sesuai dengan yang telah dibahas pada bab perancangan bahwa akan terdapat 8 atribut pada tabel suara serta 1 buah tambahan atribut yaitu *public key* pada tabel TPS.

4.2.2 Implementasi *Web Service*

Web service untuk sistem e-voting ini diinstall pada komputer dengan sistem operasi windows 8, .NET framework 2.0 dan .NET framework 4.0. untuk mengamankan akses ke *web service* ini digunakan protokol https (*Self Signed Certificate*) pada IIS server. Selanjutnya, untuk mengakses atau mengonsumsi *web service* ini dapat dilakukan melalui jaringan lokal dengan alamat https://10.151.32.55/evoteservice/Service1.asmx yang berfungsi sebagai *web service* untuk server kelurahan dan https://10.151.32.55/evoteservice2/Service1.asmx yang berfungsi sebagai *web service* server kecamatan dalam sistem e-voting tugas akhir ini. Aplikasi *Web Service* ini dibuat dengan menggunakan template yang ada pada IDE Visual Studio 2012 dengan .NET Framework 2.0 yaitu ASP .NET Web Service Application. Configurasi basis data yang terhubung dengan masing-masing *web service* dapat ditemukan dan diubah pada direktori C:\inetpub\wwwroot\evoteservice\config\DBConfig.conf.

Selanjutnya akan disampaikan implementasi untuk masing-masing *web method* yang ada pada *web service*.

a. Implementasi *web method* Login.
   web method ini digunakan untuk login admin dengan mengirimkan dua buah parameter yaitu username dan password. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.1.

```
1.  Load config file
2.  var IP = ip address from request
3.  bool ipFound = false
4.  bool found = false
5.  if IP exist in listIPAddress
6.     ipFound = true
7.  String user = username fromDB
```

```
8.  String pass = password fromDB
9.  If user == username given && pass
    == password given
10.    Found = true
11. If found && ipFound
12.    Return true
13. Else
14.    Return false
```

Kode Sumber 0.1. Kode program *web method* Login

b. Implementasi *web method* Hello.
Web method ini digunakan untuk inisialisasi alur pengiriman data yang akan dilakukan. Pada tahap ini dilakukan pengecekan alamat IP dari calon pengirim data dan penyimpanan parameter *hashedIDTPS*. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.2.

```
15. Load config file
16. var IP = ip address from request
17. bool ipFound = false
18. bool found = false
19. if IP exist in listIPAddress
20.    Add hashedID to listID
21.    Return "OK"
22. Else
23.    Return "Authentication Error"
```

Kode Sumber 0.2. Kode program *web method* Hello

c. Implementasi *web method* Auth.
Web method ini digunakan untuk mengautentikasi pengirim yang menggunakan *web service*. Pada tahap ini dilakukan pengecekan alamat *hashedIDTPS* pada list dan juga membandingkan dengan hasil dekripsi dari parameter yang sebelumnya telah dienkripsi menggunakan RSA *Private*

encryption oleh pengirim. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.3.

```
24.  Bool found = false
25.  If hashedID exist in listID
26.     Found = true
27.  Bool valid = false
28.  String key = suitable public key
     from DB
29.  RSA.loadPublicKey(key)
30.  String msg = RSA.decrypt(encID)
31.  If msg equals hashedID
32.    Valid = true
33.  If found && valid
34.     Create sessionID
35.     Return sessionID
36.  Else
37.     Return "Authentication Error"
```

Kode Sumber 0.3. Kode program *web method* Auth

d. Implementasi *web method* SendTotalSuaraPartai.
Web method ini digunakan untuk mengirimkan data perolehan suara partai sementara. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.4.

```
38.  Bool found = false
39.  If sessionID exist in listSession
40.     Found = true
41.  If found
42.     Insert data into DB
43.  Else
44.    Return "Data insertion failed"
```

Kode Sumber 0.4 Kode program *web method* SendTotalSuaraPartai

e. Implementasi *web method* SendTotalSuaraCalon.
Web method ini digunakan untuk mengirimkan data perolehan suara calon sementara. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.5.

```
45.  Bool found = false
46.  If sessionID exist in listSession
47.     Found = true
48.  If found
49.     Insert data into DB
50.  Else
51.    Return "Data insertion failed"
52.  Bool found = false
```

Kode Sumber 0.5 Kode program *web method* SendTotalSuaraCalon

f. Implementasi *web method* Send.
Web method ini digunakan untuk mengirimkan data suara asli yang masuk ke basis data sistem e-voting. Pada tahap ini dilakukan pengecekan *signature* dari setiap data yang dikirim untuk memastikan keaslian data. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.6.

```
53.  Bool valid = false
54.  String publicKey = null
55.  String senderCode = null
56.  Bool found = false
57.  If sessionID exist in listSession
58.     Found = true
59.  If found
60.     PublicKey = public key from DB
61.    senderCode = H(senderIP)
62.     RSA.loadPublicKey(publicKey)
```

```
63.     RSA.decrypt(signature)
64.     If signature valid
65.         Valid = true
66.     If valid
67.         Insert data into DB
68.     Else
69.         Return "Insertion failed"
70. Else
71.     Return "Data insertion failed"
```

Kode Sumber 0.6. Kode program *web method* Send

g. Implementasi *web method* GetSession.
Web method ini digunakan untuk meminta session yang telah dimiliki oleh pengirim sebelumnya. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.7.

```
72. Bool found = false
73. If hashedID exist in listActiveID
74.     Return sessionID
75. Else
76.     Return "Session not found"
```

Kode Sumber 0.7 Kode program *web method* GetSession

h. Implementasi *web method* GetAmountSent.
Web method ini digunakan untuk mengetahui jumlah data yang telah dikirim oleh pengirim dengan parameter *SenderCode*. Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.8.

```
77. Int amountSent = 0
78. amountSent = query from DB with
    suitable senderCode
79. Return amountSent
```

Kode Sumber 0.8. Kode program *web method* GetAmountSent

i. Implementasi *web method* Close.
Web method ini digunakan untuk mengakhiri sesi pengiriman data. Pada tahap ini dilakukan penghapusan *SessionID* dan *hashedIDTPS* dari daftar ID yang aktif Implementasi dalam bentuk *pseudocode* dapat dilihat pada kode sumber 4.9.

```
80.  Remove sessionID from listSession
81.  Remove hashedID from listActiveID
82.  Return "BYE"
```

Kode Sumber 0.9. Kode program *web method* Close

4.2.3 Implementasi *Web Reporting*

Pada implementasi *web service* ini digunakan aplikasi web ASP .NET MVC 4 dengan Razor *view engine*. Aplikasi ini diinstall pada komputer server dan dapat diakses secara lokal melalui https://10.151.32.55/evotereport/. Konfigurasi basis data yang digunakan terletak pada direktori C:\inetpub\wwwroot\evotereport\config\DBConfig.conf. Hirarki implementasi aplikasi web MVC ini dapat dilihat pada halaman LAMPIRAN gambar A.1.

1. Implementasi fitur rekapitulasi sementara e-voting.
   Untuk mengimplementasikan fitur ini, dibuatlah model, *view,* dan *controller* sebagai berikut :
   a. Model = Data.cs
   b. View = Home.cshtml
   c. Controller = HomeController.cs
   Model yang ada digunakan untuk menangani seluruh proses yang berkaitan dengan basis data terutama untuk *query* informasi TPS, suara, serta perolehan suara sementarauntuk ditampilkan pada *view*. *View* yang dibuat akan menampilkan informasi berupa jumlah suara yang masuk, nama server, serta informasi perolehan sementara dari partai dan calon peserta pemilu.
2. Implementasi fitur lihat data suara yang masuk.

Untuk fitur ini, dibuat model, *view*, dan *controller* sebagai berikut:

a. Model = Data.cs
b. View = DataSuara.cshtml
c. Controller= DataSuaraController.cs

Model pada fitur ini juga digunakan untuk memproses seluruh *query* untuk data suara yang akan ditampilkan. Pada *view* akan ditampilkan seluruh baris data suara dengan kondisi masih terenkripsi. Pada bagian ini juga ditambahkan satu buah tombol "Buka Suara" untuk melihat informasi pilihan calon dan partai yang terenkripsi dengan menggunakan *private key* yang sesuai.

3. Implementasi fitur tanda tangan panitia/saksi.
   Model, *view*, dan *controller* yang digunakan untuk implementasi fitur ini antara lain:
   a. Model = Data.cs
   b. View = TtdPanitia.cshtml
   c. Controller = TtdPanitiaController.cs

   Model pada implementasi ini digunakan untuk proses *query* dan *insert* data dari dan ke dalam basis data yang terkait dengan informasi tanda tangan panitia e-voting. Pada bagian *view* digunakan sebuah *form method* POST untuk input tanda tangan panitia. Pada *form* tersebut terdapat beberapa form input untuk melengkapi informasi panitia seperti nama, ID, dll. Pada bagian *view* juga akan ditampilkan seluruh tanda tangan panitia pada TPS atau server tersebut dalam bentuk baris tabel.
4. Implementasi fitur kirim data suara.
   Fitur ini dibuat cukup dengan menambahkan menu *dropdown* pada bagian pojok kanan atas setiap halaman pada aplikasi *web reporting* dengan satu buah tombol "Send Data" di sebelah tombol "Logout". Tombol tersebut akan mengirimkan seluruh data suara yang ada pada basis data ke server selanjutnya dengan menggunakan *web service*.

BAB V
UJI COBA DAN EVALUASI

Bab uji coba dan evaluasi berisi mengenai hasil uji coba dan evaluasi terhadap perangkat lunak dari implementasi segmentasi lesi merah pada citra fundus mata berwarna menggunakan pendekatan metode morfologi.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menjelaskan lingkungan yang digunakan untuk menguji implementasi aplikasi pada sistem tugas akhir ini antara lain aplikasi *web service*, *dummy TPS*, dan *web reporting*. Lingkungan uji coba meliputi beberapa perangkat keras dan perangkat lunak yang terbagi menjadi 2 macam dijelaskan sebagai berikut:

1. Perangkat keras
   a. PC Server, Prosesor: Intel® Core™ i3-2120 CPU @ 3.30GHz, *Memory*(RAM): 4,00 GB, Tipe sistem: 64-bit sistem operasi.
   b. PC TPS, Prosesor: Intel® Core™ i3-2310 CPU @ 2.10GHz, *Memory*(RAM): 2,00 GB, Tipe sistem: 64-bit sistem operasi.
2. Perangkat lunak
   a. Sistem operasi PC Server: *Windows* 8.1 *Professional*.
   b. Sistem operasi PC Server: *Windows* 8 *Professional*.
   c. Perangkat pengembang: *Visual Studio* 2012.

5.2 Skenario Uji Fungsionalitas

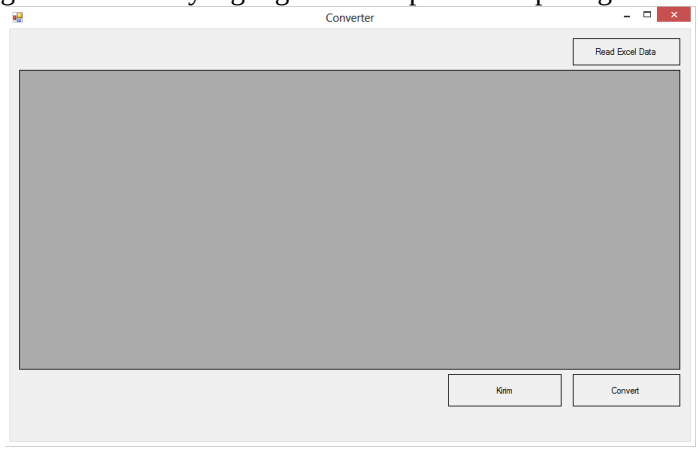
Uji coba ini dilakukan untuk mengetahui apakah fungsionalitas sistem pada tugas akhir ini berjalan dengan baik sesuai dengan studi kasus yang ditentukan. Dalam uji fungsionalitas ini terdapat 6 macam skenario pengujian. Skenario pengujian fungsionalitas tersebut antara lain :

1. Mengirimkan data suara hasil pemilihan dari TPS ke server kelurahan.

2. Mengirimkan data suara hasil pemilihan dari server kelurahan ke server kecamatan.
3. Meminta *session* untuk TPS yang sebelumnya telah memiliki *session* yang akan digunakan kembali untuk mengirimkan data.
4. Melakukan penandatanganan oleh panitia pada server.
5. Login pada TPS dengan memanfaatkan *web service.*
6. Login pada server dengan memanfaatkan *web sevice.*

5.2.1 Skenario Uji Fungsionalitas 1

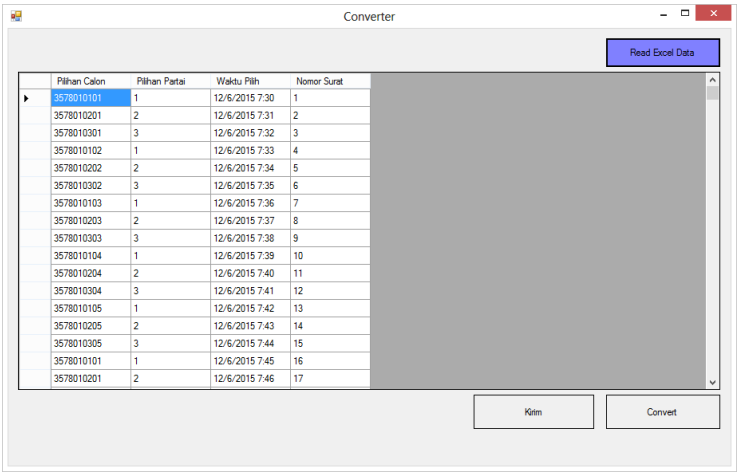
Uji coba ini dilakukan dengan menyiapkan data suara yang akan dikirimkan oleh TPS ke server kelurahan. Data suara tersebut harus mengikuti format basis data yang telah diamankan dengan *hash* dan *signature*. Untuk memenuhi hal tersebut, maka pada tugas akhir ini juga dibuat program untuk melakukan konversi data suara dari yang bersifat *plain* ke bentuk yang terformat. Tampilan program *converter* yang digunakan dapat dilihat pada gambar 5.1.



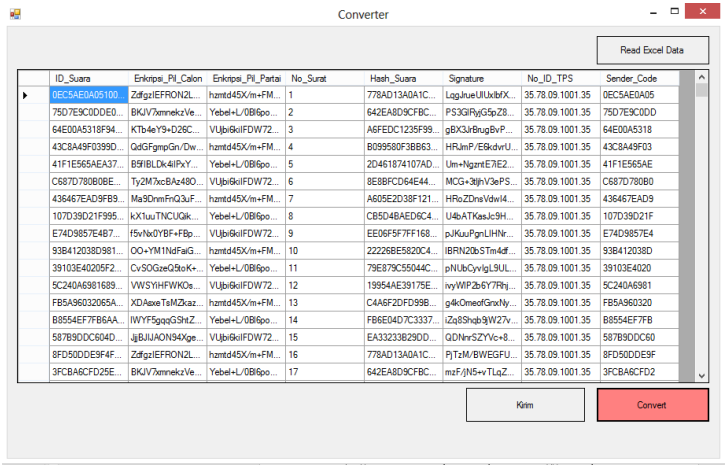
Gambar 0.1. Tampilan program converter untuk data suara.

Untuk pengujian ini, disediakan 250 data suara *plain* yang tersimpan dalam *file plainData.xls.* Contoh data suara *plain* ini dapat dilihat pada halaman LAMPIRAN tabel A.1. Data tersebut dikonversi dan dikirimkan ke server kelurahan menggunakan

aplikasi *converter* diatas. Data hasil konversi juga dapat dilihat pada halaman LAMPIRAN tabel A.2. Tampilan program konverter ketika melakukan proses konversi dapat dilihat pada gambar 5.2 dan 5.3.

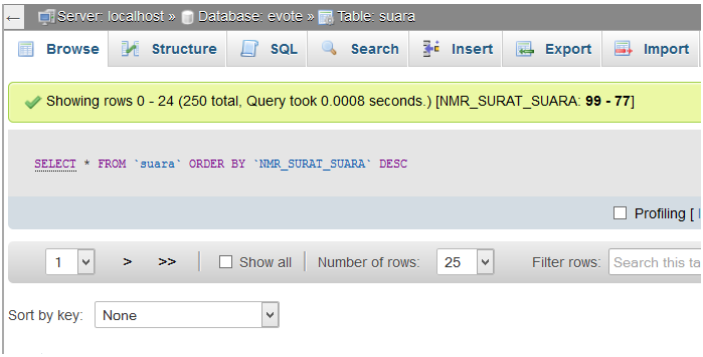


Gambar 0.2. Program *converter* ketika membuka data *plain*.

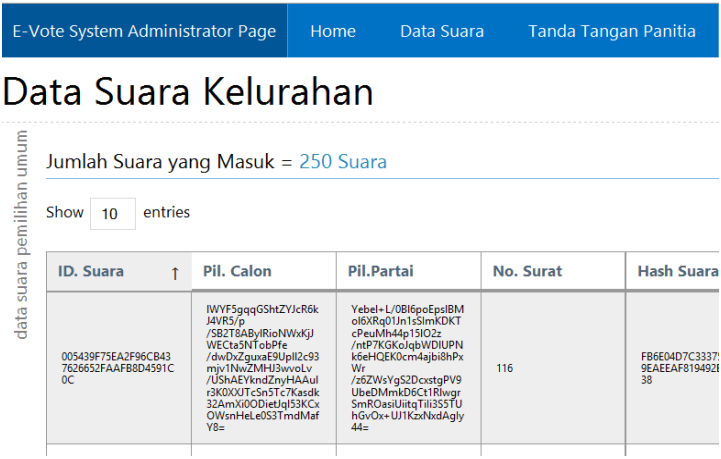


Gambar 0.3. Program *converter* setelah data berhasil dikonversi.

Setelah format data sesuai, data suara tersebut dikirimkan ke server kelurahan menggunakan *web service* pada tugas akhir ini dengan menekan tombol KIRIM pada program *converter*. Jika TPS dan *signature* dari data suara valid maka data tersebut akan masuk ke basis data server kelurahan. Pada gambar 5.4 dan 5.5 dapat dilihat jumlah data yang masuk ke server kelurahan dari proses pengiriman uji coba ini.



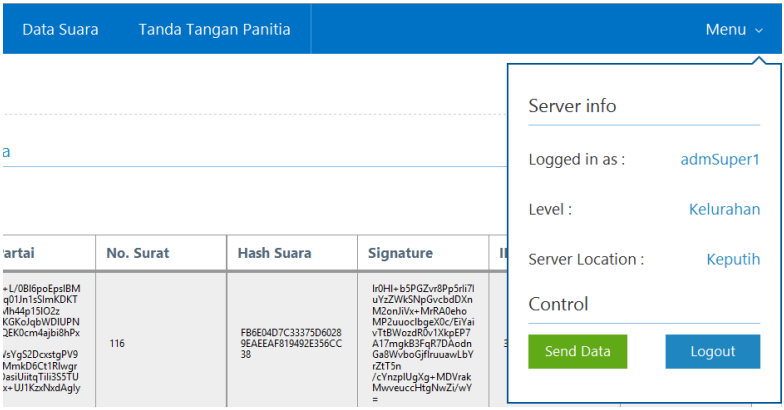
Gambar 0.4. Hasil *query select all* pada basis data server kelurahan



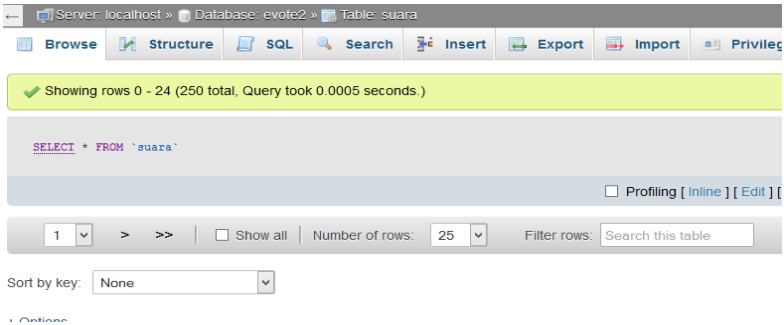
Gambar 0.5. Halaman data suara pada aplikasi web reporting

5.2.2 Skenario Uji Fungsionalitas 2

Pada uji coba ini, data yang digunakan sama dengan data suara pada skenario uji fungsionalitas 1. Uji coba ini dilakukan dengan melanjutkan proses pengiriman data dari server kelurahan ke server kecamatan. Proses pengiriman dilakukan dengan menekan tombol *Send Data* pada aplikasi *web reporting*. Tampilan halaman *web reporting* untuk melakukan proses pengiriman dapat dilihat pada gambar 5.6. Kemudian pada gambar 5.7 dapat dilihat hasil pengiriman data suara pada basis data server kecamatan.



Gambar 0.6. Tombol send data pada aplikasi *web reporting*

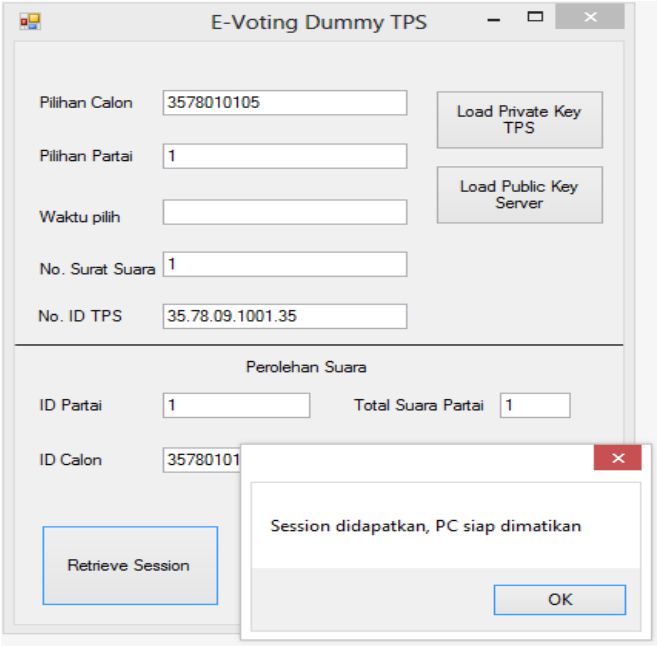


Gambar 0.7. Hasil *query select all* pada basis data server kecamatan

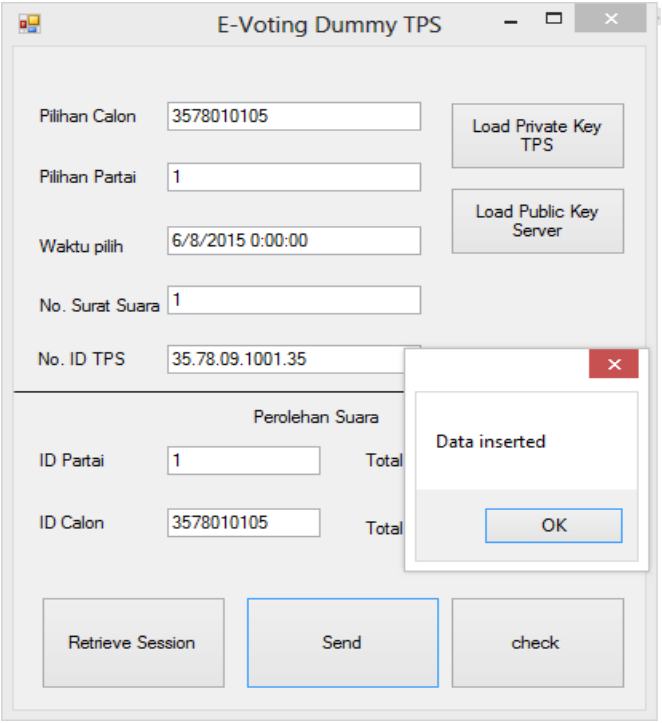
Dari uji coba pada gambar 5.6 dan gambar 5.7, tampak bahwa seluruh data dari server kelurahan berhasil terkirim ke server kecamatan dilihat dari hasil *query* pada server kecamatan.

5.2.3 Skenario Uji Fungsionalitas 3

Uji coba iki dilakukan dengan menjalankan aplikasi *Dummy TPS* yang telah dibuat. Aplikasi dijalankan hingga memperoleh *sessionID* dari server kemudian aplikasi akan dimatikan. Setelah itu, aplikasi akan dijalankan kembali dan digunakan untuk mengirimkan data memanfaatkan *SessionID* yang sebelumnya. Tampilan program ketika memperoleh *SessionID* dapat dilihat pada gambar 5.8 sedangkan tampilan aplikasi ketika mengirimkan data dapat dilihat pada gambar 5.9.



Gambar 0.8. Tampilan aplikasi *dummy tps* ketika memperoleh *session*

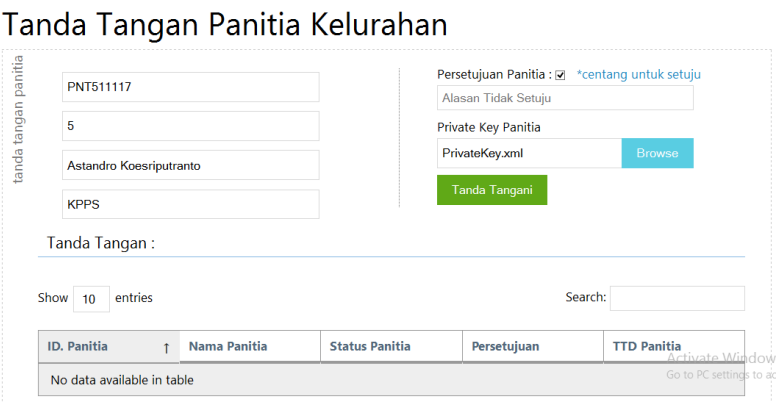


Gambar 0.9. Tampilan aplikasi *dummy tps* dengan pesan dari server

Dari uji coba pada gambar 5.9 dapat dilihat bahwa pengiriman data dengan memanfaatkan *sessionID* yang telah diperoleh pada program yang telah ditutup sebelumnya berhasil dilakukan dan data terkirim ke server.

5.2.4 Skenario Uji Fungsionalitas 4

Uji coba ini dilakukan dengan membuka halaman *Tanda Tangan Panitia* pada aplikasi *web reporting*. Tanda tangan dilakukan dengan mengisikan informasi milik panitia beserta *private key* panitia yang digunakan untuk menandatangani server. Tampilan pengujian tanda tangan panitia ini dapat dilihat pada gambar 5.10 dan gambar 5.11.



Gambar 0.10. Mengisi informasi panitia untuk penandatanganan

Gambar 0.11. Tampilan setelah proses penandatanganan selesai

Dari uji pada gambar 5.10 dan 5.11 dapat dilihat bahwa proses penandatanganan berhasil dilakukan. Tanda tangan yang masuk juga dapat ditampilkan pada halaman tanda tangan panitia.

5.2.5 Skenario Uji Fungsionalitas 5

Uji coba ini dilakukan dengan melakukan proses login pada aplikasi TPS. Proses login dilakukan dengan dua buah variasi yaitu dengan informasi *username* dan *password* yang benar dan salah.

Gambar 0.12. Login menggunakan ID admin TPS yang sesuai

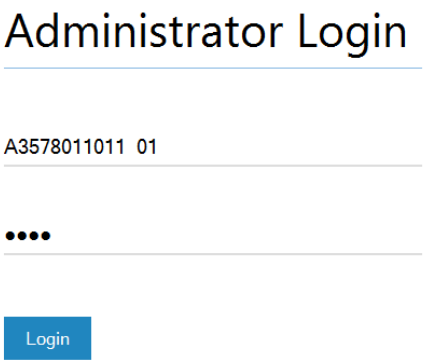
Gambar 0.13. Pesan jika proses login berhasil

Gambar 0.14. Login menggunakan ID admin TPS yang salah

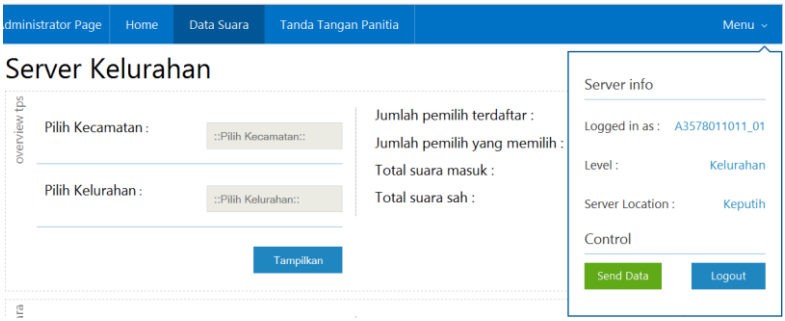
Gambar 0.15. Pesan jika proses login gagal

5.2.6 Skenario Uji Fungsionalitas 6

Uji coba ini dilakukan dengan melakukan proses login pada aplikasi *web reporting*. Proses login dilakukan dengan dua buah variasi yaitu dengan informasi *username* dan *password* yang benar dan salah.



Gambar 0.16. Login sebagai admin web reporting server kelurahan



Gambar 0.17. Tampilan halaman web reporting server kelurahan

Dari pengujian pada gambar 5.16 dengan memasukkan *username* dan *password* untuk proses login berhasil dilakukan dibuktikan dengan berhasil masuk ke halaman utama aplikasi *web reporting* pada gambar 5.17.

5.3 Skenario Uji Keamanan

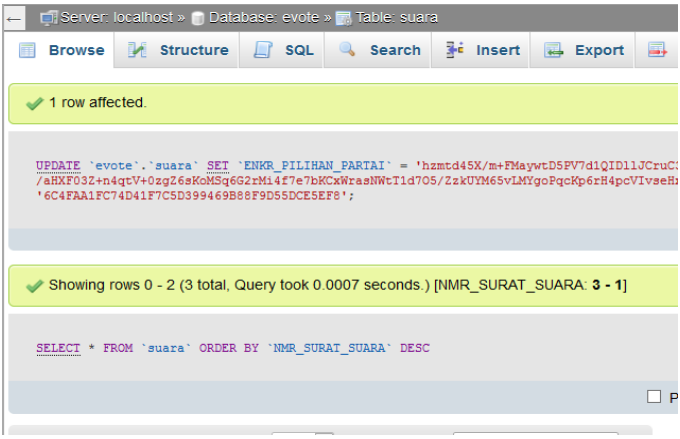
Uji coba ini dilakukan untuk menguji keamanan sistem pada tugas akhir ini dengan menerapkan beberapa metode ancaman

keamanan yang mungkin terjadi dalam studi kasus yang ditentukan. Dalam uji coba ini terdapat 5 macam skenario pengujian keamanan yaitu :

1. Pemalsuan data suara hasil pemilihan yang dikirim.
2. *Packet sniffing* terhadap informasi yang dikirimkan dari TPS ke server.
3. Melakukan *replay attack* menggunakan data suara hasil pemilihan.
4. Melakukan pengujian menggunakan *tools accunetix*, *sqlmap*, dan *nessus*.

5.3.1 Skenario Uji Keamanan 1

Uji coba ini dilakukan dengan melakukan pemalsuan data suara yang akan dikirimkan oleh server kelurahan ke server kecamatan yaitu pada bagian partai yang dipilih. Dengan data tersebut, akan diuji bagaimana reaksi dari server kecamatan ketika menerima data palsu tersebut. Uji coba pemalsuan data yang dikirim ini dapat dilihat pada gambar 5.18 dan 5.19.



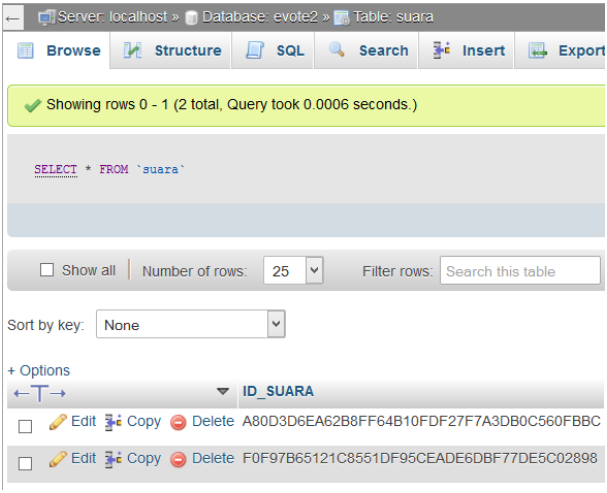
Gambar 0.18. Pemalsuan data suara pada basis data server kelurahan

Dari pengujian pada gambar 5.18 dapat dilihat bahwa terdapat 3 buah data suara yang salah satu data suaranya dilakukan

pemalsuan. Pada gambar 5.19 terbukti bahwa data yang masuk ke server kelurahan hanya ada 2 buah. Hal ini dikarenakan data yang dipalsukan ditolak oleh server dan tidak masuk ke basis data.

5.3.2 Skenario Uji Keamanan 2

Uji coba ini dilakukan dengan melakukan prose *packet sniffing* menggunakan *tools Wireshark* yang dipasang dan dijalankan menggunakan PC di luar sistem namun berada pada jaringan yang sama. Proses ini dilakukan untuk mencoba memperoleh informasi data yang dikirimkan dari TPS ke server kelurahan. Informasi yang diperoleh menggunakan wireshark dari TPS pengirim dengan alamat IP 10.151.63.18 ke server kelurahan dengan alamat IP 10.151.32.55 dapat dilihat pada halaman LAMPIRAN gambar A.3.

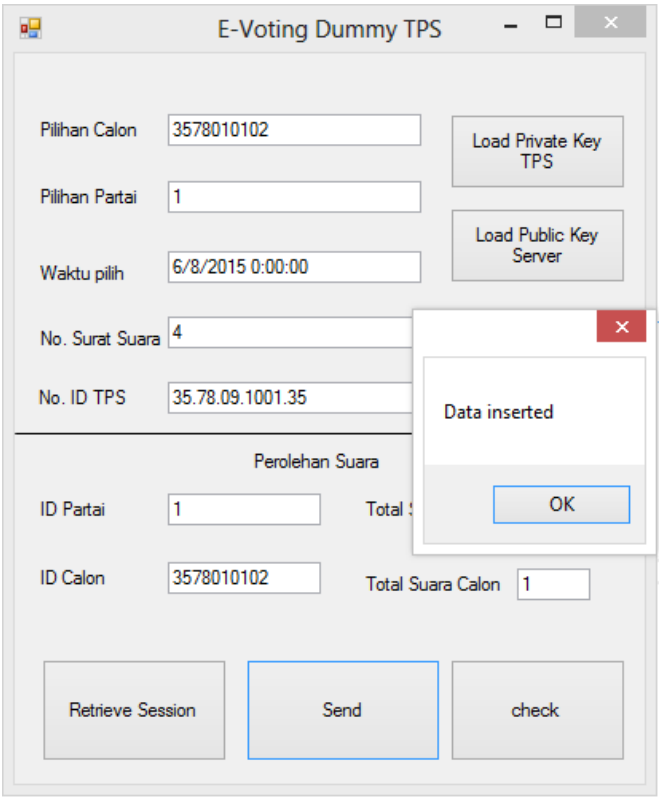


Gambar 0.19. Data yang diterima di basis data server kecamatan

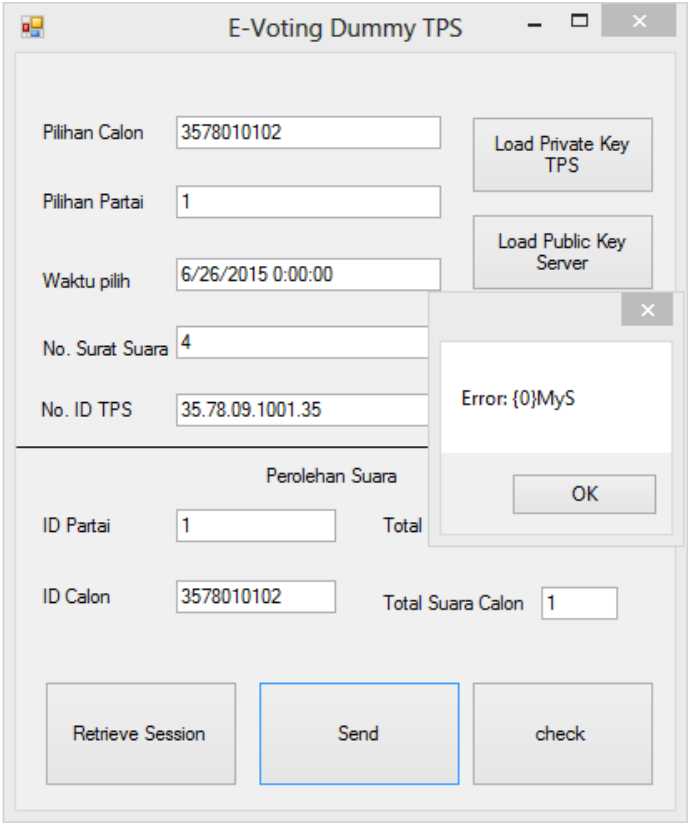
5.3.3 Skenario Uji Keamanan 3

Uji coba ini dilakukan dengan mnengirimkan data yang sama persis melalui TPS ke server kelurahan. Uji coba ini dilakukan dengan menggunakan aplikasi *dummy tps* untuk mengirimkan data

suara yang sama persis. Uji coba ini dilakukan untuk mengetahui bagaimana reaksi server terhadap adanya data suara yang sama yang dikirimkan oleh TPS.



Gambar 0.20 Pengiriman data pertama dari TPS



Gambar 0.21. Pengiriman data yang sama persis dari TPS

5.3.4 Skenario Uji Keamanan 4

Pada uji coba ini dilakukan pengecekan celah keamanan pada plikasi *web service* dan server. Dalam uji coba ini, digunakan *tools* untuk melakukan *vulnerability testing* diantaranya menggunakan Accunetix untuk melakukan *Web Vulnerability Scanning*, SQLMap untuk melakukan *SQL Inject* pada aplikasi *web service*, dan Nessus untuk melakukan pengecekan keamanan pada pc server. Dalam uji coba ini didapatkan hasil sebagai berikut :

1. *Web Vulnerability Scanning* dengan Accunetix.
   Dengan menggunakan tools ini, dilakukan *default scan* terhadap *web service* target yaitu https://10.151.32.55/evoteservice/Service1.asmx. Dari hasil *scanning* deperoleh informasi peringatan keamanan tingkat rendah (*low)* sebanyak 4 buah yaitu kemungkinan ancaman *clickjacking*, di-*enable* nya *OPTIONS Method*, dan *possible sensitive directory* sebanyak 2 buah. Sementara itu, untuk proses *scanning* khusus web service, tidak ditemukan adanya ancaman terhadap *web service* pada tugas akhir ini. Gambar detail mengenai hasil dari proses *scanning* ini dapat dilihat pada halaman LAMPIRAN gambar A.4 – gambar A.7.
2. *SQL Inject* dengan SQLMap.
   Dalam pengujian menggunakan *tools* ini, digunakan data data XML dari *request* yang dilakukan aplikasi *dummy TPS* yang direkam menggunakan aplikasi proxy FIDDLER.. Uji coba *sql inject* ini diterapkan pada seluruh *web method* yang terdapat pada *web service* tugas akhir ini dengan level 4 untuk parameter pada SQLMap. Hasil dari uji coba ini dan detail file SOAP *request* yang digunakan dapat dilihat pada halaman LAMPIRAN gambar A.10 – gambar A.15 dan kode sumber A.1 – kode sumber A.6.
3. Server *Vulnerability Scanning* dengan Nessus.
   Dalam uji coba menggunakan *tools* ini, didapatkan hasil celah keamanan *critical* sebanyak 3 buah,serta medium 12 buah. Detail hasil *scanning* dapat dilihat pada halaman LAMPIRAN gambar A.8 – gambar A.9.

5.4 Analisis Hasil Uji Coba

Dari hasil uji fungsionalitas pada bab sebelumnya, dapat dilihat bawha aplikasi-aplikasi yang dibuat mampu memenuhi fungsionalitas sistem e-voting sesuai pada studi kasus yang

digunakan. Fitur-fitur yang disediakan berjalan dengan baik mulai dari aplikasi *web service, web reporting,* dan *dummy tps*.

Dalam uji kemanan pada bab sebelumnya, pada skenario 1 uji keamanan dengan pemalsuan data yang dikirim, dilakukan pengubahan terhadap pilihan partai pada salah satu data suara. Dalam uji coba tersebut dikirimkan 3 buah data suara dengan salah satu data suaranya telah diubah. Reaksi pada server kelurahan adalah menolak data yang telah diubah tersebut sehingga hanya ada 2 buah data suara yang diterima. Hal ini dikarenakan setiap data suara yang dikirim akan dicek *signature*-nya sebelum dimasukkan ke basis data. Dengan demikian, dapat dipastikan data suara yang masuk adalah data yang tidak mengalami perubahan. Pada uji keamanan yang ketiga yaitu uji *replay attack*, data yang sama yang dikirimkan juga ditolak oleh server. Hal ini dikarenakan adanya duplikasi *Primary key* pada data suara yang dikirim sehingga menyebabkan basis data pada server mengirimkan pesan *error* dan menolak proses pemasukan data tersebut ke basis data.

Dalam uji coba menggunakan *tools*, pada *vulnerability* scanning tidak ditemukan adanya ancaman keamanan yang serius pada aplikasi *web service* maupun server.

BAB VI
KESIMPULAN DAN SARAN

Bab kesimpulan dan saran berisi mengenai simpulan-simpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang telah dikemukakan. Selan itu, pada bab ini terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan antara lain:

1. Dari hasil uji fungsionalitas yang telah dilakukan, terbukti bahwa sistem yang dibuat mampu menangani proses utama dalam sistem pemilihan umum elektronik (e-voting) dengan baik. Hal ini mencakup proses penyimpanan data suara serta proses pengiriman data suara hasil pemilihan.
2. Dari hasil uji pemalsuan data terbukti bahwa sistem mampu mencegah masuknya data palsu ke server dan menjaga integritas data yang dikirim. Dengan adanya *signature* pada setiap data suara, maka dapat dipastikan bahwa data yang berhasil masuk ke server merupakan data yang asli dan tidak mengalami perubahan sebelum atau selama proses pengiriman.
3. Dari hasil uji *packet sniffing* atau penyadapan data yang dikirimkan terbukti bahwa kerahasiaan data yang dikirim terjaga dengan baik karena data yang disadap tidak dapat terbaca oleh penyadap. Hal ini juga sekaligus membuktikan bahwa privasi pemilih terjaga dengan baik.
4. Uji *replay attack* yang telah dilakukan membuktikan bahwa sistem mampu menangani ancaman serangan tersebut dengan baik. Dengan demikian, dapat dipastikan tidak ada data duplikat yang diterima oleh server yang dapat mempengaruhi hasil perolehan suara dalam sistem e-voting.

5. Berdasarkan hasil *scan* sistem (server dan aplikasi) menggunakan *tools* Accunetix, SQLMap, dan Nessus, terbukti bahwa celah keamanan pada sistem sangat rendah. Hasil *scan* menunjukkan bahwa sistem memiliki beberapa celah keamanan dengan level rendah yang tidak terlalu berpengaruh terhadap keamanan sistem. Hasil uji penetrasi juga menunjukkan bahwa sistem (*web serviceI*) aman dari serangan SQL *Injection*.

6.2 Saran

Beberapa saran yang hendak disampaikan terkait dengan pengerjaan tugas akhir ini adalah :

1. Perlu dikembangkan metode manajemen *public* dan *private key* yang digunakan dalam sistem e-voting baik yang dimiliki oleh TPS, server, ataupun panitia/saksi pemilihan umum.
2. Perlu dilakukan pengembangan aplikasi *web reporting* yang lebih detail, dinamis, dan responsif serta menyesuaikan dengan kebutuhan pihak KPU selaku organisasi yang terlibat langsung dengan aplikasi tersebut sekaligus satu-satunya yang berhak menentukan proses bisnis apa saja serta apa saja keamanan yang diperlukan dalam aplikasi tersebut.

BAB VI
KESIMPULAN DAN SARAN

Bab kesimpulan dan saran berisi mengenai simpulan-simpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang telah dikemukakan. Selan itu, pada bab ini terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1 Kesimpulan

Kesimpulan yang diperoleh berdasarkan uji coba dan evaluasi yang telah dilakukan antara lain:

1. Dari hasil uji fungsionalitas yang telah dilakukan, terbukti bahwa sistem yang dibuat mampu menangani proses utama dalam sistem pemilihan umum elektronik (e-voting) dengan baik. Hal ini mencakup proses penyimpanan data suara serta proses pengiriman data suara hasil pemilihan.
2. Dari hasil uji pemalsuan data terbukti bahwa sistem mampu mencegah masuknya data palsu ke server dan menjaga integritas data yang dikirim. Dengan adanya *signature* pada setiap data suara, maka dapat dipastikan bahwa data yang berhasil masuk ke server merupakan data yang asli dan tidak mengalami perubahan sebelum atau selama proses pengiriman.
3. Dari hasil uji *packet sniffing* atau penyadapan data yang dikirimkan terbukti bahwa kerahasiaan data yang dikirim terjaga dengan baik karena data yang disadap tidak dapat terbaca oleh penyadap. Hal ini juga sekaligus membuktikan bahwa privasi pemilih terjaga dengan baik.
4. Uji *replay attack* yang telah dilakukan membuktikan bahwa sistem mampu menangani ancaman serangan tersebut dengan baik. Dengan demikian, dapat dipastikan tidak ada data duplikat yang diterima oleh server yang dapat mempengaruhi hasil perolehan suara dalam sistem e-voting.

5. Berdasarkan hasil *scan* sistem (server dan aplikasi) menggunakan *tools* Accunetix, SQLMap, dan Nessus, terbukti bahwa celah keamanan pada sistem sangat rendah. Hasil *scan* menunjukkan bahwa sistem memiliki beberapa celah keamanan dengan level rendah yang tidak terlalu berpengaruh terhadap keamanan sistem. Hasil uji penetrasi juga menunjukkan bahwa sistem (*web serviceI*) aman dari serangan SQL *Injection*.

6.2 Saran

Beberapa saran yang hendak disampaikan terkait dengan pengerjaan tugas akhir ini adalah :

1. Perlu dikembangkan metode manajemen *public* dan *private key* yang digunakan dalam sistem e-voting baik yang dimiliki oleh TPS, server, ataupun panitia/saksi pemilihan umum.
2. Perlu dilakukan pengembangan aplikasi *web reporting* yang lebih detail, dinamis, dan responsif serta menyesuaikan dengan kebutuhan pihak KPU selaku organisasi yang terlibat langsung dengan aplikasi tersebut sekaligus satu-satunya yang berhak menentukan proses bisnis apa saja serta apa saja keamanan yang diperlukan dalam aplikasi tersebut.

DAFTAR PUSTAKA

[1] A. Shamir, D. A. Wagner dan R. L. Rivest, "Time-lock puzzles and timed-release Crypto," 1996.

[2] M. Rouse, "What is RSA Algorithm," [Online]. Available: http://searchsecurity.techtarget.com/definition/RSA. [Diakses Januari 2015].

[3] M. Rouse, "What is digital signature?," [Online]. Available: http://searchsecurity.techtarget.com/definition/digital-signature. [Diakses Januari 2015].

[4] A. Ridwan, "Pengertian HTTP, HTTPS, URL, FTP, DOMAIN | Definisi HTTP," 3 Juni 2013. [Online]. Available: http://www.impoint.info/2013/06/pengertian-http-https-url-ftp-domain.html#axzz3bp8QzDI8. [Diakses 1 Juni 2015].

[5] J. Kyrnin, "Signed vs. Self-signed Certificates," [Online]. Available: http://webdesign.about.com/od/ssl/a/signed_v_selfsi.htm. [Diakses 1 Juni 2015].

[6] A. Kak, "Engineering.purdue.edu," 22 April 2015. [Online]. Available: https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture12.pdf. [Diakses 1 Juni 2015].

[7] R. Deviani dan H. C. Chen, "A Secure E-Voting System Based on RSA Time-Lock Puzzle Mechanism," 2012.

[8] W. Clarkson, "Time-Lock Cryptography Sending Message to the Future," 2013.

[9] "What is E-Voting," [Online]. Available: http://whatis.techtarget.com/definition/e-voting-electronic-voting. [Diakses Desember 2014].

[10] "What is a hash function," [Online]. Available: http://x5.net/faqs/crypto/q94.html. [Diakses Januari 2015].

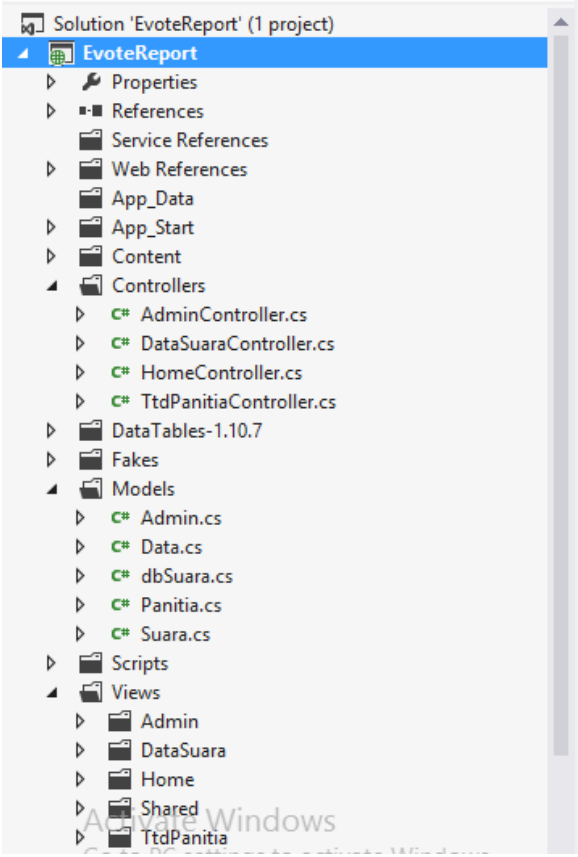
[11] "replay-attack," [Online]. Available: http://www.yourdictionary.com/replay-attack. [Diakses 1 Juni 2015].

[12] "Public Key: RSA," Oktober 2012. [Online]. Available: http://ilmukriptografi.wordpress.com/2012/10/24/public-key-rsa/. [Diakses Desember 2014].

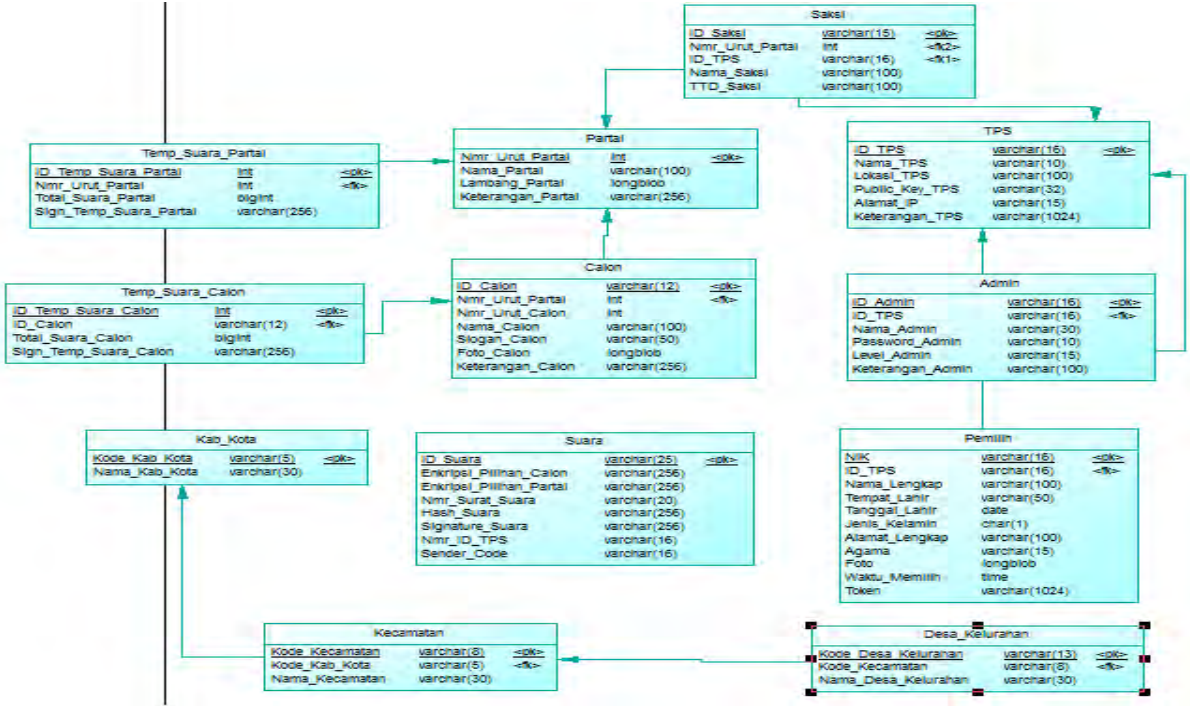
[13] "http://www.ma2.upc.edu/," [Online]. Available: http://www-ma2.upc.es/~cripto/Q2-06-07/SHA256english.pdf. [Diakses 15 6 2015].

[14] "elearning.amikom.ac.id," [Online]. Available: http://elearning.amikom.ac.id/index.php/download/materi/190000005-ST047-17/2011/06/20110604_Tutorial. [Diakses 15 6 2015].

LAMPIRAN



Gambar A.1. Hirarki aplikasi *web reporting*



GambarA.2. PDM basis data sistem e-voting

Tabel A.1. Tabel Contoh Data Suara *Plain*

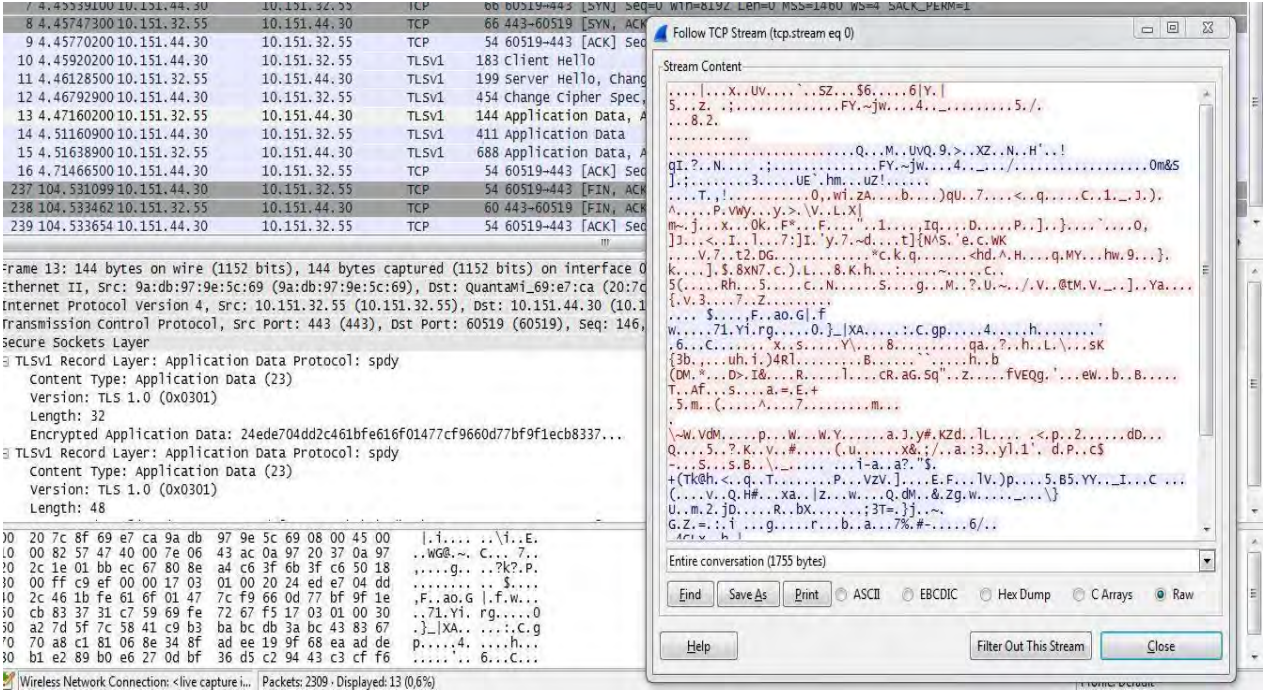
Pilihan Calon	Pilihan Partai	Waktu Pilih	Nomor Surat
3578010101	1	12/6/2015 7:30	1
3578010201	2	12/6/2015 7:31	2
3578010301	3	12/6/2015 7:32	3
3578010102	1	12/6/2015 7:33	4
3578010202	2	12/6/2015 7:34	5
3578010302	3	12/6/2015 7:35	6
3578010103	1	12/6/2015 7:36	7
3578010203	2	12/6/2015 7:37	8
3578010303	3	12/6/2015 7:38	9
3578010104	1	12/6/2015 7:39	10
3578010204	2	12/6/2015 7:40	11
3578010304	3	12/6/2015 7:41	12
3578010105	1	12/6/2015 7:42	13
3578010205	2	12/6/2015 7:43	14
3578010305	3	12/6/2015 7:44	15
3578010101	1	12/6/2015 7:45	16
3578010201	2	12/6/2015 7:46	17
3578010301	3	12/6/2015 7:47	18
3578010102	1	12/6/2015 7:48	19

Tabel A.2. Tabel Contoh Data Suara *Converted*

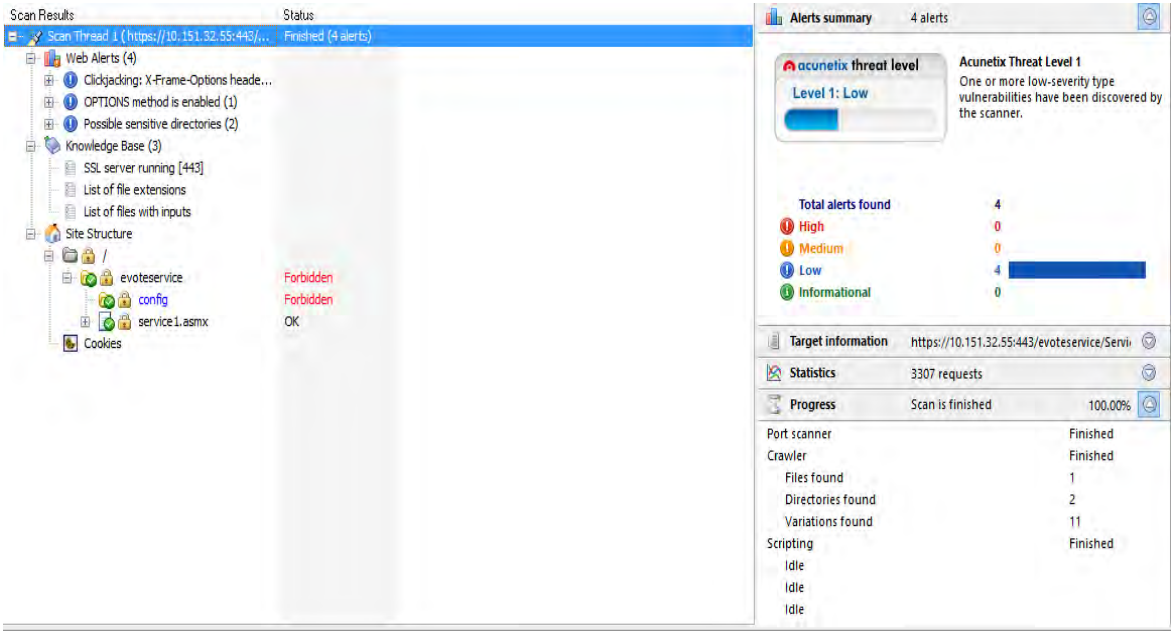
ID_Suara	Enkripsi_Pil_Calon	Enkripsi_Pil_Partai	No_Surat	Hash_Suara	Signature	No_ID_TPS	Sender_Code
0EC5AE0A0510075A	ZdfgzIEFRON2LWQnC	hzmtd45X/m+FMaywt	1	778AD13A0A1CA2E3525ED0C983314635F22C9180	LqgJrueUlUxIbfX2IdV	35.78.09.1001.35	0EC5AE0A05
75D7E9C0DDE011F3	BKJV7xmnekzVeZ3W/	Yebel+L/0Bl6poEpslBM	2	642EA8D9CFBC035208BF00F69DF84B9C0C410B2B	PS3GlRyjG5pZ8AhSxv	35.78.09.1001.35	75D7E9C0DD
64E00A5318F945AC	KTb4eY9+D26CAIPJhR	VUjbi6kiIFDW72/bjijjy	3	A6FEDC1235F99DE79D6187A2427897113 41B1E06	gBX3JrBrugBvPG7GrY	35.78.09.1001.35	64E00A5318
43C8A49F0399DF49	QdGFgmpGn/DwwyM	hzmtd45X/m+FMaywt	4	B099580F3BB634D59FE245C3286E4AB243C4279B	HRJmP/E6kdvrUCYi8E	35.78.09.1001.35	43C8A49F03
41F1E565AEA371E7	B5fIBLDk4iIPxYHqBW/	Yebel+L/0Bl6poEpslBM	5	2D461874107AD7A79CA15F773B7506A2B2501244	Um+NgzntE7iE2hiKM	35.78.09.1001.35	41F1E565AE
C687D780B0BE7C22	Ty2M7xcBAz48O86uY	VUjbi6kiIFDW72/bjijjy	6	8E8BFCD64E44D9B0D3663884415FD3A541E01A18	MCG+3tljhV3ePSGSY	35.78.09.1001.35	C687D780B0
436467EAD9FB98BE	Ma9DnmFnQ3uFv50a	hzmtd45X/m+FMaywt	7	A605E2D38F1213977CAAB31C118A28A3FDC36862	HRoZDnsVdwI4FuvZi	35.78.09.1001.35	436467EAD9
107D39D21F99536C	kX1uuTNCUQikWBZtJ	Yebel+L/0Bl6poEpslBM	8	CB5D4BAED6C48EE88527C8980D33040E375FA0F5	U4bATKasJc9HXPp6t\	35.78.09.1001.35	107D39D21F
E74D9857E4B7B296	f5vNx0YBF+FBpdGEE(	VUjbi6kiIFDW72/bjijjy	9	EE06F5F7FF168189A3BC4C8F14B3CBFFE9305DA8	pJKuuPgnLIHNrBrsfD	35.78.09.1001.35	E74D9857E4
93B412038D98121D	OO+YM1NdFaiGKGF9	hzmtd45X/m+FMaywt	10	22226BE5820C46FB0288730A15F2017D68CD57BA	lBRN20bSTm4dfq0qr	35.78.09.1001.35	93B412038D
39103E40205F21AE	CvSOGzeQ5toK+xyIM	Yebel+L/0Bl6poEpslBM	11	79E879C55044CB9FB76758ED8E14CDA054161CDE	pNUbCyvIgL9ULNmP	35.78.09.1001.35	39103E4020
5C240A698168968F	VWSYiHFWKOs3OK8/	VUjbi6kiIFDW72/bjijjy	12	19954AE39175E6C4F6AF6A7A6B928AB730B64DC5	ivyWlP2b6Y7Rhj4PZs	35.78.09.1001.35	5C240A6981
FB5A96032065A4C8	XDAsxeTsMZkazfjVtZ	hzmtd45X/m+FMaywt	13	C4A6F2DFD99B3BBC672153AFFDD560014440694F	g4kOmeofGnxNyOUf	35.78.09.1001.35	FB5A960320
B8554EF7FB6AA54E	lWYF5gqqGShtZYJcR6	Yebel+L/0Bl6poEpslBM	14	FB6E04D7C33375D60289EAEEAF819492E356CC38	iZq8Shqb9jW27vPdJ7	35.78.09.1001.35	B8554EF7FB
587B9DDC604D8147	JjjBJIJAON94Xge0NYg	VUjbi6kiIFDW72/bjijjy	15	EA33233B29DD431859F41E32912C556903DE8611	QDNnrSZYVc+8AKfVF	35.78.09.1001.35	587B9DDC60
8FD50DDE9F4F0BD/	ZdfgzIEFRON2LWQnC	hzmtd45X/m+FMaywt	16	778AD13A0A1CA2E3525ED0C983314635F22C9180	PjTzM/BWEGFUTWKc	35.78.09.1001.35	8FD50DDE9F
3FCBA6CFD25EB395	BKJV7xmnekzVeZ3W/	Yebel+L/0Bl6poEpslBM	17	642EA8D9CFBC035208BF00F69DF84B9C0C410B2B	mzF/jN5+vTLqZqSYSk	35.78.09.1001.35	3FCBA6CFD2
78F711DA795A4183	KTb4eY9+D26CAIPJhR	VUjbi6kiIFDW72/bjijjy	18	A6FEDC1235F99DE79D6187A2427897113 41B1E06	cSiDTf8TVxufAla9v0F	35.78.09.1001.35	78F711DA79
30F43C94E4D1C6FE	QdGFgmpGn/DwwyM	hzmtd45X/m+FMaywt	19	B099580F3BB634D59FE245C3286E4AB243C4279B	Qqi490KyX45XXoGfh	35.78.09.1001.35	30F43C94E4

Tabel A.3. Tabel Hasil Uji Fungsionalitas Sistem E-voting

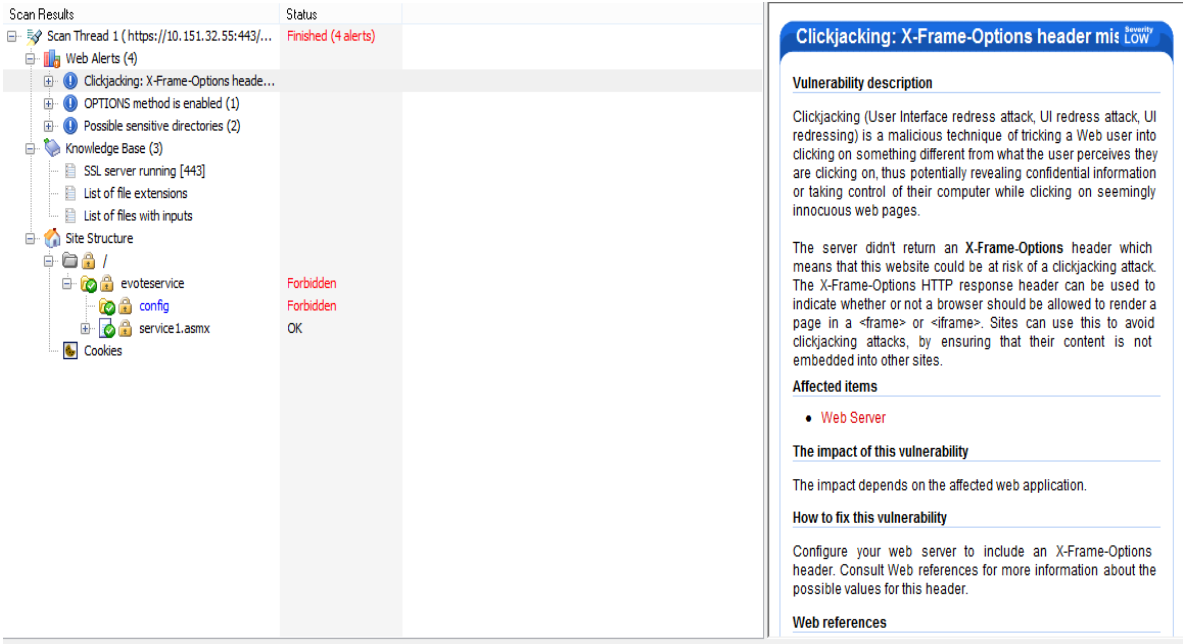
No.	Nama Uji	Aplikasi	Alamat IP	IP Server	Jumlah data	Status
1	Kirim Data (TPS)	Dummy TPS	10.151.43.113	10.151.32.55	250	Sukses
2	Kirim Data (Web Reporting)	Web Reporting	10.151.32.55	10.151.32.55	250	Sukses
3	Connection Lost	Dummy TPS	10.151.43.113	10.151.32.55	1	Sukses
4	Tanda Tangan Panitia	Web Reporting	10.151.32.55	10.151.32.55	3	Sukses
5	Login TPS	Aplikasi TPS	10.151.43.173	10.151.32.55	1	Sukses
6	Login Web Reporting	Web Reporting	10.151.32.55	10.151.32.55	1	Sukses



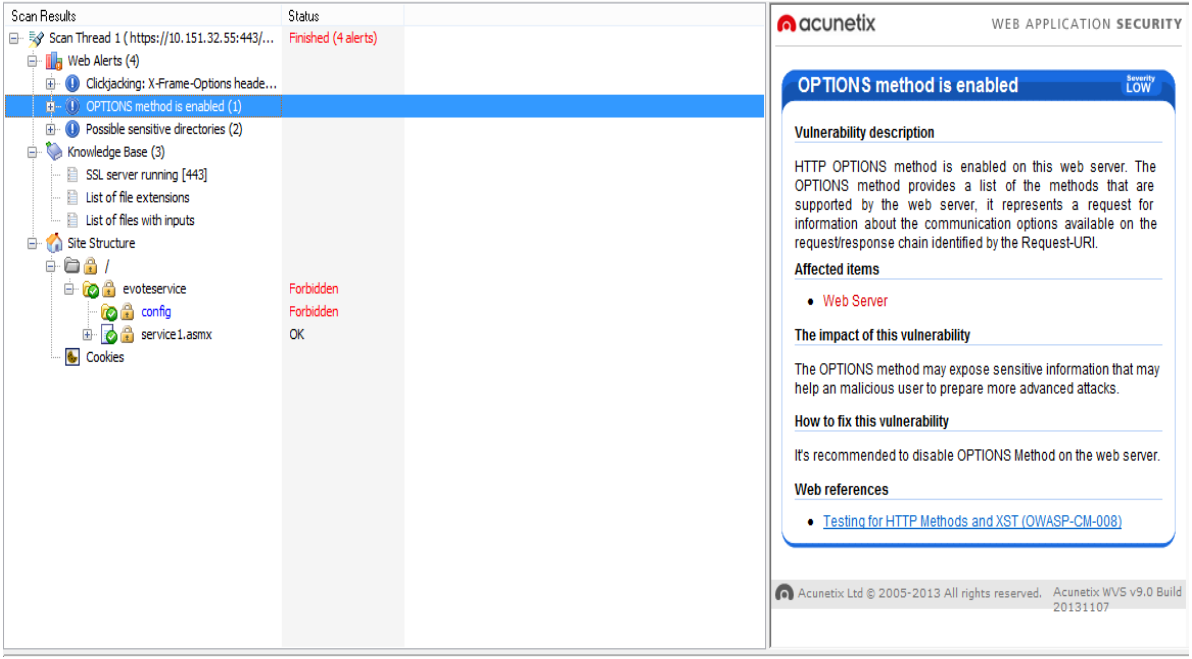
Gambar A.3. Hasil packet sniffing dengan wireshark



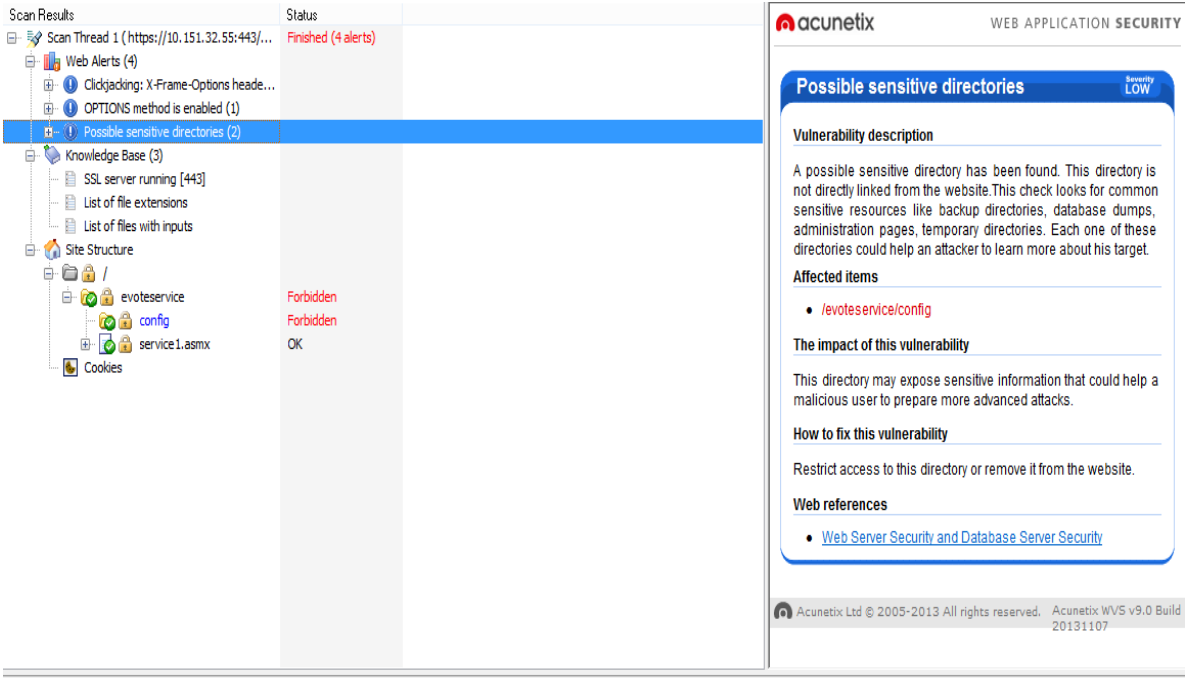
Gambar A.4. Hasil *vulnerability scan web service* dengan accunetix(1)



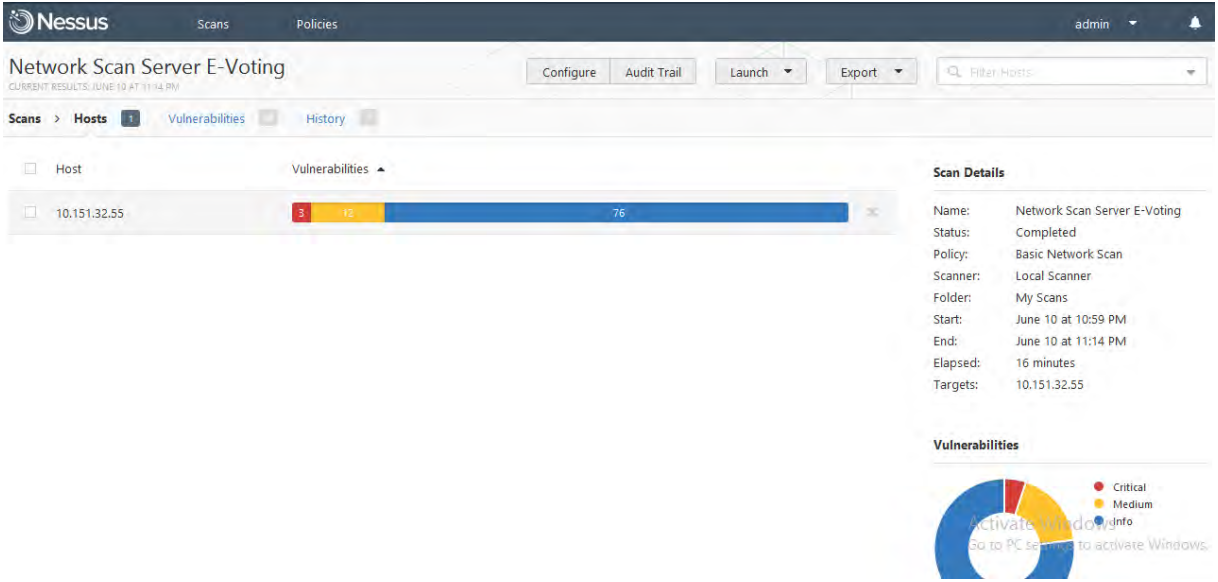
Gambar A.5. Hasil *vulnerability scan web service* dengan accunetix(2)



Gambar A.6. Hasil *vulnerability scan web service* dengan accunetix(3)



Gambar A.7. Hasil *vulnerability scan web service* dengan accunetix(4)



Gambar A.8. Hasil *network scan* server kelurahan dengan nessus

Severity ▲	Plugin Name	Plugin Family	Count
CRITICAL	MS15-034: Vulnerability in HTTP.sys Could Allow Remote Code Execution (30...	Windows	2
CRITICAL	MS14-066: Vulnerability in Schannel Could Allow Remote Code Execution (29...	Windows	1
MEDIUM	SSL Certificate Cannot Be Trusted	General	3
MEDIUM	Nonexistent Page (404) Physical Path Disclosure	Web Servers	2
MEDIUM	SSL RC4 Cipher Suites Supported	General	2
MEDIUM	SSL Self-Signed Certificate	General	2
MEDIUM	SMB Signing Required	Misc.	1
MEDIUM	SSL Version 2 and 3 Protocol Detection	Service detection	1
MEDIUM	SSLv3 Padding Oracle On Downgraded Legacy Encryption Vulnerability (POO...	General	1
INFO	netstat portscanner (SSH)	Port scanners	15
INFO	DCE Services Enumeration	Windows	8
INFO	Service Detection	Service detection	7

Gambar A.9. Detail celah keamanan pada server dari hasil *scan* nessus

```
POST
https://10.151.32.55/evoteservice/Service1.a
smx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE
6.0; MS Web Services Client Protocol
4.0.30319.17929)
VsDebuggerCausalityData:
uIDPo5sYD1/v77NEltoX9xrx/9IAAAAAU9e+mkEqYUOO
H0qdRX9FGR4hAFR5sUVDi+pofF8HFDgACQAA
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri.org/Auth"
Host: 10.151.32.55
Content-Length: 579
Expect: 100-continue

<?xml version="1.0" encoding="utf-
8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
><soap:Body><Auth
xmlns="http://tempuri.org/"><HashedID_TPS>02
C2540C3FB89DE2C0D9943CDD37E6952DECF342FA1AAA
BEB5101CB9C3F5621B</HashedID_TPS><EncryptedI
D>bFX8Plsug33or+zWiq45vB9E3dGAOU6llR3xI7qmBZ
SpGPHspKl1nVNMUqgY1HEnGKrynztvA+1zipVDowWt43
+JpfpxBKwE0qPcEjRTjkzHKhY5jdfo5nKzUCkisDZjLS
kQwXhdagyql5iK2WHyY9uvXZdGoev4Slg7H9Kyqqk=</
EncryptedID></Auth></soap:Body></soap:Envelo
pe>
```

Kode Sumber A.1. SOAP *request web method* auth

```
POST
https://10.151.32.55/evoteservice/Service1.a
smx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE
6.0; MS Web Services Client Protocol
4.0.30319.17929)
VsDebuggerCausalityData:
uIDPo50YD1/v77NEltoX9xrx/9IAAAAAU9e+mkEqYUOO
H0qdRX9FGR4hAFR5sUVDi+pofF8HFDgACQAA
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri.org/Close"
Host: 10.151.32.55
Content-Length: 469
Expect: 100-continue

<?xml version="1.0" encoding="utf-
8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
><soap:Body><Close
xmlns="http://tempuri.org/"><HashedID_TPS>02
C2540C3FB89DE2C0D9943CDD37E6952DECF342FA1AAA
BEB5101CB9C3F5621B</HashedID_TPS><SessionID>
EEAD13D999A931BE4C0CB4FE183AE2FF8B1D00CC67D9
9E5C92489004F08ADFDC</SessionID></Close></so
ap:Body></soap:Envelope>
```

Kode Sumber A.2. SOAP *request web method* close

```
POST
https://10.151.32.55/evoteservice/Service1.a
smx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE
6.0; MS Web Services Client Protocol
4.0.30319.17929)
VsDebuggerCausalityData:
uIDPo5kYD1/v77NEltoX9xrx/9IAAAAAU9e+mkEqYUOO
H0qdRX9FGR4hAFR5sUVDi+pofF8HFDgACQAA
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri.org/GetSession"
Host: 10.151.32.55
Content-Length: 392
Expect: 100-continue
Connection: Keep-Alive

<?xml version="1.0" encoding="utf-
8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
><soap:Body><GetSession
xmlns="http://tempuri.org/"><HashedID_TPS>02
C2540C3FB89DE2C0D9943CDD37E6952DECF342FA1AAA
BEB5101CB9C3F5621B</HashedID_TPS></GetSessio
n></soap:Body></soap:Envelope>
```

Kode Sumber A.3. SOAP *request web method* getSession

```
POST
https://10.151.32.55/evoteservice/Service1.a
smx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE
6.0; MS Web Services Client Protocol
4.0.30319.17929)
VsDebuggerCausalityData:
uIDPo5oYD1/v77NEltoX9xrx/9IAAAAAU9e+mkEqYUOO
H0qdRX9FGR4hAFR5sUVDi+pofF8HFDgACQAA
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri.org/Hello"
Host: 10.151.32.55
Content-Length: 382
Expect: 100-continue

<?xml version="1.0" encoding="utf-
8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
><soap:Body><Hello
xmlns="http://tempuri.org/"><HashedID_TPS>02
C2540C3FB89DE2C0D9943CDD37E6952DECF342FA1AAA
BEB5101CB9C3F5621B</HashedID_TPS></Hello></s
oap:Body></soap:Envelope>
```

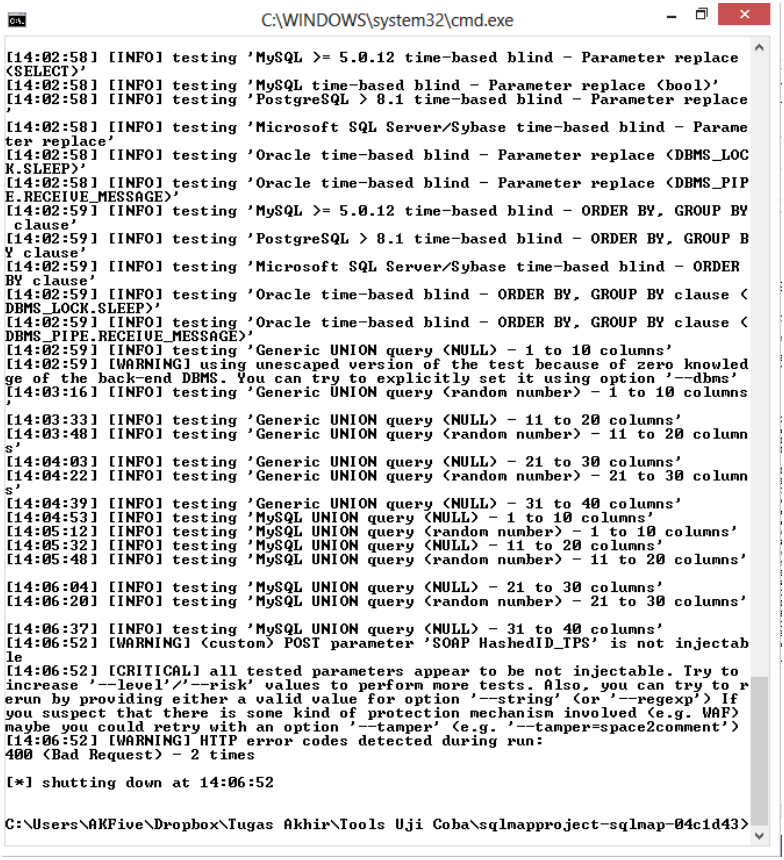
Kode Sumber A.4. SOAP *request web method* hello

```
POST
https://10.151.32.55:443/evoteservice/Servic
e1.asmx HTTP/1.1
Accept-Encoding: identity
Content-Length: 327
Host: 10.151.32.55:443
Content-Type: text/xml; charset=utf-8
Connection: close
User-Agent: 3936

<?xml version="1.0" encoding="utf-
8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
><soap:Body><Login
xmlns="http://tempuri.org/"><user>string</us
er><pass>string</pass></Login></soap:Body></
soap:Envelope>
```

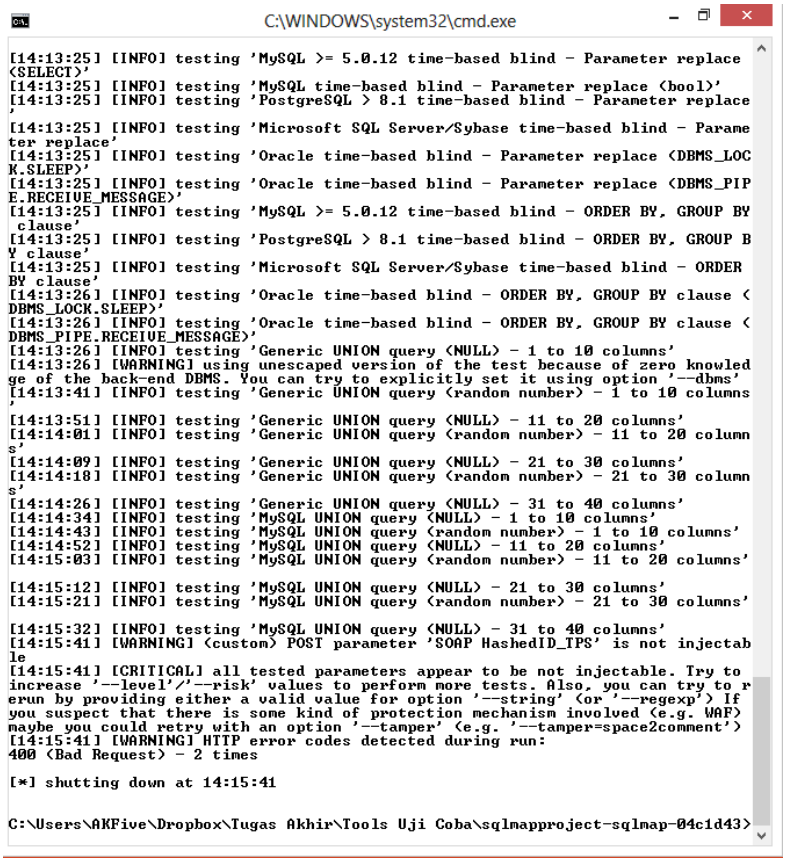
Kode Sumber A.5. SOAP *request web method* login

```
POST https://10.151.32.55/evoteservice/Service1.asmx
HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; MS Web
Services Client Protocol 4.0.30319.17929)
VsDebuggerCausalityData:
uIDPo5wYD1/v77NEltoX9xrx/9IAAAAAU9e+mkEqYUOOH0qdRX9FGR
4hAFR5sUVDi+pofF8HFDgACQAA
Content-Type: text/xml; charset=utf-8
SOAPAction: "http://tempuri.org/Send"
Host: 10.151.32.55
Content-Length: 1215
Expect: 100-continue

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"><soap:Bod
y><Send
xmlns="http://tempuri.org/"><SessionID>EEAD13D999A931B
E4C0CB4FE183AE2FF8B1D00CC67D99E5C92489004F08ADFDC</Ses
sionID><IDSuara>45D56DA179626E9A2F057BA68BB366CD02149C
7C7454461E2A5650010E3FEE94</IDSuara><EnkripsiCalon>k7v
12fOBtfXYHjrvnPXix2bNWU0DD22aruCVHPErc1kbHx7KLRx28ulut
6tFCfX6n4RXonYeMfJL6RiMZnfzuUhRhedCpzruufo7cBDS39ix5gN
agQ3jda+X/zwakrdrKVW/ae4Vbf0gm4XHsPuMbXBKV07EtlL4/3rUL
7KBmiE=</EnkripsiCalon><EnkripsiPartai>hzmtd45X/m+FMay
wtD5PV7d1QIDllJCruC3ONFOcCPX4+MD+Fj/wUmTCMDYF/aHXF03Z+
n4qtV+0zgZ6sKoMSq6G2rMi4f7e7bKCxWrasNWtT1d7O5/ZzkUYM65
vLMYgoPqcKp6rH4pcVIvseHxpNvVs9hjTk8PSJGVizT5j9UE=</Enk
ripsiPartai><NoSurat>125</NoSurat><HashSuara>0C0DC6E6F
9F506DDB7D4B512C0204BA456E239563F74BCF6C18E57C9813C9FF
E</HashSuara><Signature>STttPQCX66XtLShHFKoM1FcDZTDIpf
OkBC0BkD3u3RKmQ76dEK68MCT5CsmqRmasbOPc5EQUmm7Jzz6a/8uQ
GYnIkAK57ZAAjBmL9wzXYKvtx+PwPapXK1Pi0nYw0ZXuTpiczqB3Qf
L6RcMISl863Ht0WJ5yn6XxBhm7m0sOtfw=</Signature><additio
nalParameter>TPS</additionalParameter></Send></soap:Bo
dy></soap:Envelope>
```

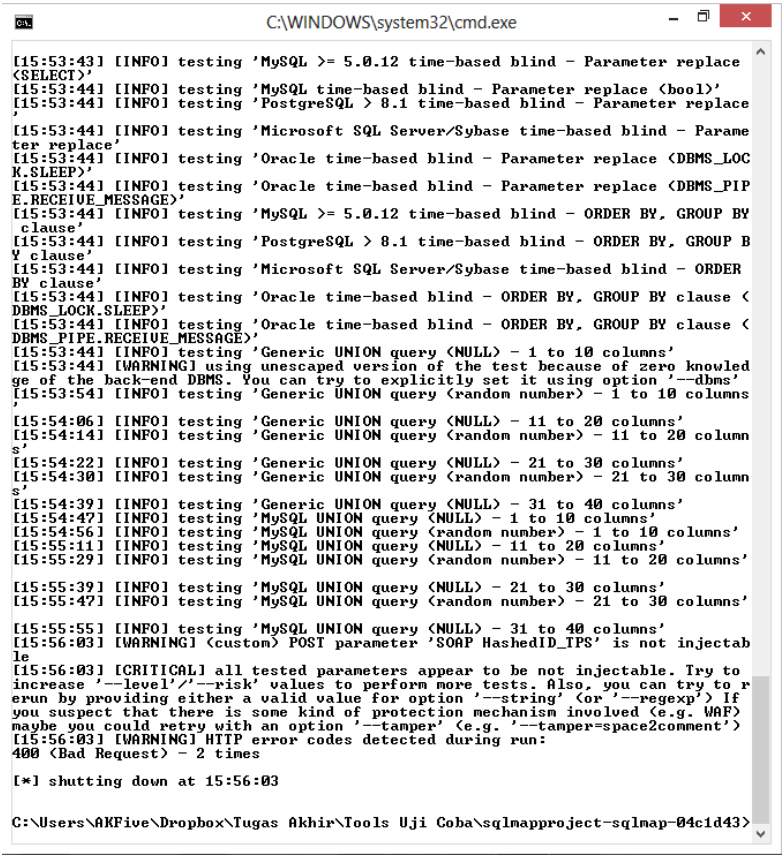
Kode Sumber A.6. SOAP *request web method* send



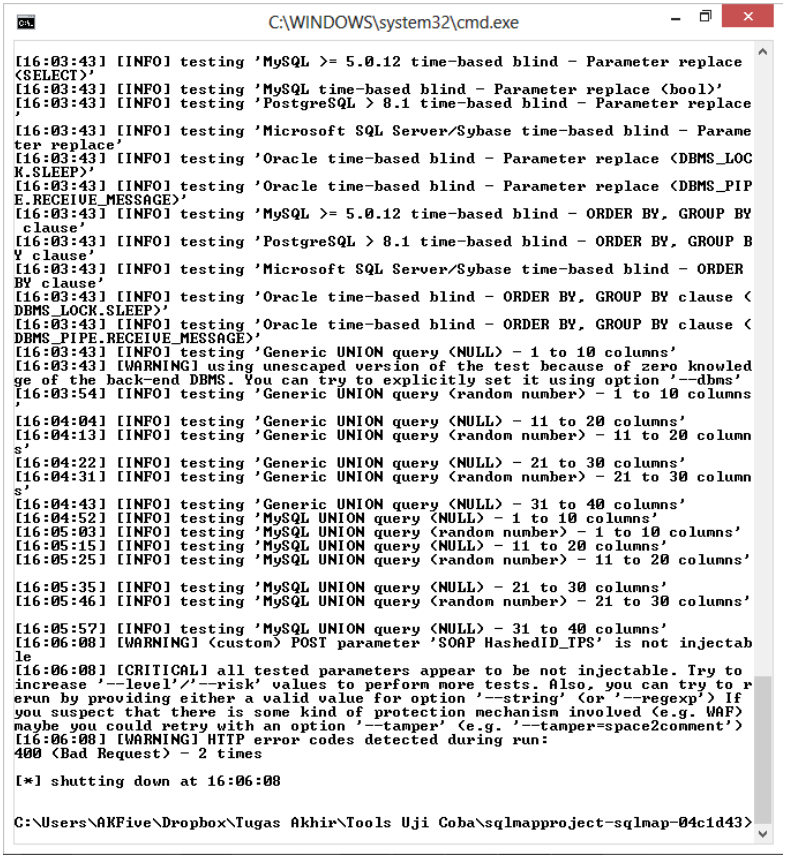
Gambar A.10. Hasil uji *sql inject* pada *web method* auth



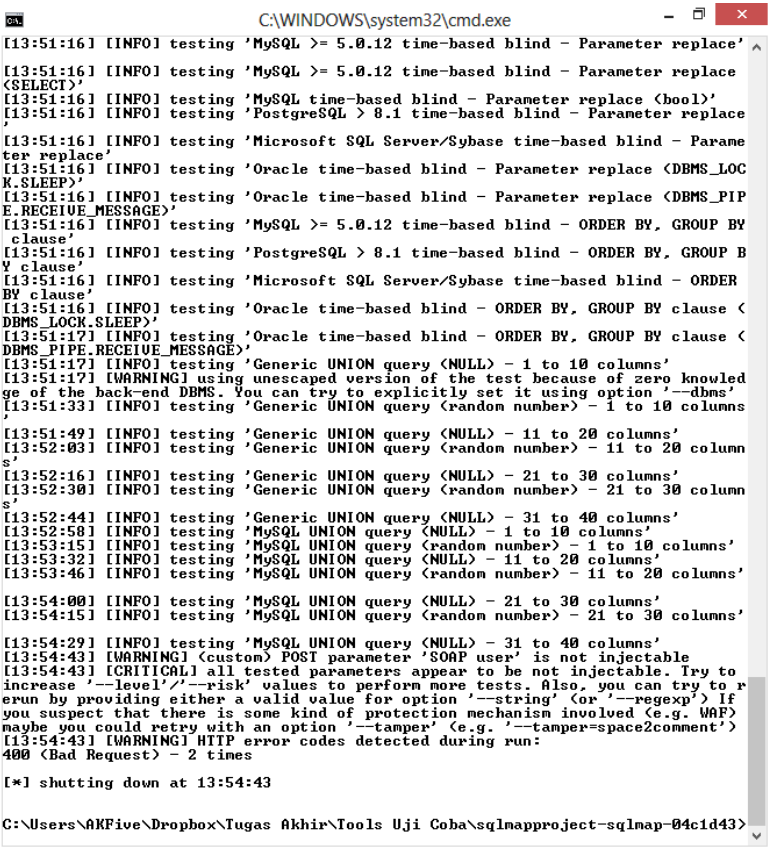
Gambar A.11. Hasil uji *sql inject* pada *web method* close



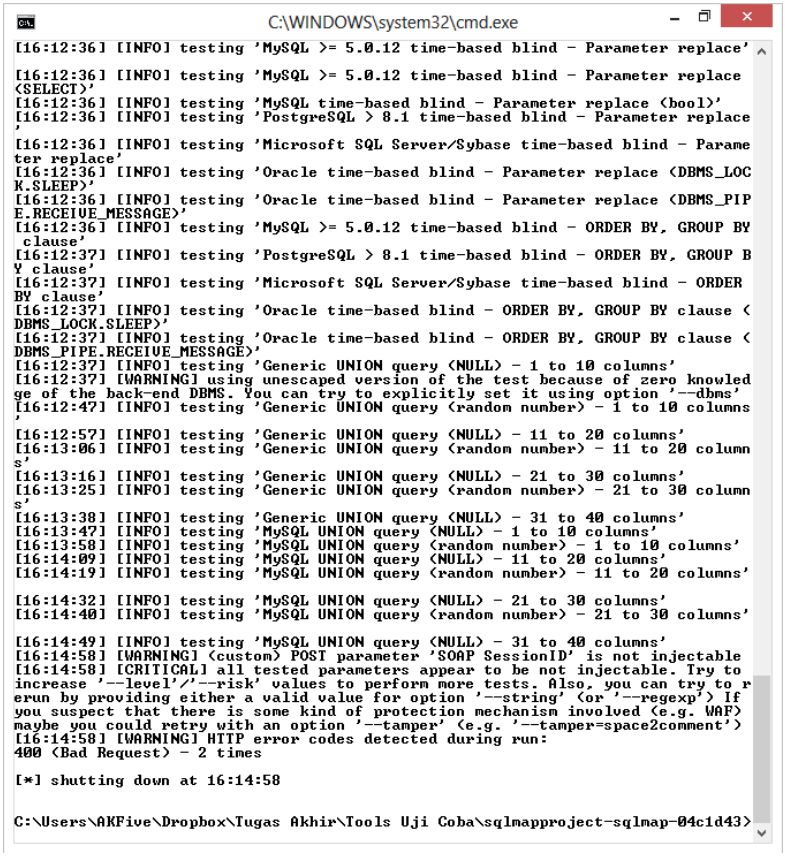
Gambar A.12. Hasil uji *sql inject* pada *web method* getSession



Gambar A.13. Hasil uji *sql inject* pada *web method* hello



Gambar A.14. Hasil uji *sql inject* pada *web method* login



Gambar A.15. Hasil uji *sql inject* pada *web method* send

Tabel A.4. Tabel Hasil Uji SQLMap pada *Web Service*

No.	Nama Web Method	Nama File Request	Parameter	Level	Status Penetrasi
1	Auth	Soapauth.txt	HashedID_TPS	4	Gagal
2	Close	Soapclose.txt	SessionID	4	Gagal
3	GetSession	Soapgetsession.txt	SessionID	4	Gagal
4	Hello	Soaphello.txt	HashedID_TPS	4	Gagal
5	Login	soaplogin.txt	User	4	Gagal
6	Send	Soapsend.txt	SessionID	4	Gagal

BIODATA PENULIS

Astandro Koesriputranto, lahir di Jakarta, pada tanggal 28 Juni 1993. Penulis menempuh pendidikan mulai dari SD Islam Tuban (1999-2005), SMP Negeri 1 Tuban (2005-2008), SMA Negeri 1 Tuban (2008-2011) dan S1 Teknik Informatika ITS (2011-2015). Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTC). Diantaranya adalah menjadi staff departemen Kewirausahaan dan Minat Bakat Himpunan Mahasiswa Teknik Computer-Infotmatika ITS 2012-2013. Penulis juga aktif dalam kegiatan kepanitiaan Schematics. Diantaranya penulis pernah menjadi Staff Perlengkapan Schematics 2012 dan Wakil Koordinator Perlengkapan dan Transportasi Schematics 2013. Selama kuliah di teknik informatika ITS, penulis mengambil bidang minat Komputasi Berbasis Jaringan (KBJ). Penulis pernah menjadi asisten dosen mata kuliah Teori Graf dan Otomata. Komunikasi dengan penulis dapat melalui email: **astandro.tc@gmail.com**.