TUGAS AKHIR – KI141502

IMPLEMENTASI KONTROL INTEGRITAS E-KIOSK UNTUK PENGAMANAN SISTEM PEMUNGUTAN SUARA SECARA ELEKTRONIK (*E-VOTING)*

ISHOM MUHAMMAD DREHEM
NRP 5111100153

Dosen Pembimbing:
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Baskoro Adi Pratomo, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

FINAL PROJECT – KI141502

IMPLEMENTATION OF E-KIOSK INTEGRITY CONTROL FOR THE SECURITY OF ELECTRONIC VOTING (E-VOTING) SYSTEM

ISHOM MUHAMMAD DREHEM
NRP 5111100153

Supervisor:
Prof. Ir. Supeno Djanali, M.Sc., Ph.D.
Baskoro Adi Pratomo, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA 2016

LEMBAR PENGESAHAN

IMPLEMENTASI KONTROL INTEGRITAS E-KIOSK UNTUK PENGAMANAN SISTEM PEMUNGUTAN SUARA SECARA ELEKTRONIK (*E-VOTING*)

TUGAS AKHIR

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

Oleh:
ISHOM MUHAMMAD DREHEM
NRP: 5111 100 153
Disetujui oleh Pembimbing Tugas Akhir:

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

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

SURABAYA
JANUARI 2016

IMPLEMENTASI KONTROL INTEGRITAS E-KIOSK UNTUK PENGAMANAN SISTEM PEMUNGUTAN SUARA SECARA ELEKTRONIK (E-VOTING)

Nama : Ishom Muhammad Drehem
NRP : 5111100153
Jurusan : Teknik Informatika FTIf ITS Surabaya
Dosen Pembimbing I : Prof. Ir. Supeno Djanali,M.Sc.,Ph.D.
Dosen Pembimbing II : Baskoro Adi Pratomo, S.Kom.,M.Kom.

Abstrak

Pemungutan suara dalam pemilu di Indonesia masih dilakukan secara manual,yaitu menggunakan media kertas. Dalam sistem tersebut, terjadi risiko kesalahan yang tinggi dalampenghitungan suara mengingat surat suara yang diproses terbilang banyak.Selain itu, rawan terjadi kecurangan terhadap jumlah suara demi memenangkan kelompok atau golongan tertentu.Akibatnya,pelaksanaan pemilu menjadi tidak sesuai dengan asas yang berlaku dan hasilnya tidak akurat.Untuk mengatasinya, dirancanglah sistem pemungutan suara yang lebih modern, yang disebut dengan sistem pemungutan suara secara elektronik (e-voting).Sistem e-voting menggunakan e-kioskyang memudahkan pemilih dalam memberikan suaranya karena tidak perlu mencoblos dan memasukkan kertas kedalam kotak kertas suara.Selain itu, faktor integritas data dan keamanan data pemilih lebih aman karenamenggunakan metode enkripsi dan transmisi data yang aman.

Pada tugas akhir ini, penulis menggunakan dua skenario, yaitu fungsionalitas dan faktor keamanan data. Dari sisi fungsionalitas, sistem aplikasiyang dibuat sesuai dengan kebutuhan pengguna yaitu dari sisi admin. Terdapat enam fungsionalitas, yaitu cek koneksi untuk servis pengirman SMS, generate token untuk mendapatkan token melalu servis pengiriman SMS,, muat kunci untuk metode pengiriman data

menggunakan XML, rekap data pemilih untuk memantau hasil pemilihan partai dan anggota legislatif pada hari itu juga, tanda tangan saksi dan KPPS, serta kirim data suara dari server lokal menuju server pusat. Sedangkan dari sisi keamanan data, sistem aplikasi mengamanakan data menggunakan metode hash dan enkripsi data untuk dikirimkan menuju server pusat. Yang diamankan adalah data suara basis data. Kemungkinan besar basis data tersebut dapat digunakan oleh admin untuk mengubah data.

Skenario yang dirancang tersebut terbukti dapat mengamankan sistem e-voting. Diharapkan skenario tersebut dapat diterapkan pada penggunaan sistem e-voting yang sebenarnya di masa mendatang.

Kata kunci: Pemilu, Electronic Voting (E-Voting), E-Kiosk, Metode Enkripsi, Pengamanan Aplikasi.

IMPLEMENTATION OF E-KIOSK INTEGRITY CONTROL FOR THE SECURITY OF ELECTRONIC VOTING (E-VOTING) SYSTEM

Name : Ishom Muhammad Drehem
NRP : 5111100153
Department :Informatics Engineering,FTIf ITS
AdvisorI : Prof. Ir. Supeno Djanali,M.Sc.,Ph.D.
AdvisorII : Baskoro Adi Pratomo,S. Kom.M. Kom.

Abstract

Voting in Indonesian elections is still carried out manually by using paper as the media. In such system, there is a high risk of errors in counting the huge numbers of ballots. In addition, electoral fraud to bring the winning for certain groups often occurs. As a result, the implementation of election is incompatible with the applied principles and the result is not accurate. To overcome this, the voting system is designed to be more modern called as electronic voting (e-voting). E-voting system uses e-kiosk that allows voters to cast their vote because they do not need to be cast and put the paper into a ballot box. In addition, data integrity and security factors of voters are more secure because it uses secure encryption method and data transmission.

In this thesis, the author used two scenarios, namely functionality and data security test. In terms of functionality, the application system made was in accordance with the needs of users in terms of admin. There were six functionalities, they were connection check for SMS delivery service, generate token to get token through SMS delivery service, key load for the method of sending data using XML, voters recap data to monitor the results of party and member of legislative elections on the same day, the witness and KPPS signature, and sending voting data from the local server to the central server. While in terms of data security,

the application system secures the data using the hash method and encryption data to be sent to the central server. Voting-based data was secured. The database can be used by admin to change the data.

The designed scenario was proven to be able to secure e-voting system. It is expected that the scenario can be implemented in the use of actual e-voting system in the future,

Keywords: Election, Electronic Voting (E-Voting), E-Kiosk, Encryption Method, Application Security

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **"*Implementasi Kontrol Integritas E-Kiosk untuk Pengamanan Sistem Pemungutan Suara Secara Elektronik (E-Voting)*"**.

Harapan dari penilis, semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata bagi kampus Teknik Informatika, ITS Surabaya, dan bangsa Indonesia.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT, karena atas limpahan rahmat-Nya, penulis diberikan kemudahan dan kelancaran dalam mengerjakan Tugas Akhir ini.
2. Ummi (Maryam Maziun) dan Abi (Muhammad Shaleh Drehem) tercinta, Kakak (Shaleh Muhammad Drehem) Adik (Bassam Muhammad Drehem dan Mustafa Muhammad Drehem) tersayang, serta keluarga besar yang memberikan dukungan dan moral, spritual, semangat dan perhatian, selalu setia dan sabar dalam menghadapi curhatan dari penulis, serta selalu memberikan doa yang tiada habisnya yang dipanjatkan untuk penulis.
3. Prof. Ir. Supeno Djanali, M.Sc., Ph.D. dan Baskoro Adi Pratomo, S.Kom., M.Kom selaku dosen pembimbing penulis yang telah memberikan banyak arahan dan bantuan sehingga penulis dapat menyelesaikan Tugas Akhir ini.
4. Bapak Dr.Darlis Herumurti selaku ketua jurusan Teknik Informatika ITS, dan segenap dosen Teknik Informatika yang telah memberikan ilmu kepada penulis.

5. Bapak dan Ibu staf Tata Usaha yang telah memberikan bantuan dan kemudahan kepada penulis selama masa perkuliahan di Teknik Informatika ITS.
6. Teman-teman satu tim dalam penyelesain tugas akhir ini dengan penulis (Karsono Puguh Nindyo Cipto, Danang Prawira Nugraha, dan Astandro Koesriputranto) dan temen-temen satu angkatan yang telah memberikan bantuan dan dukungan terhadap penulis. Terima kasih telah berjuang bersama dengan penulis. Teman teman satu BEM ITS (Hublubelle, Hubluable, dan Hublukece dll) Kabinet Mahakarya, Kabinet Transformasi dan Kabinet Muda Bersahabat.
7. Terimakasih pula buat admin NCC, user TA NCC dan adik adik yang menyemangati saya agar segera keluar dari lab NCC sebagai user TA, kalian luar biasa ngusirnya.
8. Pihak-pihak yang lain yang tidak bisa penulis sebutkan satu-persatu disini yang juga telah memberikan semangat dan membantu penulis menyelesaikan tugas akhir ini.

Kesempurnaan tentu masih belum tercapai pada tugas akhir ini. Karena itu, penulis mengharapkan saran dan kritik dari pembaca untuk perbaikan selanjutnya.

Surabaya, Januari 2016

Ishom Muhammad Drehem

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, serta sistematika penulisan.

1.1 Latar Belakang

Indonesia merupakan negara yang menganut sistem politik demokrasi, Pemilihan umum (Pemilu) merupakan salah satu ciri suatu negara demokrasi untuk menentukan calon pemimpin. Pemilihan Umum (Pemilu) yang disebut juga pesta demokrasi diselenggarakan setiap lima tahun sekali. Selain pemilihan kepala negara, di Indonesia juga terdapat pemilihan kepala daerah, baik bupati, gubernur, anggota Dewan Perwakilan Rakyat (DPR) dan sampai pemilihan kepala desa juga menggunakan sistem pemungutan suara.

Pemungutan suara dalam pemilu di Indonesia masih dilakukan secara manual dengan menggunakan media kertas. Para calon pemilih yang telah terdaftar sah sebagai pemilih mendatangi Tempat Pemungutan Suara (TPS), kemudian mencoblos atau mencontreng media kertas sebagai tanda bahwa pemilih telah melaksanakan haknya untuk memberikan suaranya terhadap pemilu. Setelah selesai dilakukan perhitungan suara per TPS yang kemudian akan dilaporkan ke pusat lalu data dikumpulkan sehingga terhitung secara keseluruhan. Pemungutan suara secara manual tersebut memiliki kelemahan-kelemahan, antara lain:

1. Ditinjau dari lama proses pelaksanaannya secara keseluruhan, maka proses pemilu di Indonesia terhitung cukup lama hingga mencapai beberapa minggu bahkan mencapai berbulan-bulan,
2. Terjadinya kesalahan dalam proses pemilu, bahkan banyak terjadinya tindak pencurangan sehingga data tidak akurat,
3. Rusaknya media kertas yang akan digunakan sehingga menjadi masalah mengenai sah tidaknya kertas digunakan.

Padahal kertas suara yang disediakan oleh Komisi Pemilihan Umum (KPU) terbatas tidak boleh melebihi jumlah pemilih terdaftar ditambah cadangannya agar dapat mengurangi tindak kecurangan,

Dengan banyaknya masalah yang timbul dari proses pemungutan suara yang secara manual, maka banyak gagasan yang muncul untuk menggantikan pemungutan suara secara manual dengan yang lebih modern, yang dikenal dengan sebutan *e-voting (electronic voting).E-Voting* dirancang untuk memudahkan pemilih memberikan suaranya dengan tidak mengabaikan asas pemilu yang berlaku, yaitu Langsung Umum Bebas Rahasia (LUBER). *E-voting*diharapkan dapat menerapkan pelaksanaan pemilu yang lebih mudah, cepat, akurat, dan menghemat biaya, dengan tetap melindungi warga negara sebagai pemilih yang memberikan suaranya serta menjamin kerahasiaan dan keabsahan hasil pelaksanaan pemilu[1].

E-voting dinilai cukup efektif untuk mengatasi permasalahan yang terjadi pada sistem konvensional.Dilihat dari segi waktu, untuk menghitung hasil pemilihan dapat dilaksanakan secara langsung dan data hasil pemilih tersebut langsung masuk dan diproses di *server*. Dilihat dari segi biaya juga lebih murah karena pada sistem konvensional masih harus mendistribusikan surat dan kotak suara di seluruh area pemilihan, tetapi dengan menggunakan sistem elektronik dapat menggunakan jaringan komputer, selain itu komunikasi akan lebih mudah dibandingkan dengan sistem konvensional[2].

Dalam perancangan tugas akhir ini, faktor keamanan harus diperhatikan untuk mendukung pelaksanaan yang berdasarkan asas Pemilu yang berlaku. Aspek yang terdapat dalam faktor keamanan antara lain *authentication*, *confidentiality* dan *integrity*. Tanpa adanya hal tersebut maka sistem *e-voting* mudah mengalami serangan yang dapat mengacaukan sistem itu sendiri. Penulis mengimplementasikan aspek *integrity* dan *confidentiality*pada pemilu dengan menggunakan *e-kiosk* menjadi alternatif utama untuk pemilihan dengan mengenkripsi beberapa

data pemilih dan terbukti berhasil meningkatkan kemanan pada sistem tersebut. Penulis juga akan menerapkan sesuai dengan kondisi pemilu yang berada di Indonesia. Harapannya sistem *e-voting* dapat meningkat animo memilih masyarkat dan peningkatan kepercayaan masyarakat terhadap hasil pemilu dan mengurangi kecurangan yang terjadi selama pelaksanaan pemilu.

1.2 Rumusan Masalah

Berikut beberapa hal yang menjadi rumusan masalah dalam Tugas Akhir ini :

1. Bagaimana *E-Voting* dapat menjadi solusi terhadap permasalahan yang ada pada pemilihan suara secara manual?
2. Bagaimana data hasil suara pemilihan umum dapat diproses dalam sistem *E-Voting* yang dibuat?
3. Bagaimana membuat aplikasi *E-Voting* yang dapat dijalankan sesuai dengan kebutuhan pemilu untuk saat ini?
4. Bagaimana mengamankan data ketika admin memalsukan data di dalam basis data?

1.3 Batasan Masalah

Berikut beberapa hal yang menjadi batasan masalah dalam pengerjaan Tugas Akhir ini:

1. Pembuatanaplikasi*E-Voting* untuk pemilihan umum anggota DPRD Kota Surabaya.
2. Bahasa pemrograman yang dipakai adalah C# dengan *framework* .NET.
3. Sistem *E-Voting* yang digunakan adalah berbasis *desktop*.
4. Keamanan berfokus pada aspek *confidentiality* dan *integrity.*
5. Pengujian dilakukan pada keamanan aplikasi, pengolahan data dalam sistem, serta penyimpanan dan pengiriman data suara.
6. Sistem kriptografi yang digunakan adalah sistem kriptografi RSA dan *hash* yang digunakan adalah SHA-256.

1.4 Tujuan

Tujuan dari penulisan Tugas Akhir ini adalah sebagai berikut.

1. Membuat aplikasi sistem *E-Voting*yang menekankan faktor keamanan.
2. Membuat media pemungutan suara secara elektronik yang berfungsi untuk menampung hasil pemungutan suara sehingga dapat menggantikan media kertas sebagai media pemungutan suara.
3. Hasil suara dari pemilihan umum akan lebih mudah dan cepat dalam proses penghitungan suara.
4. Mengamankan data didalam basis data secara aman menggunakan metode enkripsi dan hash.

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini adalah:

1. Mendapatkan pemahaman implementasi asas Pemilu ke dalam sistem *e-voting* dan masa peralihan menggunakan sistem tersebut.
2. Mendapatkan wawasan tambahan tentang implementasi pemilu dalam suatu aplikasi teknologi informasi.
3. Mengurangi kecurangan yang terjadi pada sistem pemilu dengan memberikan keamanan pada data.

1.6 Metodologi

Metodelogi yang digunakan untuk menyelesaikan Tugas Akhir ini sebagai berikut:

a. Studi Literatur

Pada tahap ini dilakukan studi terhadap permasalah dari referensi yang tersedia dari berbagai sumber. Mulai dari pembuatan aplikasi *E-Voting* menggunakan bahasa C# , pengamanan sistem aplikasi *E-Voting* dan Pengiriman database voter dari *server* lokal menuju *server* pusat dengan

aman dan *secure* sampai pada pemahaman materi yang berhubungan dengan kebutuhan pada Tugas Akhir ini.

b. Perancangan Aplikasi

Pada tahap ini dilakukan perancangan terhadap sistem yang akan dibuat, desain basis data, arsitektur sistem sampai dengan desain antarmuka serta perancangan fitur fitur aplikasi.

c. Implementasi

Tahap ini merupakan tahap dalam pembuatan aplikasi berdasarkan rancangan yang telah dibuat sebelumnya. Mulai dari implementasi cara menggunakan aplikasi *E-Voting* sampai pemilih atau *voter* meninggalkan TPS dan pengiriman data suara menuju *server* pusat.

d. Pengujian dan Evaluasi

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibuat berdasarkan tujuan pembuatan program dan mengidentifikasi masalah-masalah yang mungkin muncul. Dalam pengujian, aplikasi dipasang ke dalam perangkat yang sebenarnya sehingga dapat diketahui tingkat kenyamanan voter dalam memilih dan keamanan data *E-Voting* dari program yang telah dibuat. Selain itu pada tahap ini juga dilakukan perbaikan apabila aplikasi kurang sesuai dengan tujuan awal pembuatan.

e. Penyusunan Buku Tugas Akhir

Tahap ini merupakan penyusunan laporan yang memuat dokumentasi mengenai pembuatan aplikasi dan hasil dari aplikasi yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan

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

2. Bab II. Tinjauan Pustaka

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

3. Bab III. Analisis dan Perancangan

   Bab ini 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 tentang implementasi dari desain sistem yang dibuat pada tahap perancangan.

5. Bab V. Uji Coba dan Evaluasi

   Bab ini berisi hasil evaluasi aplikasi dengan menggunakan aplikasi yang dibangun. Juga disertakan analisis dari hasil evaluasi perangkat lunak.

6. Bab VI. Kesimpulan dan Saran

   Bab ini 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 ini 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.

1.1 *E-Voting*

Pengertian E-Voting secara umum adalah sebuah pengambilan suara untuk mencari keputusan sebuah pilihan yang ditentukan oleh banyak orang yang diambil secara *voting*dengan menggunakan media komputer yang telah diberi sistem aplikasi *voting*. Prosedur dari pelaksanaan *voting* telah banyak dilakukan penelitian diberbagai instansi ataupun berbagai negara yang dirancang dengan memperhatikan kerahasiaan dan keabsahan dari proses pelaksanaan *voting* tersebut, sehingga terjaga keakuratan data dari hasil pemungutan suara. Dari setiap aplikasi *E-Voting* yang telah dibuat tidak memiliki standar cara penggunaan yang ditetapkan, semakin mudahnya aplikasi digunakan dengan tingkat keamanan data yang tinggi, maka semakin baik aplikasi yang dibuat. Secara umum *voting* memiliki tiga tahapan proses pelaksaaan dalam pengambilan suara yaitu:

1. Pendaftaran pada awalnya dilakukan pendaftaran untuk setiap calon pemilih dipastikan terlah terdaftar dan memiliki hak akses untuk dapat memiliki hak untuk memberikan suaranya.
2. Pemungutan suara, calon pemilih terdaftar dan memilik hak akses untuk memilih, maka dapat memberikan suaranya dalam voting.
3. Perhitungan suara pada tahapan ini menampilkan hasil suara yang telah masuk dalam *voting* , dalam aplikasi *E-Voting* hasil suara yang masuk dapat dilihat secara langsung, mana kandidat yang mendapatkan hasil suara terbanyak[3].

Gambar 1.1Ilustrasi dalam *E-Voting*

Pada Gambar **1.1** Ilustrasi dalam E-Voting tersebut adalah seseorang yang menggunakan e-kiosk, proses tersebut memudahkan pemilih dalam memilih.

1.1.1 Kelebihan*E-Voting*

Beberapa kelebihan dari sistem E-Voting adalah sebagai berikut:

1. Dalam pemungutan hasil suara dibutuhkan sebuah kecepatan informasi data,sehingga pemilu yang dilakukan tidak memakan banyak waktu guna mendapatkan hasil pengumuman perolehan terbanyak dari pemungutan suara. Dengan menggunakan teknologi komputer diharapkan masalah kelambatan informasi data dapat diselesaikan,
2. Tingkat keamanan ataupun keakuratan data untuk saat ini semakin tinggi dengan banyaknya pengembangan keamanan komputer,
3. Untuk pengembangan telah banyak dilakukan dibanyak daerah, universitas, lembaga ataupun di berbagai negara, yang mengacunya pada pemanfaatan teknologi komputer yang ramah lingkungan dengan tingkat akurasi yang tinggi,
4. Dengan dilakukan banyak penelitian E-Voting dirancang agar cara operasional yang mudah, sehingga dapat digunakan oleh orang banyak dari berbagai latar belakang,

5. Dengan E-Voting selain kecepatan data, keakuratan ataupun keamanan data yang semakin baik E-Voting juga mampu menampilkan data yang autentik[4].

1.1.2 Kelemahan *E-Voting*

Selain banyak kelebihan dari sistem *E-Voting* tidak tertutupkemungkinan sebuah kelemahan-kelemahanya. Banyak diberbagai organisasi ataupun negara yang masih belum menerima teknologi *E-Voting* sebagai pengganti sistem pemungutan suara manual. Beberapa kelemahan yang sering menjadi alasan belum dapat diterimanya *E-Voting* sebagai pengganti pemungutan suara diantaranya adalah sebagai berikut :

1. Walaupun untuk teknologi keamanan sistem *E-Voting* terus dalam pengembangan untuk keamanan data yang tinggi dan layak digunakan dengan mampu mempertanggung jawabkan akan keamanan datanya namun masih banyak pihak yang meragukan akan hal itu,
2. Pola pikir masih sederhana denga latar belakang pendidikan calon pemilih yang beragam, walaupun teknologi yang dirancang dengan sistem yang mudah digunakan namun harus dilakukan sosialisasi terhadap masyarakat. Yang terkadang membutuhkan waktu agar *E-Voting* yang diperkenalkan dengan masyarakat dapat diterima dengan baik,
3. Sumber daya manusia yang handal di bidang IT kurang termasuk pemindahan sistem dari sistem lama yang manual dengan sistem baru yang menggunakan teknologi komputer membutuhkan usaha yang cukup besar,
4. Masalah kehandalan dari suatu perangkat lunak atau perangkat keras. Masih menjadi masalah dalam keandalan perangkat lunak, artinya jika dipakai beberapa kali kadangkala masih adanya kesalahan dalam perhitungan jumlah suara. Begitu pula dalam hal perangkat kerasnya

seperti peralatan kadang-kadang tidak dapat merespon dengan cepat,

5. Masalah faktor manusia seperti petugas pemilu, kadang-kadang petugas pemilu belum paham benar dalam mengoperasikan perangkat teknologi yang dipakai dalam pemilu. Dan selanjutnya adalah faktor pemilihnya sendiri yang belum paham atau belum pernah melakukan pemilihan dengan menggunakan teknologi *E-Voting*sehingga terjadi kesalahan dalam mengoperasikannya, akibatnya banyak kegagalan dalam melakukan pemilihan Kandidat yang dituju,
6. Masih terbatasnya tempat akses. Hal ini juga dipicu oleh masih belum adanya infrastruktur teknologi informasi dan komunikasi yang baik di daerah,
7. Sumber-sumber yang berpotensi untuk diganggu/dirusak.
   a. Serangan dapat dilakukan secara Electronik (*Hacking*).
   b. Serangan dapat menggunakan perangkat lunak jahat (*Malicious User Interface*).

Masih banyak hal yang membuat sistem *E-Voting* belum dapat dilaksanakan dalam skala pemerintah pusat, masih banyak prasangka *negative*mengenai sistem *E-Voting*. Dan lembaga pemerintahan juga terkenal dengan lemahnya atau miskinnya budaya TI di dalam pengorganisasian pemerintahannya sehingga prasangka negatif tersebut khususnya terkait dengan transparansi atau keterbukaan. Pada beberapa negara dengan tingkat korupsi yang cukup tinggi seperti Indonesia masalah transparansi merupakan hal yang sering dihindari oleh para aparat pemerintah yang korup. Mereka tidak senang apabila penggunaan sistem *E-Voting* akan menjadikan proses pemilu semakin transparan sehingga kedudukan mereka di pemerintahan akan terancam[5].

Solusi yang dapat diambil untuk menyelesaikan hambatan-hambatan yang sering terjadi dalam pengimplementasian *E-Voting* adalah sebagai berikut :

1. Mengubah pola pikir melalui diklat, seminar, video, berita.
2. Menambah jumlah *server* sesuai dengan kebutuhan dan menggunakan aplikasi berbasis *open source*,
3. Menambah jumlah sumber daya manusia bidang TI yang handal,
4. Mengintegrasikan basis datadan sistem aplikasi secara terpadu,
5. Dukungan pemerintah dengan meningkatkan tingkat literasi komputer dan internet bagi aparat pemerintahan.

Kondisi perkembangan teknologi *E-Voting*selalu berubah seiring dengan berkembangnya teknologi yang sangat cepat, kendala-kendala *E-Voting* yang pernah terjadi diberbagai daerah atau negara yang menjalankan sistem *E-Voting* ini terus dilakukan pembenahan yang diharapkan *E-Voting* benar-benar dapat berjalan dengan sempurna. Dengan menerapkan sistem *E-Voting* dalam pemilu ini diharapkan juga dapat menjadikan alternatif yang ramah lingkungan dan hematnya biaya yang dikeluarkan dibanding dengan pemilu secara manual. Penggunaan *E-Voting* di Indonesia telah banyak digunakan namun masih dalam skala terbatas yang masih dalam lingkup organisasi, perusahaan, untuk dipemerintah menggunakan skala kecil, yaitu dusun atau desa[6].

1.2 Basis Data dan MySQL

Basis data merupakan sekumpulan data-data yang sangat kompleks yang memiliki hubungan satu dengan yang lainnya. Di dalam sebuah *database* data diatur dengan menggunakan sebuah pengelompokan dengan *table*. Pada tabel sendiri juga masih dikelompokkan menjadi beberapa bagian yang berupa *field-field.*

My Structure Query language (MySQL) adalah sebuah sistem manajemen database relasi *(relational database management system)* yang bersifat *open source*, artinya MySQL boleh di download oleh siapa saja,baik versi kode program aslinya *(source code program)* maupun versi binernya *(executable program)* dan bisa digunakan secara (relatif) gratis

baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer. Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah dioleh atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

DBMS adalah satu set program untuk mengakses data yang biasanya menggunakan *QueryStructured Query Language* (SQL). Untuk setiap DBMS yang memiliki konektor sebagai *driver*agar dapat diakses oleh bahasa pemrograman dapat digunakan sebagai tempat penyimpanan data yang pERsisten[7].

1.2.1 Kelebihan MySQL

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas, MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X *Server*, Solaris, Amiga, dan masih banyak lagi,
2. Perangkat lunak sumber terbuka, MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis,
3. *Multi-user*, MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik,
4. *Performance tuning*, MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu,
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed/unsigned integer, float, double, char, text, date, timestamp,* dan lain-lain,
6. Perintah dan fungsi, MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*),

7. Keamanan, MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses pengguna dengan sistem perizinan yang mendetail serta sandi terenkripsi,
8. Skalabilitas dan pembatasan, MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya,
9. Konektivitas, MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unixsocket* (UNIX), atau *Named Pipes* (NT),
10. Lokalisasi, MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya[8].

1.2.2 Kunci Primer dan Kunci Sekunder

Pada konsep basis data relasional, biasanya sebuah tabel memiliki kunci primer (*primary key*) yang memiliki nilai unik (tidak ada yang sama satu dengan yang lainya). Namun pada praktiknya sebuah tabel dapat tidak memiliki kunci primer. Kunci primer merupakan satu atau lebih kolom yang berisi dari nilai kolom-kolom yang menjadi kunci primer adalah unik sehingga dapat menjadi tanda bagi setiap baris data dalam mengaksesnya. Pada basis data relasional juga dikenal kunci asing atau *foreign key*. Kunci asing adalah sebuah kolom yang merupakan kunci primer dari tabel lain[9].

1.2.3 Tabel dan Struktur Relasi Basis Data

Tabel adalah media untuk menyimpan data yang telah diolah dan mempunyai sesuatu tema tertentu, misalnya tabel yang digunakan untuk menyimpan data tentang pemilih, berisikan nomor induk pemilih, nama pemilih, alamat pemilih dan status pemilih yang disimpan ke dalam *field-field* tertentu.*Field* adalah

tempat dimana data atau informasi dalam kelompok sejenis dimasukan.*Record* adalah data lengkap dalam jumlah tunggal yang tersimpan dalam bentuk baris *horizontal* pada tabel. Dalam satu tabel dapat diinputkan beberapa *record* sekaligus.Hubungan data antar tabel dalam basis datajuga disebut dengan relasi. Relasi digunakan untuk meringkas data yang ada dalam basis datasehingga penggunaan data akan lebih *fleksibel* juga dalam penggunaan memori penyimpanan pun akan lebih efesien. Manfaat dari relasi data adalah sebagai berikut :

1. Penyimpanan data lebih efesien karena penulisan tidak dilakukan secara berulang-ulang,
2. Tingkat efektifitas dan konsisten data lebih terjamin,
3. Data mudah dipantau atau dikontrol dalam basis data[8].

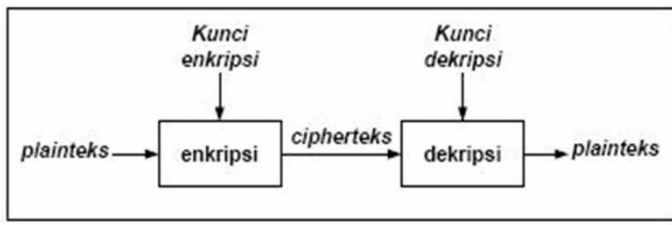
Normalisasi adalah suatu teknik untuk mengorganisasi data ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi.Tujuan normalisasi adalah:

1. untuk menghilangkan kerangkapan data,
2. untuk mengurangi kompleksitas,
3. untuk mempermudah pemodifikasian data.

1.3 Sistem Kriptografi RSA

RSA merupakan algoritma kriptografi asimetri, dimana kunci yang digunakan untuk mengenkripsi berbeda dengan yang digunakan untuk mendekripsi. Kunci yang digunakan untuk mengenkripsi disebut dengan kunci public, dan yang digunakan untuk mendekripsi disebut dengan kunci privat. RSA adalah salah satu algoritma kriptografi yang menggunakan konsep kriptografi kunci publik. RSA membutuhkan tiga langkah dalam prosesnya, yaitu pembangkitan kunci, enkripsi, dan dekripsi. Proses enkripsi dan dekripsi merupakan proses yang hampir sama. Jika bilangan acak yang dibangkitkan kuat, maka akan lebih sulit untuk melakukan cracking terhadap pesan. Parameter kuat tidaknya suatu kunci terdapat pada besarnya bilangan acak yang

digunakan[9]. Gambaran sistem kriptografi RSA terlihat pada Gambar 1.2.



Gambar 1.2Sistem kriptografi RSA

Algoritma RSA dijabarkan pada tahun 1976 oleh tiga orang: Ron Rivest, Adi Shamir dan Len Adleman dari Massachusetts Institute of Technology. Huruf "'RSA'" itu sendiri berasal dari inisial nama mereka (Rivest - Shamir -Adleman). Clifford Cocks, seorang matematikawan Inggris yang bekerja untuk GCHQ, menjabarkan tentang sistem ekuivalen pada dokumen internal di tahun 1973. Penemuan Clifford Cocks tidak terungkap hingga tahun 1997 karena alasan "top-secret classification". Algoritma RSA dipatenkan oleh Massachusetts Institute of Technology pada tahun 1983 di Amerika Serikat sebagai US patent 4405829. Paten tersebut berlaku hingga 21 September 2000. Setelah bulan September tahun 2000, paten tersebut berakhir, sehingga saat ini semua orang dapat menggunakannya dengan bebas.RSA adalah sebuah algoritma berdasarkan skema kriptografi public-key. Lebih jauh, RSA adalah algoritma yang mudah untuk diimplementasikan dan dimengerti. algoritma RSA adalah sebuah aplikasi dari sekian banyak teori seperti extended Euclid algorithm, euler's function sampai fermat theoreme.

Konsep fundamental dari Kriptografi Kunci Publik ditemukan oleh Whitfield Diffie dan Martin Hellman, dan secara terpisah oleh Ralph Merkle. Sedangkan konsep dasar Kriptografi Kunci Publik terletak pada pemahaman bahwa kunci selalu berpasangan: kunci enkripsi dan kunci dekripsi. Juga perlu diingat

bahwa sebuah kunci tidak dapat dibangkitkan dari kunci lainnya. Pemahaman kunci enkripsi dan dekripsi sering disebut sebagai kunci publik dan kunci privat. Seseorang harus memberikan kunci publiknya agar pihak lain dapat mengenkripsi sebuah pesan. Dekripsi hanya terjadi jika seseorang mempunyai kunci privat[10].

Pertama-tama penerima pesan sebagai pendekripsi menghasilkan kunci publik dan kunci privat miliknya. Pasangan kunci tersebut dihasilkan dengan algoritma berikut.

1. Hasilkan bilangan prima besar (misalnya p dan q) dengan menggunakan algoritma pengujian bilangan prima (misalnya algoritma Miller-Rabin).
2. Kalikan p dan q sehingga didapat hasil kali n untuk faktorisasi kunci publik dan kunci privat serta digunakan sebagai panjang kunci.
3. Hitung $\varphi(n) = (p\text{-}1)(q\text{-}1)$.
4. Pilih sebuah bilangan bulat e secara acak sebagai eksponen kunci publik sehingga memenuhi $1 < e < \varphi(n)$ dan faktor persekutuan terbesar (FPB) dari e dan $\varphi(n)$ adalah 1.
5. Hitung $d \equiv e^{-1} \pmod{\varphi(n)}$.
6. Tetapkan (e, n) sebagai kunci privat dan d sebagai kunci publik.

Setelah pasangan kunci penerima pesan dihasilkan, maka sembarang pengirim pesan dapat menggunakan kunci publik penerima untuk mengirim pesan teks sandi kepada penerima dengan rumus:

$$C = P^e \bmod n$$

dengan C = teks sandi (*cipher text*) dan P = teks asli (*plain text*).

Akhirnya, setelah pesan teks sandi sampai pada penerima, penerima dapat menggunakan kunci privatnya untuk mengembalikan pesan teks sandi menjadi pesan teks asli dengan rumus:

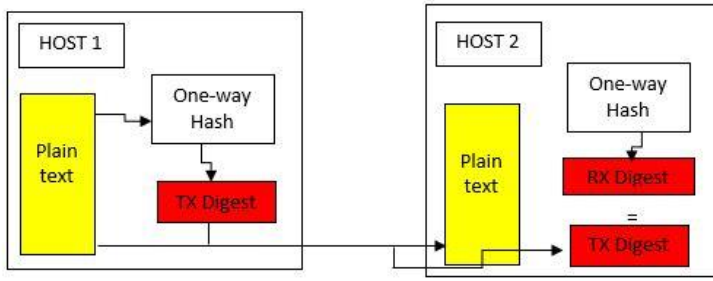
$$P = C^d \bmod n$$

Disarankan besar *p* dan *q* adalah $\geq$ 512 bit sehingga besar *n* menjadi minimal 1024 bit agar sistem kriptografi ini menjadi aman dan *n* susah difaktorkan. RSA banyak dimanfaatkan untuk proses autentikasi pengguna ke dalam sistem dan*digital signature*[3].

Akses ke sistem file telepon (perhatikan bahwa beberapa ponsel bekerja juga sebagai perangkat penyimpanan USB dan mereka tidak dapat diakses melalui Gammu).

1.4 Fungsi *Hash* SHA-256

Secure Hash Algoritm (SHA) dikembangkan oleh National Institute and Standard Technology (NIST) pertama kali pada tahun 1993. Generasi pertama SHA diberi nama **SHA-0**. Kemudian pada tahun 1995, generasi kedua SHA, **SHA-1**, muncul dan dipublikasikan oleh NIST dengan kode publikasi FIPS PUB 180-1. Generasi kedua SHA ini muncul dalam waktu 12 jam dari setelah dilaporkannya terdapat kelemahan dalam algoritma SHA-0. Generasi ketiga algortima SHA, **SHA-2**, dipublikasikan pada tahun 2001 dengan berbagai pilihan jumlah bit yang digunakan, yaitu: 224, 256, 384, dan 512. Baik SHA-1 dan SHA-2 pada dasarnya memiliki algortima yang serupa, hanya berbeda di jumlah karakter outputnya saja. SHA-1, SHA-256, dan SHA-516 memiliki jumlah karakter output masing-masing secara berurutan adalah 40, 32, dan 64 [11].Fungsi Hash merupakan sebuah algoritma yang mengubah text atau message menjadi sederetan karakter acak yang memiliki jumlah karakter yang sama. Hash juga termasuk salah satu bentuk teknik kriptografi dan dikategorikan sebagai kriptografi tanpa key (*unkeyed cryptosystem*). Selain itu hash memiliki nama lain yang juga dikenal luas yaitu "*one-way functio*n".



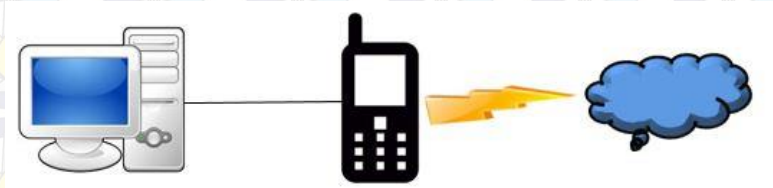
Gambar 1.3 Proses *one way hash*

Sering kali dijumpai hash di website-website yang menyediakan layanan untuk download file ataupun program secara resmi. Hash memang umumnya digunakan untuk mengecek integritas dari sebuah pesan atau file. File atau pesan yang sudah berubah akan memiliki nilai hash yang berbeda. Sebagai contoh, dengan sebuah algoritma hash, pesan'hello' akan memberikan nilai hash 12345 sedangkan pesan 'hallo' memiliki nilai hash83746. Dengan kata lain output hash dari kata 'hello' tidak akan sama dengan 'hallo'. Bahkan sekalipun dalam pesan tersebut terlihat hanya memiliki perbedaan sedikit saja, namun nilai hash yang dimiliki oleh kedua pesan tersebut sangat jauh berbeda.

Berbeda dengan teknik enkripsi dalam kriptografi, tujuan hash memang mengubah sebuah pesan yang dapat dibaca (readable text) menjadi pesan acak (unreadable text) sama seperti enkripsi, namun hal mendasar yang menjadi perbedaan dari hash adalah pesan yang telah acak tadi tidak dapat diubah kembali menjadi pesan yang seharusnya. Inilah mengapa hash disebut juga sebagai "*one-way function*".

1.5 GSM.Com.Lib

GSMComm adalah metode pengiriman SMS melalui pengembang perpustakaan atau *library* yang disediakan oleh Visual Studio, bisa menjadi perpusatkaan komunikasi yang terdapat didalam GSM. Fungsi dari GSMComm hanya untuk SMS.



Gambar 1.4 Ilustrasi GSM Comm

Gamabr 2.4 merupakan proses pengiriman menggunakan aplikasi komunikasi yang terkoneksi dengan GSM Modem, *library*membantu aplikasi tersebut untuk melakukan proses pengiriman *token* menuju nomor handphone yang dituju.

BAB III
DESAIN DAN PERANCANGAN

Bab ini berisi pembahasan 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.

1.1 Perancangan Alur Sistem secara Umum

Pada Tugas Akhir ini dibangun sistem aplikasi *e-voting* yang mengimplementasika integritas data suara pemilih. Secara garis besar aplikasi *e-voting* ini terdiri dari aplikasi halaman admin dan aplikasi halaman voter, aplikasi halaman admin ini terdiri jendela utama halaman admin dan enam *tab* untuk proses authentikasi pemilih selama proses berjalan dan perhitungan suara.

Gambar 3.1 menunjukan alur sistem secara umum. Secara garis besar alur sistem ini terbagi menjadi: (a) proses pemungutan suara dan (b) proses penghitungan suara. Tahapan tahapan yang terjadi selama proses pemungutan suara adalah sebagai berikut :

1. Pemilih/voter datang di Tempat Pemungutan Suara (TPS) sambil membawa Kartu Tanda Pengenal (KTP). Pemilih menuju *admin* untuk mengecek data pada KTP milik pemilih tersebut dengan bantuan perangkat komputer.
2. A. Jika data pemilih terdapat dalam basis data dan belum pernah memilih, maka diizinkan untuk masuk TPS.

   B. Jika data pemilih tidak terdapat dalam basis data atau kalaupun terdapat dalam basis data namun sudah memilih sebelumnya, maka pemiliih tidak diizinkan untuk masuk TPS.
3. Pemilih yang diizinkan masuk TPS, harus mengantre untuk menunggu gilirannya dipanggil panitia Kelompok Penyelenggara Pemungutan Suara (KPPS). Setelah

pemilihdipanggil KPPS, Pemilih datang dan akan dikirimkan token melalui SMS dari KPPS/Panitia.

4. Pemilih menuju bilik suara untuk memilih secara online dengan meng-*inputkan* token tersebut dihalaman voter.
5. Pemilih dapat memilih Partai dan Anggota Legislatif, jika : Token tersebut sesuai dengan *hash value* yang sudah ditentukan. Jika ada perubahan dalam sistem database maka muncul pesan, "Anda gagal masuk untuk memilih Partai dan Anggota DPRD Kota Surabaya" .
6. Pemilih meninggalkan TPS.

Tahapan-tahapan yang terjadi selama proses perhitunga suara adalah sebagai berikut :

1. Sebelum proses perhitungan suara dimulai,admin harus memastikan bahwa semua surat suara telah siap untuk diolah serta kunci publi dan privat telah dimuat dalam aplikasi. Kedua tersebut nantinya digunakan dalam proses penyimpanan data suara secara aman dan tanda tangan panitia.
2. Surat suara dihitung satu persatu secara online dengan cara memilih partai dan anggota DPRD Kota Surabaya di aplikasi yang sudah disiapkan.
3. Surat suara tersebut juga dihitung secara elektronik dengan cara mengolah data suara. Data suara disimpan dalam basis data dalam bentuk mentahan (plain) dan terenkripsi (ciphered). Data suara mentahan digunakan untuk menghitung suara secara elektronik, sedangkan data suara terenkripsi dihasilkan dengan memanfaatkan kunci publik dan privat yang telah dimuat sebelumnya ke dalam aplikasi.
4. Admin kemudian meminta panitia yang hadir di TPS untuk memeriksa data suara yang telah masuk ke dalam basis data. Caranya dengan menampilkan data suara dari basis data, kemudian panitia memberikan tanda tangan disertai dengan status persetujuannya. Tanda tangan dihasilkan dengan

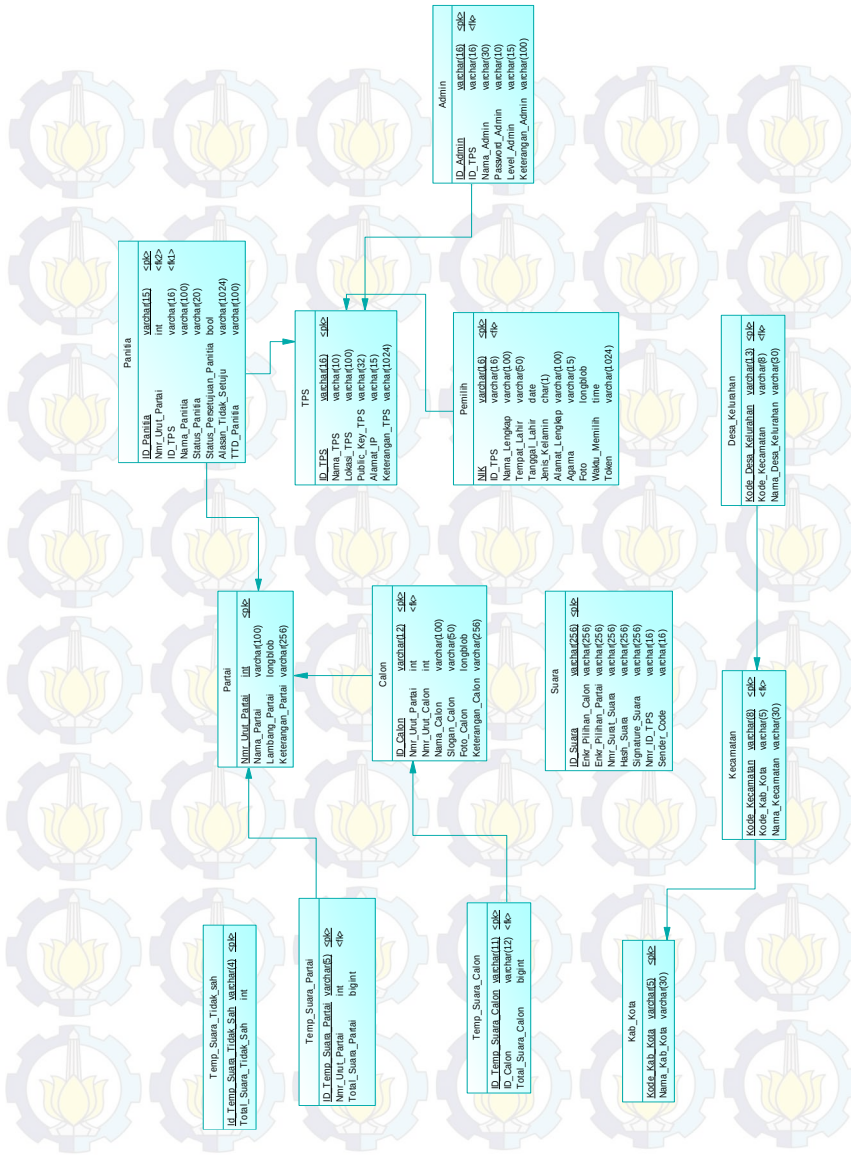
memanfaatkan kunci publik dan privat yang telah dimuat sebelumnya kedalam aplikasi.

5. Data suara yang telah diperiksa oleh panitia tersebut lalu dikirim ke *server* pusat untuk diprses datanya secara keseluruhan.

1.2 Perancangan Basis Data

1.2.1 Perancangan PDM

Berikut ini adalah rancangan dari sistem *E-Voting* yang membutuhkan basis data untuk menyimpan keseluruhan data. Sebelum pembuatan basis data dilaksanakan akan dilakukan pemodelan terlebih dahulu. Gambar 1.1 berikut adalah rancangan berupa *Physical Data Model* (PDM) yang akan digunakan pada sistem yang akan dibangun.



Gambar 1.1 Physical Data Model (PDM) sistem *e-voting*

1.2.2 Perancangan Tabel-tabel dalam Basis Data berserta Penjelasannya

Basis data sistem E-Voting dirancang sesuai dengan diagram PDM yang telah dibuat. Tabel-tabel yang terdapat dalam basis data tersebut adalah sebagai berikut. Untuk diperhatikan bahwa atribut bertanda *) merupakan kunci utama/*primary key* dan **) merupakan kunci tamu/*foreign key* yang merujuk pada atribut milik tabel kain.

- Tabel TPS
  Tabel ini berfungsi untuk menyimpan data tempat pemungutan suara (TPS).

Tabel 1.1Penjelasan atribut-atributtabel TPS

Nama Atribut/Field	Tipe Data	Keterangan
ID_TPS*)	varchar(16)	Nomor ID TPS
Nama_TPS	varchar(10)	TPS diikuti dua digit nomor urut
Lokasi_TPS	varchar(100)	Alamat TPS
Public_Key_TPS	varchar(32)	Public Key untuk keperluan koneksi ke *server* pusat
Alamat_IP	varchar(15)	Alamat IP TPS/*server* lokal
Keterangan_TPS	varchar(1024)	Info tambahan mengenai TPS

- Tabel Pemilih
  Tabel ini berfungsi untuk menyimpan data pemilih.

Tabel 1.2 Penjelasanatribut-atributtabel Pemilih

Nama Atribut/Field	Tipe Data	Keterangan
NIK*)	varchar(16)	Nomor ID pemilih
ID_TPS**)	varchar(16)	Nomor ID TPS
Nama_Lengkap	varchar(100)	Nama lengkap pemilih
Tempat_Lahir	varchar(50)	Tempat lahir pemilih
Tanggal_Lahir	date	Tanggal lahir pemilih
Jenis_Kelamin	char	Jenis kelamin pemilih
Alamat_Lengkap	varchar(100)	Alamat lengkap pemilih
Agama	varchar	Agama pemilih
Foto	blob	Foto diri pemilih
Waktu_Memilih	Time	Sebagai flag, sudah memilih atau belum
Token	varchar(1024)	Nomor token pemilih untuk masuk sistem
Hashrow	varchar(1024)	Hash masing masing row

- Tabel *Admin*
  Tabel ini berfungsi untuk menyimpan data *Admin* TPS.

Tabel 1.3Penjelasan atribut-atributtabel *Admin*

Nama Atribut/Field	Tipe Data	Keterangan
ID_*Admin**)	varchar(16)	Nomor ID *admin*
ID_TPS**)	varchar(16)	Nomor ID TPS
Nama_*Admin*	varchar(30)	Nama *admin*
Password_*Admin*	varchar(10)	Sandi masuk *admin*
Level_*Admin*	varchar(15)	Level *admin* (TPS, desa/kelurahan, kecamatan, atau kabupaten/kota

Lanjutan Tabel 3.3		
Nama Atribut/Field	**Tipe Data**	**Keterangan**
Keterangan_*Admin*	varchar(100)	Info tambahan mengenai *admin*

- Tabel Suara
  Tabel ini berfungsi untuk menyimpan data suara.

Tabel 1.4Penjelasan atribut-atributtabel suara

Nama Atribut/Field	Tipe Data	Keterangan
ID_Suara*)	varchar(256)	Nomor ID suara, merupakan *hash* dari gabungan antara *string* partai pilihan, calon pilihan, jam pindai surat suara, nomor surat suara, dan ID TPS.
Enkripsi_Pilihan_ Partai	varchar(256)	Partai pilihan, terenkripsi. Dienkripsi menggunakan RSA dan dikonversi dalam bentuk string Base64.
Enkripsi_Pilihan_ Calon	varchar(256)	Calon pilihan terenkripsi. Dienkripsi menggunakan RSA dan dikonversi dalam bentuk string Base64.
Hash_suara	varchar(256)	Hash dari data suara
Signature_Suara	varchar(256)	Penanda suara, merupakan *hash* dari gabungan antara *string* partai pilihan terenkripsi, calon pilihan terenkripsi, nomor surat suara, dan ID TPS, lalu dienkripsi menggunakan

		RSA dan dikonversi dalam bentuk string Base64
Nmr_ID_TPS	varchar(16)	Sama dengan ID TPS
Sender_Code	varchar(16)	Kode yang dipakai di web service saat akan mengecek data yang dikirim ke *server* jika sewaktu-waktu koneksi mati/terputus.

- Tabel Partai
  Tabel ini berfungsi untuk menyimpan data partai peserta *E-Voting*.

Tabel 1.5Penjelasan atribut-atribut tabel Partai

Nama Atribut/Field	Tipe Data	Keterangan
Nmr_Urut_Partai*)	Int	Nomor urut partai peserta Pemilu
Nama_Partai	varchar(100)	Nama dan singkatan partai
Lambang_Partai	Blob	Gambar lambang partai
Keterangan_Partai	varchar(256)	Info tambahan mengenai partai

- Tabel Calon
  Tabel ini berfungsi untuk menyimpan data calon partai peserta *E-Voting*.

Tabel 1.6Penjelasan atribut-atribut tabel calon

Nama Atribut/Field	Tipe Data	Keterangan
ID_Calon	varchar(12)	Nomor ID calon
Nmr_Urut_Partai**)	Int	Nomor urut partai peserta Pemilu
Nmr_Urut_Calon	Int	Nomor urut calon peserta Pemilu

Lanjutan Tabel 3.6		
Nama Atribut/Field	**Tipe Data**	**Keterangan**
Nama_Calon	varchar(100)	Nama lengkap calon
Slogan_Calon	varchar(50)	Slogan yang diusung calon
Foto_Calon	Blob	Foto diri calon
Keterangan_Calon	varchar(256)	Info tambahan mengenai calon

- Tabel Panitia
  Tabel ini berfungsi untuk menyimpan data panitia diTPS, saksi partai dan KPPS.

Tabel 1.7Penjelasan atribut-atributtabel Panitia

Nama Atribut/Field	Tipe Data	Keterangan
ID_Panitia*)	varchar(15)	Nomor ID panitia
Nmr_Urut_Partai**)	Int	Nomor urut partai peserta Pemilu
ID_TPS**)	varchar(16)	Nomor ID TPS
Nama_Panitia	varchar(100)	Nama lengkap panitia
Status_Panitia	varchar(20)	Status panitia di TPS (saksi atau KPPS)
Status_Persetujuan_Panitia	Bool	Status setuju atau tidak setuju
TTD_Panitia	varchar(100)	Tanda tangan panitia

- Tabel Temp_Suara_Partai
  Tabel ini berfungsi untuk menyimpan suara sah partai sementara.

Tabel 1.8Penjelasan atribut-atributtabel Temp_Suara_Partai

Nama Atribut/Field	Tipe Data	Keterangan
ID_Temp_Suara_Partai*)	varchar(5)	Nomor ID suara partai sementara
Nmr_Urut_Partai**)	Int	Nomor urut partai peserta Pemilu
Total_Suara_Partai	Int	Menyimpan rekapitulasi suara partai sementara di TPS

- Tabel Temp_Suara_Calon
  Tabel ini berfungsi untuk menyimpan suara sah calon sementara.

Tabel 1.9 Penjelasan atribut-atributtabel Temp_Suara_Calon

Nama Atribut/Field	Tipe Data	Keterangan
ID_Temp_Suara_Calon*)	varchar(5)	Nomor ID suara calon sementara
ID_Calon**)	varchar(12)	Nomor urut calon peserta Pemilu
Total_Suara_Calon	Int	Menyimpan rekapitulasi suara calon sementara di TPS

- Tabel Kab_Kota
  Tabel ini berfungsi untuk menyimpan data wilayah kabupaten/kota

Tabel 1.10Penjelasan atribut-atribut tabel Temp_Suara_Tidak_Sah

Nama Atribut/Field	Tipe Data	Keterangan
ID_Temp_Suara_Tidak_Sah*)	varchar(20)	Nomer ID suara tidak sah sementara
Total_Suara_Tidak_Sah	Int	Menyimpan rekapitulasi suara tidak sah sementara di TPS

1.3 Perancangan Data

Data yang digunakan dan diproses dalam aplikasi ini adalah data sms token untuk autentikasi halaman voter.

Proses pembuatan token ini dilakukan oleh *admin* yaitu dengan cara meng-generate *hashing* antara NIK Pemilih + Nama_Lengkap lalu di random dan akan mendapatkan kode unik, dan diambil hanya tujuh karakter dari kiri.Gambar 1.2 menunjukkan kode sumber pembuatan token pemilih.

```
string hashed = sha256(txtPemilih.Text +
txtNIK.Text);

string subhashed = (hashed.Substring(0, 7));
```

Gambar 1.2Kode sumber cara pembuatan token

1.4 Perancangan Antarmuka Aplikasi

Antarmuka aplikasi ini dirancang untuk mempermudah interaksi antara aplikasi dengan pengguna. Pengguna disini adalah seorang *admin* yang bertugas di TPS. Terdapat dua aplikasi, yaitu aplikasi halaman admin, dan aplikasi halaman voter.

1.4.1 Halaman Admin

Halaman admin merupakan halaman yang muncul pertama kali saat aplikasi dijalankan. Halaman admin digunakan untuk *login admin.* Gambar 1.3 menunjukkan rancangan halaman admin.

Gambar 1.3Rancangan halaman untuk login *admin*

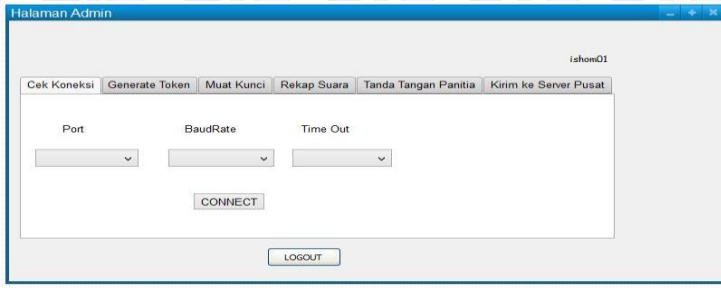
1.4.2 Halaman Admin Aplikasi

Halaman admin aplikasi ini merupakan halaman yang muncul setelah admin berhasil *login.* Terdapat lima *tab* yang tersedia dalam jendela ini, yaitu tab Cek Pemilih, Muat Kunci Privat dan Publik, Rekap Suara, Tanda tangan panitia, serta kirim ke *server* pusat.

1.4.2.1 *Tab* Cek Pemilh

Tab ini merupakan *tab* yang muncul pertama kali saat halaman utama aplikasi dijalankan. Terdapat dua kotak teks untuk mengisi data pemilih, yaitu kota teksi pertama untuk NIK pemilih dan kota teks kedua untuk nama lengkap pemilih. Selain itu, ada tombol periksa / kirim SMS token yang berfungsi sebagai periksa dan verifikasi pemilih dan mengirimkan token

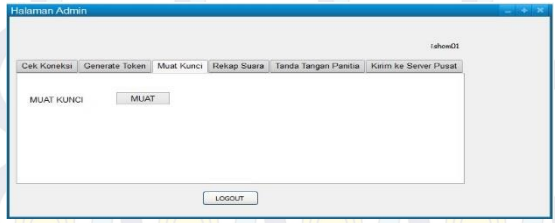
melalui SMS. Gambar 3.4 menunjukkan rancangan *tab* cek pemilih.



Gambar 1.4Perancangan halaman *admin* untuk cek koneksi

1.4.2.2 *Tab* Muat Kunci Private

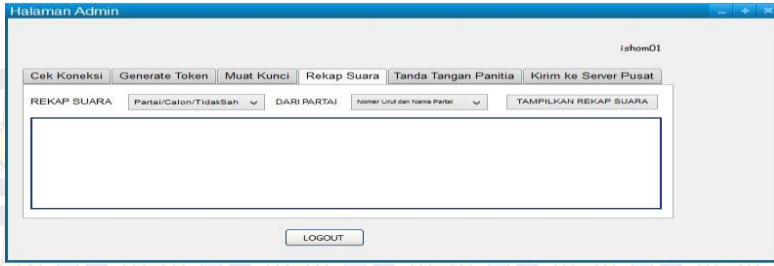
Tab ini digunakan untuk memuat kunci privat TPS dan publik KPU.Gambar 1.5 menunjukkan rancangan *tab* Muat Kunci Privat dan Publik.



Gambar 1.5Perancangan halaman *admin* untuk muat kunci privat

1.4.2.3 Tab Rekap Suara

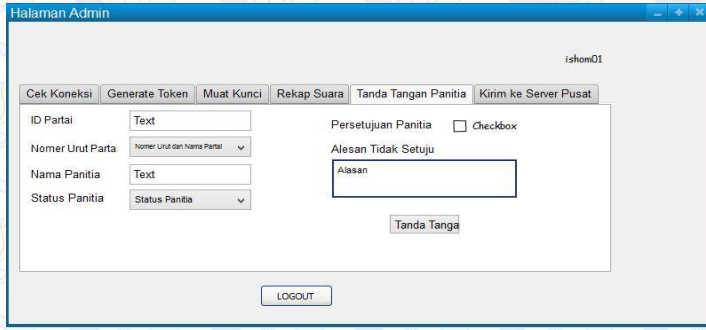
Tab ini digunkan untuk menghitung suara dengan memperhatikan surat suara yang sudah diolah. Gambar 1.6 menunjukkan rancangan *tab* rekap suara.



Gambar 1.6Perancangan Halaman Admin untuk Proses Rekap Suara

1.4.2.4 *Tab* Tanda Tangan Panitia

Tab ini digunakan untuk menandatangani data suara dan menyimpan tanda tangan tersebut yang bertugas di TPS ke dalam basis data *server* lokal dan mengirimkannya ke *server* pusat. Gambar 3.7 menunjukkan rancangan *tab* tanda tangan panitia.



Gambar 1.7Perancangan Halaman Admin untuk Proses Tanda Tangan Panitia

1.4.2.5 *Tab* Kirim ke Server Pusat

***Tab* ini digunakan untuk menampilkan data suara yang telah masuk dalam basis data *server* lokal dan**

mengirimkannya ke *server* pusat.

Gambar 1.8 menunjukkan *tab* Kirim ke Server Pusat.

Gambar 1.8Perancangan Halaman Admin untuk Proses Kirim ke Server Pusat

1.4.3 Halaman Voter

Halaman Voter adalah halaman dimana pemilih dapat melakukan proses pemilihan, setelah berhasil memasukan token, makan voter dapat melakukan proses, yaitu: (a) memasukan token sesuai dengan SMS yang diterima, (b) memilih partai, (c) memilih anggota legislatif dari partai yang bersangkutan. Setelah itu pemilih dapat meninggal Tempat Pemungutan Suara (TPS)

1.4.3.1 Form Memasukkan Token

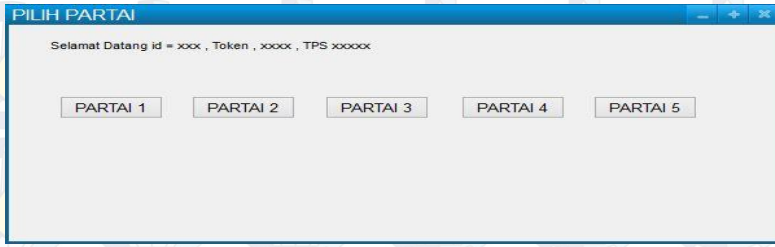
Form memasukkan token adalah halaman dimana pemilih memasukan nomer token ketika mendapatkan sms dari panitia berupa token yang bersifat rahasia, dan hanya pemilih saja yang

mengetahui nomer token tesebut. Gambar 1.9 menunjukkan Form Memasukkan Token.

Gambar 1.9Perancangan halaman voter untuk input token

1.4.3.2 Form Memilih Partai

Form memililih partai adalah halaman voter dimana pemilih bebas memilih partai yang dipilihnya. Gambar 1.10 menunjukkan form Memilih Partai.



Gambar 1.10Perancangan halaman voter untuk pilih partai

1.4.3.3 Form Memilih Anggota DPRD Kota Surabaya

Form memilih Anggota DPRD Kota Surabaya adalah halaman voter dimana voter / pemilih bebas memilih anggota legislatif DPRD Kota Surabaya. Gambar 1.11 menunjukkan form Memilih Anggota DPRD Kota Surabaya.

Gambar 1.11Perancangan halaman voter untuk pilih anggota legislatif

BAB IV
IMPLEMENTASI

Bab ini berisi pembahasan mengenai implementasi perangkat lunak berdasarkan perancangan yang telah dibuat. Tahap perancangan merupakan tahap dasar dari implementasi perangkat lunak.

1.1 Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk mengembangkan tugas akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti yang ditampilkan pada Tabel 4.1.

Tabel 1.1Lingkungan perancangan perangkat lunak

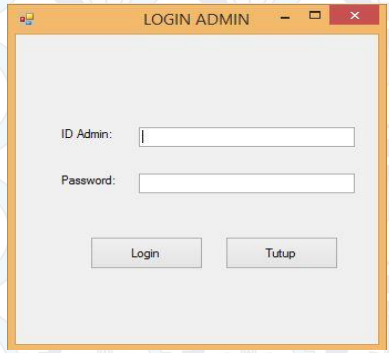
Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Core™ i7-4710HQ CPU @ 2.50GHz Memori: 4096 MB
Perangkat lunak	Sistem Operasi: 1. Microsoft Windows 8.1 Enterprise 64-bit Perangkat Pengembang: 1. Microsoft Visual Studio 2. Windows Dekstop with .NET Framework 4.5

	Perangkat Pembantu: 1. XAMPP PHPMyAdmin 2. Notepad++ 3. Microsoft Word 2013 4. Microsoft Visio 2013 Pustaka: 1. mySqldata.CF 2. RSAEncryptionLib 3. GSMComm.Lib

1.2 Implementasi Antarmuka Aplikasi

1.2.1 Implementasi Aplikasi Halaman Login Admin

Gambar 1.1 menunjukkan implementasi Halaman Login Admin.



Gambar 1.1 Implementasi halaman Login Admin

1.2.1.1 Implementasi Tombol Login

Tombol login berfungsi untuk mengizinkan *admin* untuk akses kedalam menu utama aplikasi halaman login *admin*. Setelah tombol diklik, maka aplikasi akan memeriksa koneksi ke *server* pusat untuk memanggil *web service* karena data *admin* terdapat dalam basis data *server* pusat. Apabila koneksi berhasil,

apliksi terebut akan mengecek data admin yang berusaha untuk *login* aplikasi ke basis data *server* pusat tersebut. Jika data *admin* ditemukan dan cocok, maka pesan *login* berhasil.

Jika data *admin* ditemukan dan cocok, maka pesan *login* berhasil akan ditampilkan. Namun, jika data *admin* tidak ditemukan atau tidak cocok, aplikasi akan menampilkan pesan kepada *admin* untuk mencoba *login* kembali. Jika koneksi ke *web service* gagal, maka aplikasi menampilkan pesan untuk memeriksa kembali koneksi tersebut.

```
privatevoid btnLogin_Click(object sender, EventArgs e)
{
try
   {
          ServicePointManager.ServerCertificateValidationCallba
ck
      = delegate(Object obj, X509Certificate
           certificate, X509Chain chain,
          SslPolicyErrors errors)
        {
return (true);
        };

        WebReference.Service1 ws = new
WebReference.Service1();
bool valid = false;
        valid = ws.Login(txtID_Adm.Text, txtPwd.Text);

if (valid == true)
        {
Form_Utama menu = newForm_Utama();
UserInfo.AdminId = txtID_Adm.Text;
MessageBox.Show("Anda berhasil login.", "Sukses!",
MessageBoxButtons.OK, MessageBoxIcon.Information);

this.Hide();
            menu.Show();
         }
else
        {
MessageBox.Show("Username Admin dan/atau password salah.
Coba lagi.", "Login Admin gagal", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
     }
```

```
catch
    {
MessageBox.Show("Ada masalah saat menghubungkan ke
        basis data. Mohon periksa kembali koneksi ke basis
            data.\n\n","Error koneksi",
MessageBoxButtons.OK,
        MessageBoxIcon.Error);
  }
}
```

Gambar 1.2 Kode sumber implementasi tombol Login

1.2.1.2 Implementasi Tombol Tutup

Tombol tutup berfungsi untuk menutup halaman admin, sekaligus menutup aplikasi. Setelah tombol ini diklik, maka aplikasi tertutup dan *admin* keluar dari aplikasi.

```
private void btnTutup_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

Gambar 1.3 Kode sumber implementasi tombol Tutup

1.2.2 Implementasi Halaman Utama Aplikasi

1.2.2.1 Implementasi Pemuatan Halaman Utama

Saat halaman utama aplikasi dimuat, maka beberapa elemen yang ada didalamnya juga ikut dimuat. Dalam fungsi yang ditunjukkan pada gambar dibawah ini, halam utama memuat label admin serta membuat koneksi ke basis data *server* lokal untuk memuat fitur *autocomplete* pada kota teks NIK pemilih. Selain itu koneksi tersebut juga digunakan untuk memuat isi ke dalam beberapa *combobox*. Implementasi label admin akan dijelaskan lebih lanjut dalam subbab 1.2.2.2, sedangkan omplementasi kotak teks NIK pemilih akan

dijelaskan dalam subbab 1.2.2.3.1. Selain itu, dalam fungsi tesebut juga diset beberapa tombol lain yang terdapat di tab-tab lainnya pada halaman utama tersebut menjadi tidak aktif. Implementasi masing-masing tombol tersebut akan dijelaskan pada subbab-subbab berikutnya.

```
privatevoid Form2_Load(object sender, EventArgs e)
        {
         lblAdmin.Text = "Admin: " + UserInfo.AdminId;
try
            {
             conn =
newMySqlConnection("server=localhost;database=evote;userna
me=root;password=;");
                conn1 =
newMySqlConnection("server=localhost;database=evote;userna
me=root;password=;");
                conn2 =
newMySqlConnection("server=localhost;database=evote;userna
me=root;password=;");
                conn.Open();
                conn1.Open();
                conn2.Open();

string query1 = "SELECT nik FROM pemilih WHERE nik
LIKE 'SBY%';";

cmd1 = newMySqlCommand(query1, conn1);
                dr1 = cmd1.ExecuteReader();

while (dr1.Read())
                {
                   n_nik.Add(dr1.GetString(0));
                }

dr1.Close();

string query2 = "SELECT CONCAT_WS('. ',
LPAD(tp.nmr_urut_partai, 2, '0'), p.nama_partai) AS
daftar_partai FROM partai p, temp_suara_partai tp " +
"WHERE tp.nmr_urut_partai = p.nmr_urut_partai AND
p.nmr_urut_partai != 0;";
cmd2 = newMySqlCommand(query2, conn2);
```

```
            dr2 = cmd2.ExecuteReader();
while (dr2.Read())
            {
                cboCalonPartai.Items.Add(dr2.GetString(0));
            }
dr2.Close();
MySqlDataAdapter adp = newMySqlDataAdapter();
DataSet ds = newDataSet();
string query3 = "SELECT nmr_urut_partai FROM partai;";
            cmd = newMySqlCommand(query3, conn);


adp.SelectCommand = cmd;
            adp.Fill(ds);
            adp.Dispose();
            cmd.Dispose();
            cboIdPartai.DataSource = ds.Tables[0];
            cboIdPartai.ValueMember = "nmr_urut_partai";
            cboIdPartai.DisplayMember =
cboIdPartai.ValueMember;
cboIdPartai.SelectedIndex = 0;
            cboStatusPan.SelectedIndex = 0;
conn.Close();
            conn1.Close();
            conn2.Close();
if (n_nik != null)
            {
lock (this)
                {
foreach (var item in n_nik)
                    {
                        acsc_nik.Add(item);
                    }
MessageBox.Show("Proses loading selesai.", "Done!",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
```

```
            }

        }
catch
        {
MessageBox.Show("Ada masalah saat menghubungkan ke basis
data. Mohon periksa kembali koneksi ke basis data.\n\n" +
"Aplikasi akan ditutup.", "Error koneksi",
MessageBoxButtons.OK, MessageBoxIcon.Error);
Application.Exit();
        }
    }
```

Gambar 1.4 Kode sumberimplementasi pemuatan halaman utama aplikasi

1.2.2.2 Implementasi Label Admin

Label Admin berfungsi untuk menampilkan data berupa ID *admin* yang sedang *login* di halaman utama aplikasi saat itu. Label ini berfungsi untuk menerima data masukan ID admin yang terdapat dalam kota teks txtID_Adm dalam halaman utama melalui kelas perantara **UserInfo.**

Kelas UserInfo.cs

```
static class UserInfo
{
    public static string AdminId;
}
```

Gambar 1.5 Kode sumber kelas UserInfo

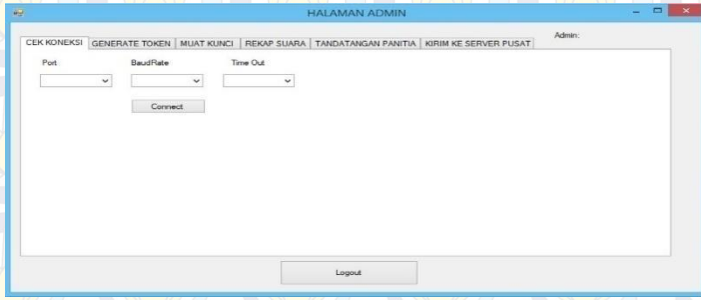
Halaman_Admin.cs

```
lblAdmin.Text = "Admin: " + UserInfo.AdminId;
```

Gambar 1.6 Kode sumber implementasi label Admin

1.2.2.3 Implementasi HalamanAdmin

Gambar 1.7 Implementasi *tab* 1 halaman utama aplikasimenunjukkan implementasi *tab* halaman utama aplikasi.



Gambar 1.7 Implementasi *tab* 1 halaman utama aplikasi

1.2.2.3.1 Implementasi Cek Koneksi

Halaman pertama atau tab pertama dalam halaman admin adalah cek koneksi, yang dimaksud dengan cek koneksi adalah, sebelum kita memulai pemilu, kita harus cek koneksi dari modem,ketika modem mempunyai pesan “Koneksi berhasil” maka silahkan melanjutkan proses yang selanjutnya. Berikut implementasi gambar 4.8 dari cek koneksi modem

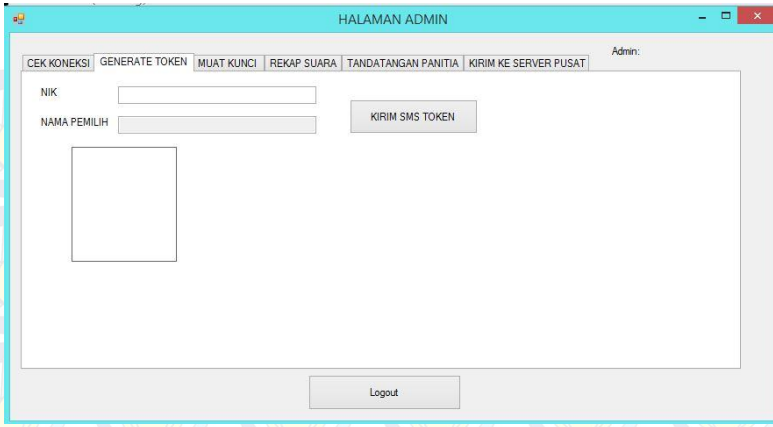
```
private void button3_Click_1(object sender, EventArgs e)
      {
string a = comboBox1.Text.Substring(3);

          port = Convert.ToInt32(a);
          baudRate = int.Parse(comboBox2.Text);
          timeout = int.Parse(comboBox3.Text);

GsmCommMain comm = newGsmCommMain(port, baudRate,
timeout);
          System.Console.Write(port);
```

```
try
        {
            comm.Open();
MessageBox.Show("Koneksi Berhasil", "info");
            comm.Close();
        }
catch
        {
MessageBox.Show("Koneksi gagal!", "Error"); comm.Close();
        }
    }
```

Gambar 1.8 Kode sumber implementasi tab cek koneksi

1.2.2.3.2 Impementasi Generate Token

Kotak teks NIK berguna untuk menerima masukan berupa NIK pemilih (bertipe data string) dan memiliki *autocomplete*. Implementasi *autocomplete* terdapat didalam fungsi halaman admin. Ketika kotak teks tersebut menerima masukan data, maka secara otomatis kota teks tersebut mengaktifkan fitur *autocomplete* berisi daftar NIK yang diambil dalam basis data *server* lokal sesuai dengan masukan yang diterima. Lalu ketika ada satu NIK yang dipilih dari daftar tersebut, maka secara otomatis aplikasi juga menampilkan nama pemilih dan fotonya yang sesuai dengan NIK yang dipilih tadi.



Gambar 1.9 Implementasi *tab* 2 halaman utama aplikasi

```
privatevoid txtNIK_TextChanged(object sender, EventArgs e)
      {
try
         {
if (!(string.IsNullOrEmpty(txtNIK.Text)))
            {
lock (this)
               {
                  txtNIK.AutoCompleteCustomSource = acsc_nik;
               }
            }
         }
catch (Exception ex)
         {
MessageBox.Show(ex.ToString());
         }
privatevoid txtNIK_KeyDown(object sender, KeyEventArgs e)
      {
if (e.KeyData == Keys.Enter)
```

```
        {
 selItem = txtNIK.Text;
            conn =
newMySqlConnection("server=localhost;database=evote;userna
me=root;password=;");
            conn.Open();

string query = "SELECT nama_lengkap FROM pemilih
WHERE nik = '" + selItem + "';";
            cmd = newMySqlCommand(query, conn);
            dr = cmd.ExecuteReader();
while (dr.Read())
            {
               txtPemilih.Text = dr.GetString(0);
            }
            dr.Close();
            conn.Close();
            conn.Open();
query = "SELECT foto FROM pemilih WHERE nik = '" +
selItem + "';";
            cmd = newMySqlCommand(query, conn);
            cmd.ExecuteNonQuery();
DataTable dt = newDataTable();
MySqlDataAdapter adp = newMySqlDataAdapter(query, conn);
            adp.Fill(dt);
byte[] bits = null;
try
            {
               bits = (byte[])dt.Rows[0][0];
MemoryStream ms = newMemoryStream(bits);
this.photoImage.Image = Bitmap.FromStream(ms);
            }
catch
            {
this.photoImage.Image = null;
```

```
MessageBox.Show("Foto pemilih belum/tidak ada.", "Foto
Kosong!", MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
conn.Close();
        }
```

Gambar 1.10Kode sumber implementasi kotak teks TxtNIK

1.2.2.3.3 Implementasi Kirim SMS Token

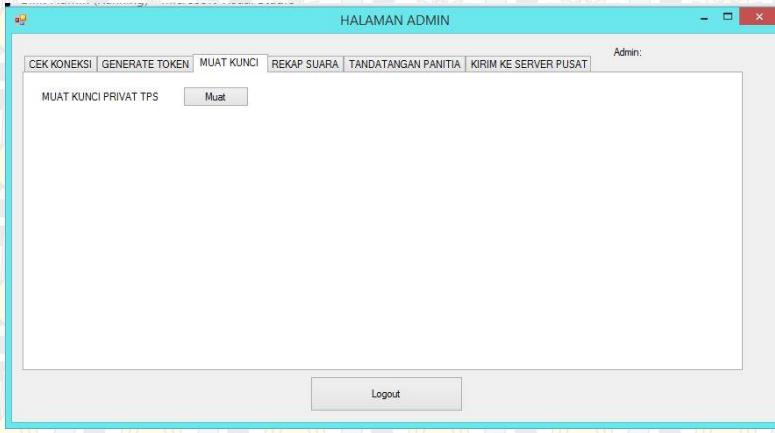
Setiap pemilih mendapatkan token melalui SMS dari panitia pemilu. Karena pemilu sendiri menggunakan asasnya, maka penulis mencarikan solusi dengan mempertimbangkan faktor keamanan.

```
try
                    {

string a = comboBox1.Text.Substring(3);

                        port = int.Parse(a);
                        baudRate = int.Parse(comboBox2.Text);
                        timeout = int.Parse(comboBox3.Text);
GsmCommMain comm = newGsmCommMain(port, baudRate,
timeout);
                        comm.Open();

SmsSubmitPdu pdu = newSmsSubmitPdu("Terimakasih berikut
nomer token anda "+subhashed, nomer_hp, "");
                        comm.SendMessage(pdu);
MessageBox.Show("Pesan Dikirim", "Info");
                        comm.Close();
                    }
catch { MessageBox.Show("Pesan Gagal Dikirm", "Error"); }
```

Gambar 1.11 Kode sumber impelementasi kirim SMS token

1.2.2.3.4 Implementasi Muat Kunci Privat

Terdapat kunci privat yang berfungsi untuk memuat berkas kunci privat yang berwujud XML. Kunci ini harus dimuat terlebih dahulu sebelum melakukan proses penghitungan suara dimulai untuk nantinya digunakan sebagai media bantuk enkripsi data suara dan tanda tangan panitia. Setelah tombol diklik, aplikasi membuka kotak dialog Open dengan penyaring berkas dan direktori awal yang sudah ditentukan. Setelah berkas ditemukan dan tombol open pada kotak dialog tersebut diklik, aplikasi membukanya dan menampilkan pesan bahwa berkas kunci privat telah dimuat.



Gambar 1.12 Implementasi tab 3 halaman utama aplikasi

```
privatevoid btnLoadPubKey_Click(object sender, EventArgs e)
        {
try
            {
```

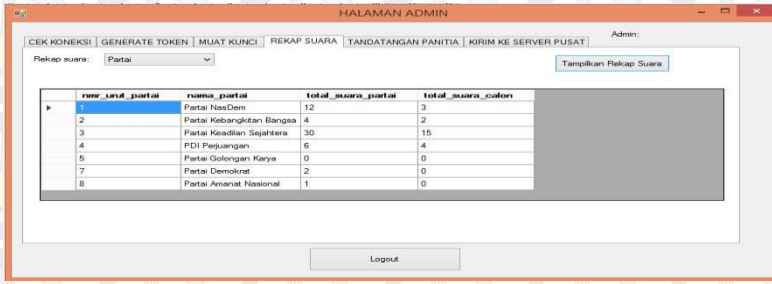
```
            openFileDialog1.InitialDirectory =
Application.StartupPath;
            openFileDialog1.Filter =
"(publickey.xml)|publickey.xml";
if (openFileDialog1.ShowDialog() != DialogResult.OK)
return;


myRsa2.LoadPublicFromXml(openFileDialog1.FileName);
            publicKeyLoaded = true;
MessageBox.Show("Kunci publik telah dimuat!", "Berhasil",
MessageBoxButtons.OK, MessageBoxIcon.Information);
this.button1.Enabled = false;
        }
catch (Exception ex)
        {
MessageBox.Show("Ada kesalahan saat mencoba memuat
kunci.\nPesan: " + ex.Message, "Oops..!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
```

Gambar 1.13 Kode sumber implementasi tombol Muat Kunci Privat TPS

1.2.2.3.5 Implementasi Rekap Suara

Gambar 1.14 Implementasi *combobox* Rekap Suara menunjukkan implementasi *tab* 4 halaman utama aplikasi, combobox jenis rekap berfungsi untuk menyimpan pilihan rekap suara : Partai, Calon, dan Tidak sah. Jika pilihan calon dipilih maka akan muncul combox pada calon partai yang berfungsi untuk menyaring daftar rekap suara calon berdasarkan partai

yang mendukungnya. Jika pilihan partai atau tidak sah dipilih, maka calon partai tersebut tidak muncul.



Gambar 1.14 Implementasi *combobox* Rekap Suara

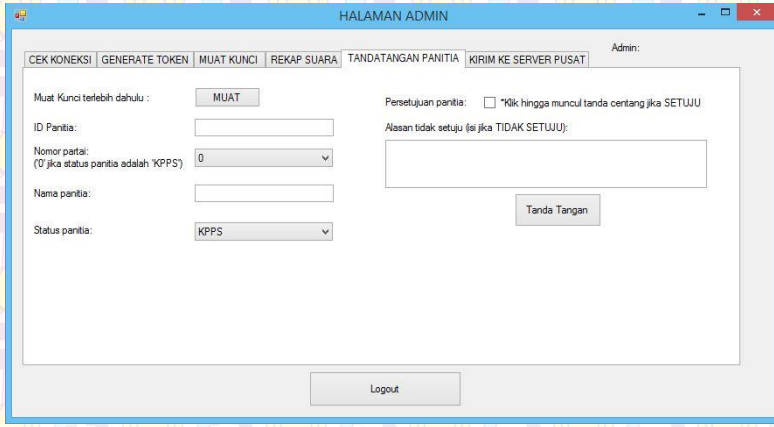
Gambar 1.15 berikut merupakan implementasi dari rekap suara.

```
privatevoid cboJenisRekap_SelectedIndexChanged(object
sender, EventArgs e)
      {
if (cboJenisRekap.SelectedIndex == 0 ||
cboJenisRekap.SelectedIndex == 2)
         {
            lblDrPartai.Hide();
            cboCalonPartai.Hide(); }
else
         {
            lblDrPartai.Show();
            cboCalonPartai.Show();
         }
      }
```

Gambar 1.15 Kode sumber implementasi *combobox* Rekap Suara

1.2.2.3.6 Implementasi Tanda Tangan Panitia

Gambar 1.16 menunjukkan implementasi tab 5 halaman utama aplikasi.

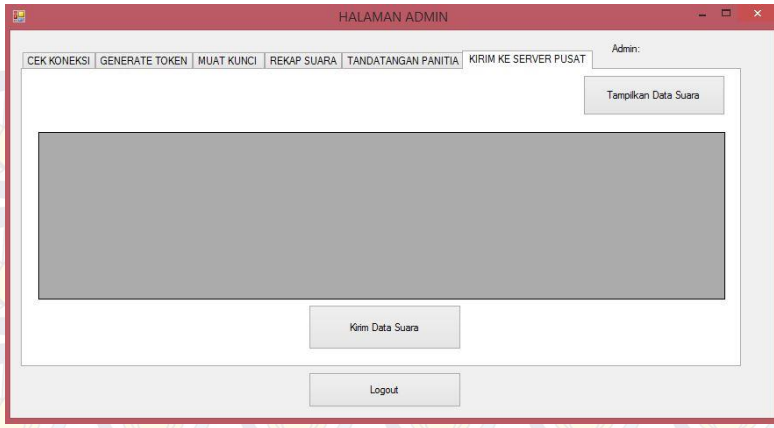


Gambar 1.16 Implementasi *tab* 5 halaman utama aplikasi

Dalam tab ini terdapat tombol tanda tangan yang berfungsi untuk memfasilitasi panitia yang bertugas di TPS agar bisa menandatangani data suara dan menyimpan tanda tanga tersebut ke dalam basis data server lokal. Setelah tombol ini di-klik aplikasi menghasilkan tada tangan panitia dan status persetujuannya menggunakan hash SHA-256 dan enkripsi RSA, lalu membuka koneksi ke basis data server lokal untuk menyimpan tanda tangan teebut ke dalamnya.

1.2.2.3.7 Implementasi Kirim ke Server Pusat

Gambar 1.17 menunjukkan implementasi tab 6 jendela utama aplikasi.



Gambar 1.17 Implementasi *tab* 6 Halaman Utama Aplikasi

Tombol tampilkan data suara berfungsi untuk menampilkan data suara dari basis data server lokal. Setelah tombol ini diklik, aplikasi membuat koneksi ke basis data tersebut, lalu mendapatkan dan menampilkan data uara dalam wujud tampilan datagrid. Setelah data suara tersebut tampil, aplikasi mengaktifkan tombol kirim data suara.

```
private void btnDispVotingData_Click(object sender, EventArgs
e)
        {
onn =
newMySqlConnection("server=localhost;database=evote;userna
me=root;password=;");
            conn.Open();
string query = "SELECT * FROM suara;";
cmd = newMySqlCommand(query, conn);
            dr = cmd.ExecuteReader();
while (dr.Read())
            {
Data temp = newData();
```

```
            temp.ID_Suara = dr.GetString(0);
            temp.Enkripsi_Pil_Calon = dr.GetString(1);
            temp.Enkripsi_Pil_Partai = dr.GetString(2);
            temp.No_Surat = dr.GetString(3);


temp.Hash_Suara = dr.GetString(4);
            temp.Signature = dr.GetString(5);
            temp.No_ID_TPS = dr.GetString(6);
            temp.Sender_Code = dr.GetString(7);
            listDataSuara.Add(temp);
        }

var bindingList = newBindingList<Data>(listDataSuara);
var source = newBindingSource(bindingList, null);
        dgvVotingData.DataSource = source;

btnSendVotingData.Enabled = true;
    }
temp.Hash_Suara = dr.GetString(4);
            temp.Signature = dr.GetString(5);
            temp.No_ID_TPS = dr.GetString(6);
            temp.Sender_Code = dr.GetString(7);
            listDataSuara.Add(temp);
        }

var bindingList = newBindingList<Data>(listDataSuara);
var source = newBindingSource(bindingList, null);
        dgvVotingData.DataSource = source;

btnSendVotingData.Enabled = true;
    }
```

Gambar 1.18 Kode Sumber Implementasi Tombol Tampilkan Data Suara

Tombol kirim data suara berfungsi untuk mengirim data suara yang tampil di dalam datagrid view. Setelah tombol ini di klik, aplikasi membuat koneksi ke web service milik service pusat. Selama proses pengiriman, aplikasi akan menampilkan urutan pesan pengiriman data suara yang berhasil maupun yang gagal.

```
privatevoid btnSendVotingData_Click(object sender, EventArgs
e)
        {
//Kirim data suara dr DB server lokal ke server pusat dlm
keadaan terenkripsi
ServicePointManager.ServerCertificateValidationCallback
            = delegate(Object obj, X509Certificate certificate,
X509Chain
            chain, SslPolicyErrors errors)
            {
return (true);
            };

            WebReference.Service1 serv = new
WebReference.Service1();
SHA256 sha256 = SHA256.Create();
byte[] hashData =
sha256.ComputeHash(Encoding.Default.GetBytes(idTps));
String hashedIDTPS =
BitConverter.ToString(hashData).Replace("-", "");
String encryptedIDTPS = "";

try
            {
byte[] message = Encoding.UTF8.GetBytes(hashedIDTPS);
byte[] encMessage = null;
```

```
            encMessage =
Form2.myRsa1.PrivateEncryption(message);
            encryptedIDTPS =
Convert.ToBase64String(encMessage);
        }
catch (Exception ex)
        {
MessageBox.Show("Ada kesalahan saat mencoba mengenkripsi
data.\nPesan: " + ex.Message);
return;
        }

try
        {
MessageBox.Show("Sending getSession()");
String session = serv.GetSession(hashedIDTPS);
if (session.Equals("Session not found!"))
            {
MessageBox.Show("Session not Found!");
MessageBox.Show("Sending Hello()");
String ok = serv.Hello(hashedIDTPS);
if (ok.Equals("OK"))
                {
MessageBox.Show("Hello OK");
MessageBox.Show("Sending Auth()");
                    session = serv.Auth(hashedIDTPS,
encryptedIDTPS);
MessageBox.Show(session);
if (session.Equals("AUTHENTICATION ERROR"))
                    {
MessageBox.Show("Sending data failed, Authentication
failed"); return;

else
                    {
```

```
MessageBox.Show("Sending Send()");
foreach (Data suara in listDataSuara)
                    {
string servMsg = serv.Send(session, suara.ID_Suara,
suara.Enkripsi_Pil_Calon, suara.Enkripsi_Pil_Partai,
suara.No_Surat, suara.Hash_Suara, suara.Signature, "TPS");
MessageBox.Show(servMsg);
                    }

MessageBox.Show("Sending Close()");
                    serv.Close(hashedIDTPS, session);
                }
            }
else
            {
MessageBox.Show("Sending data failed, IP address or ID
invalid");
return;
            }
        }
else
        {
MessageBox.Show("Session didapatkan!");
MessageBox.Show("Session : " + session);
MessageBox.Show("Sending Send()");
foreach (Data data in listDataSuara)
            {
string servMsg = serv.Send(session, data.ID_Suara,
data.Enkripsi_Pil_Calon, data.Enkripsi_Pil_Partai,
data.No_Surat, data.Hash_Suara, data.Signature, "TPS");
MessageBox.Show(servMsg);
            }

MessageBox.Show("Sending Close()");
```

```
                serv.Close(hashedIDTPS, session);
            }
        }
catch (Exception ex)
        {
MessageBox.Show(ex.Message);
return;
        }
    }
}
```

Gambar 1.19 Kode Sumber Implementasi Data Suara

1.2.2.3.8 Implementasi Tombol Logout

Tombol logout berfungsi untuk mengizinkan admin keluar dari halaman utama aplikasi. Setelah tombol logout diklik maka aplikasi akan memunculkan pesan konfirmasi untuk logout. Jika admin menekan tombol yes, maka aplikasi akan menutup halama admin dan menampilkan kembali halaman admin.

```
privatevoid btnLogout_Click(object sender, EventArgs e)
    {

Form1 home = newForm1();
DialogResult res = MessageBox.Show("Anda akan logout. Anda
yakin?", "Logout", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning);

if (res == DialogResult.Yes)
        {
this.Close();
            home.Show();
        }
```

```
    }
```

Gambar 1.20 Kode Sumber Implementasi Tombol Logout

BAB V
UJI COBA DAN EVALUASI

Bab ini berisi mengenai hasil uji coba dan evaluasi terhadap perangkat lunak dari implementasi aplikasi sistem *e-voting*.

1.1 Lingkungan Uji Coba

Lingkungan uji coba yang digunakan dalam pembuatan Tugas Akhir ini meliputi perangkat lunak dan perangkat keras yang digunakan untuk melakukan uji coba perangkat lunak dari implementasi kontrol integritas data pemilih suara dalam sistem *e-voting*. Lingkungan uji coba sistem yang spesifikasinya dijelaskan sebagai berikut.

1. Perangkat Keras
    a. Processor: Intel® Core™ i7-4710HQ CPU @ 2.50GHz
    b. Memory (RAM): 4.00 GB
    c. Tipe Sistem: 64-bit
2. Perangkat Lunak
    a. Sistem Operasi: Windows 8.1 Enterprise 64 bit.
    b. Perangkat Pengembang: Microsoft Visual Studio 2012
    c. Pustaka : MySqlData.CF dan RSAEncryption.lib
    d. Perangkat Tambahan: XAMPP, PHPmyAdmin dan Sistem manajemen basis Data MySQL.

1.2 Skenario Uji Coba

Pada bagian ini dijelaskan mengenai skenario uji coba yang telah dilakukan. Ada dua jenis uji coba yang dilakukan, yaitu:

1. Perangkat Keras

    Jenis uji coba ini berfungsi untuk menguji fungsionalitas dari aplikasi sistem yang dibuat. Uji coba yang telah dilakukan, yaitu:

a. Login Admin disaat terhubung dengan *web serviceserver* pusat maupun disaat tidak terhubung atau koneksi terputus.
b. Mengecek data pemilih.
c. Memuat kunci Publik dan Privat
d. Menyimpan data suara ke basis data *server* lokal.
e. Tanda tangan panitia.
f. Mengirim data suara ke *server* pusat.

2. Uji coba keamanan

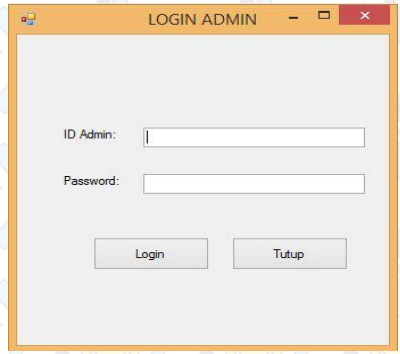
Jenis uji coba ini berfungsi untuk menguji keamanan dari aplikasi sistem yang dibuat. Uji coba yang telah dilakukan adalah:

a. Pemalsuan token dan nomer handphone
b. Pengiriman data suara ganda

1.2.1 Uji Coba Fungsionalitas

1.2.1.1 Skenario Pengujian 1: Login Admin

Dalam skenario ini, dilakukan uji coba login admin dalam keadaan terhubung atau tidak terhubung dengan *server* pusat. Login *admin* memerlukan *web service* yang dimiliki *server* pusat.



Gambar 1.1 Halaman *admin* yang diisi data login *admin*

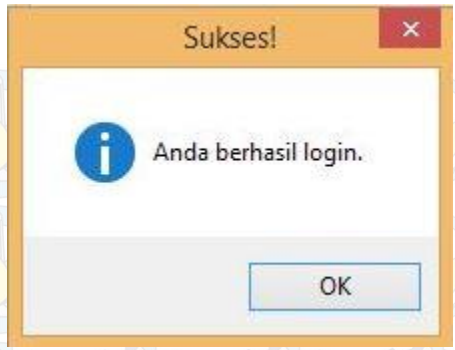
Apabila koneksi ke *server* pusat terganggu, misalnya terputus atau mati total, maka aplikasi menampilkan pesan seperti pada Gambar 1.2. Aplikasi akan meminta *admin* untuk mengecek kembali koneksi tersbut.

Gambar 1.2 Pesan gagal koneksi ke *server* pusat

Jika berhasil terhubung ke *server pusat*, *server* pusat merespon masukan data *login* dari aplikasi. Apabila data login tidak sesuai atau ada isian yang kosong, aplikasi menampilkan pesan seperti pada Gambar 5.3.

Gambar 1.3 Pesan bahwadata login salah

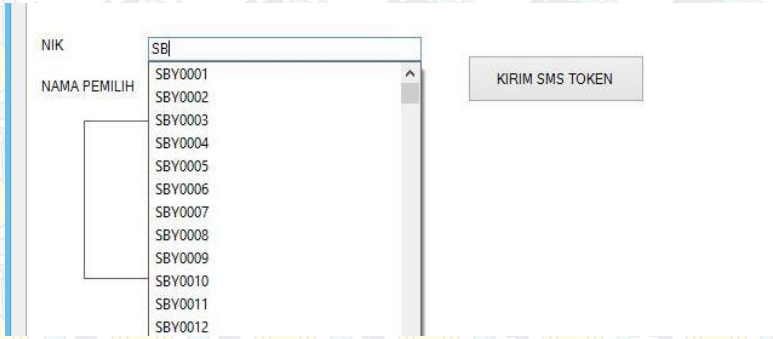
Sebaliknya, jika data *login* ditemukan, maka muncul pesan seperti padaGambar 1.4. Aplikasi akan menutup halaman login *admin*.



Gambar 1.4 Pesan bahwa *login*berhasil

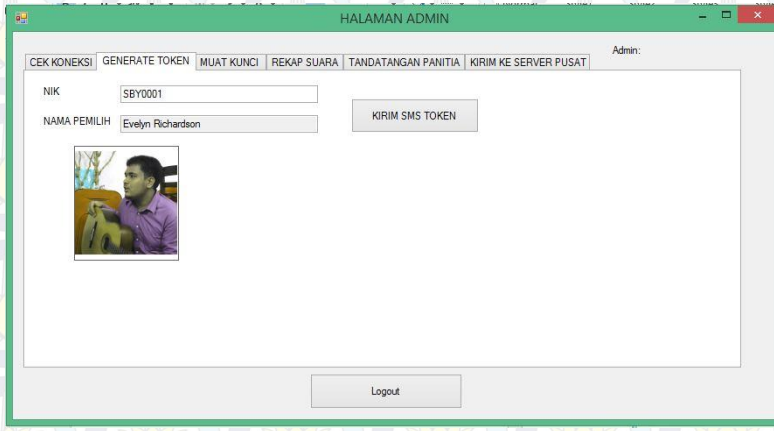
1.2.1.2 Skenario Pengujian 2: Mengecek Data Pemilih dan Kirim SMSToken

Dalam skenario ini, dilakukan uji coba cek pemilih untuk memverifikasi pemilih dan mengirimkan sms token pemilih sebelum masuk ke TPS dengan cara masukan data ke dalam kotak teks yang menggunakan fitur *autocomplete*. Gambar 1.5 menunjukkan bahwa tampilan *autocomplete* pada kotak teks NIK berubah-ubah menyesuaikan dengan masukan apapun yang diketik oleh *admin*.



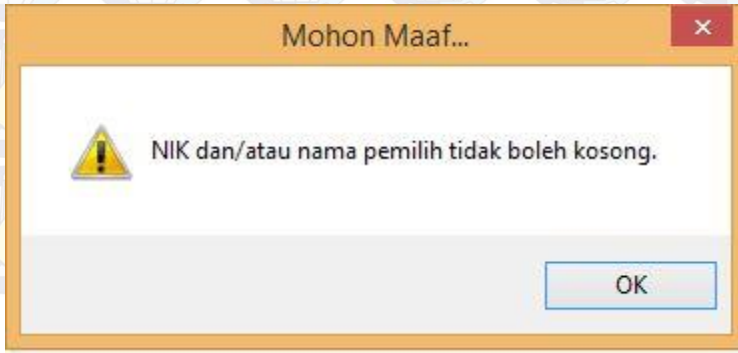
Gambar 1.5 Tampilan *autocomplete* saat memasukkan data NIK pemilih

Admin harus memilih salah satu NIK pemilih untuk menampilkan nama dan foto pemilih (jika ada). Gambar 1.6 menunjukkan tampilan *admin* setelah memilih salah satu NIK dari daftar yang tersedia pada *autocomplete* tersebut.



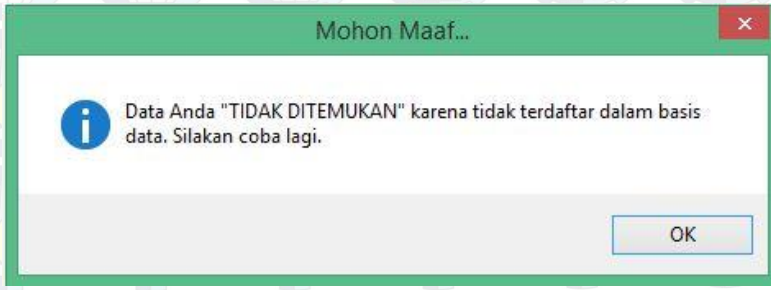
Gambar 1.6 Tampilan halaman Uuama dengan *tab* Generate Token setelah memilih NIK

Jika ada isian yang kosong, maka aplikasi akan memunculkan pesan seperti pada Gambar 1.7 berikut.



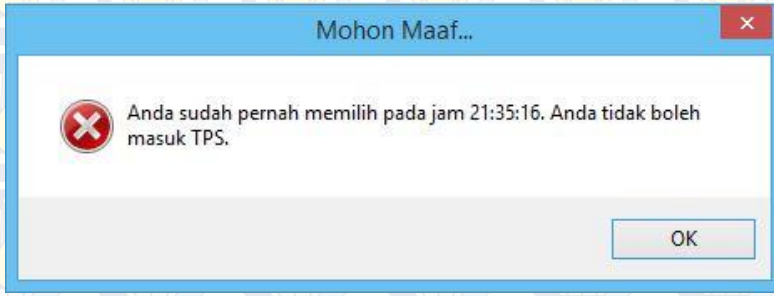
Gambar 1.7 Pesan bahwa terdapat isian yang masih kosong

Jika data pemilih tidak ditemukan di basis data, maka aplikasi akan memunculkan pesan seperti pada Gambar 1.8 berikut.



Gambar 1.8 Pesan bahwa data pemilih yang dimasukkan tidak ditemukan

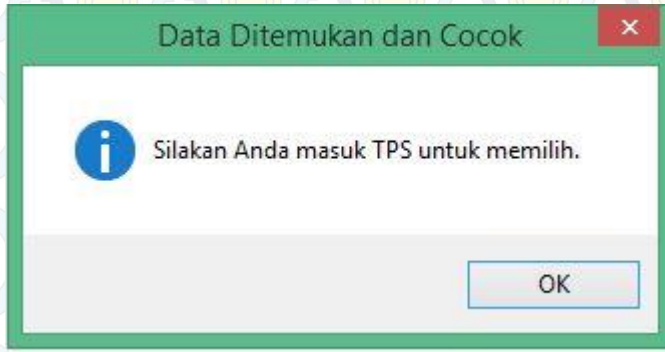
Jika data pemilih sudah pernah memilih sebelumnya, maka aplikasi akan memunculkan pesan seperti pada Gambar 1.9 berikut.



Gambar 1.9 Pesan bahwa pemilih sudah pernah memilih disertai jam memilihnya

Jika data pemilih ditemukan dan pemilih belum pernah memilih, maka aplikasi akan menampilkan pesan seperti pada

Gambar 1.10 berikut. Selain itu, aplikasi juga mengeset jam memilih tersebut kedalam basis data.



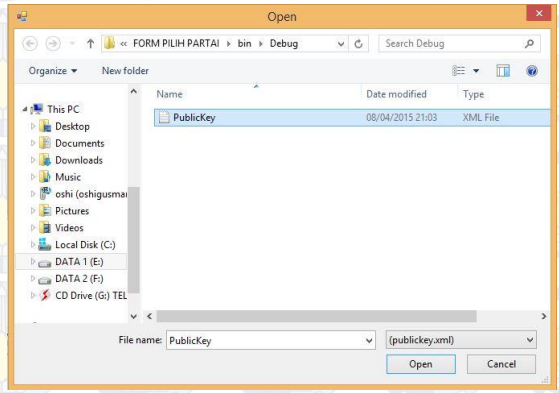
Gambar 1.10 Pesan bahwa Pemilih Dapat Memilih

Jika data pemilih ditemukan, namun belum memiliki foto maka aplikasi akan menampilkan pesan seperti Gambar 1.11 berikut.

Gambar 1.11 Pesan bahwa foto pemilih kosong

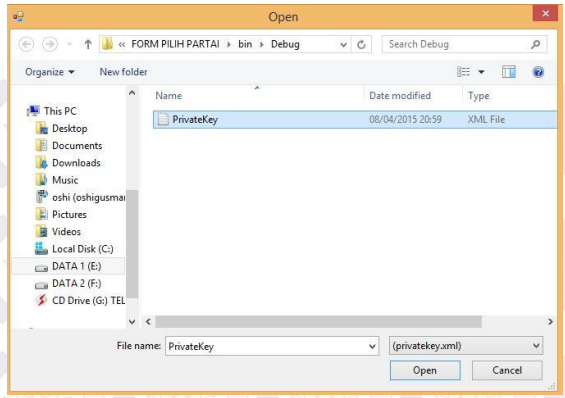
1.2.1.3 Skenario Pengujian 3: Memuat Kunci Publik dan Privat

Dalam skenario ini, dilakukan uji coba memuat kunci publik dan privat sebelum melakukan proses penyimpanan data suara dan tanda tangan panitia ke dalam basis data *server* lokal.



Admin tinggal mengeklik kedua tombol muat... untuk memuat masing-masing berkas kunci ke aplikasi melalui kota dialog Open seperti pada Gambar 1.12 berikut. Kotak dialog Open memilik penyaring sehingga mengurangi kesalahan dalam membuat berkas kunci.

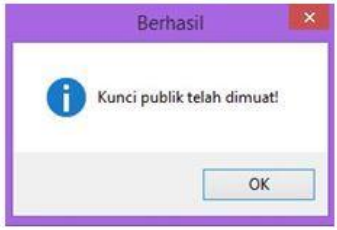
(a)

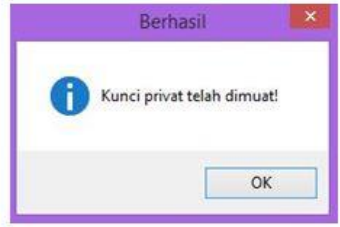


(b)

Gambar 1.12 Kotakdialog Open untuk memuat berkas (a) kunci privat dan (b) kunci publik

Selanjutnya aplikasi akan menampilkan pesan seperti pada Gambar 1.13 berikut.

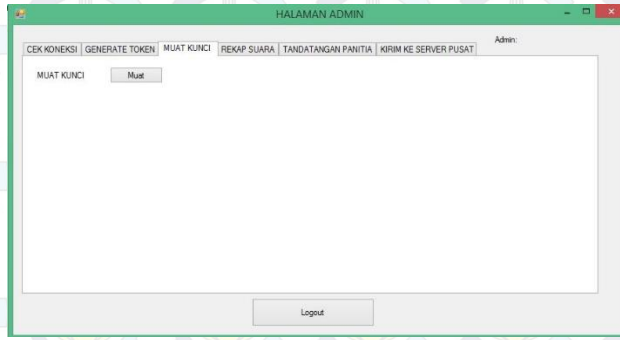




(a) **(b)**

Gambar 1.13 Pesan bahwa aplikasi berhasil memuat berkas (a) kunci privat dan (b) kunci publik

Gambar 1.14 berikut menunjukkan tampilan aplikasi setelah berkas kunci publik dan privat dimuat.



Gambar 1.14 Tampilan aplikasi setelah kunci privat dan publik dimuat

1.2.1.4 Skenario Pengujian 4: Menyimpan Data Suara ke Basis Data Server lokal

Dalam skenario ini, dilakukan uji coba menyimpan data suara. Data suara disimpan dalam bentuk mentahan (*plain*) maupun dalam keadaan terenkripsi (*cipher*).

Data suara mentaan masuk ke dalam basis data berdasarkan hasil suaranya: suara sah untuk partai,suara sah untuk calon, dan suara tidak sah. Setiap data suara dalam basis data sebanyak satu sauara. Gambar 5.18 berikut menunjukkan tampilan dalam basis data sebelum dan sesidah data suara sah untuk calon dengan nomer ID 3578010103 berhasil masuk.

ID_CALON	TOTAL_SUARA_CALON
3578010101	5
3578010102	0
3578010103	0
3578010104	0
3578010105	0

(a)

ID_CALON	TOTAL_SUARA_CALON
3578010101	5
3578010102	0
3578010103	1
3578010104	0
3578010105	0

(b)

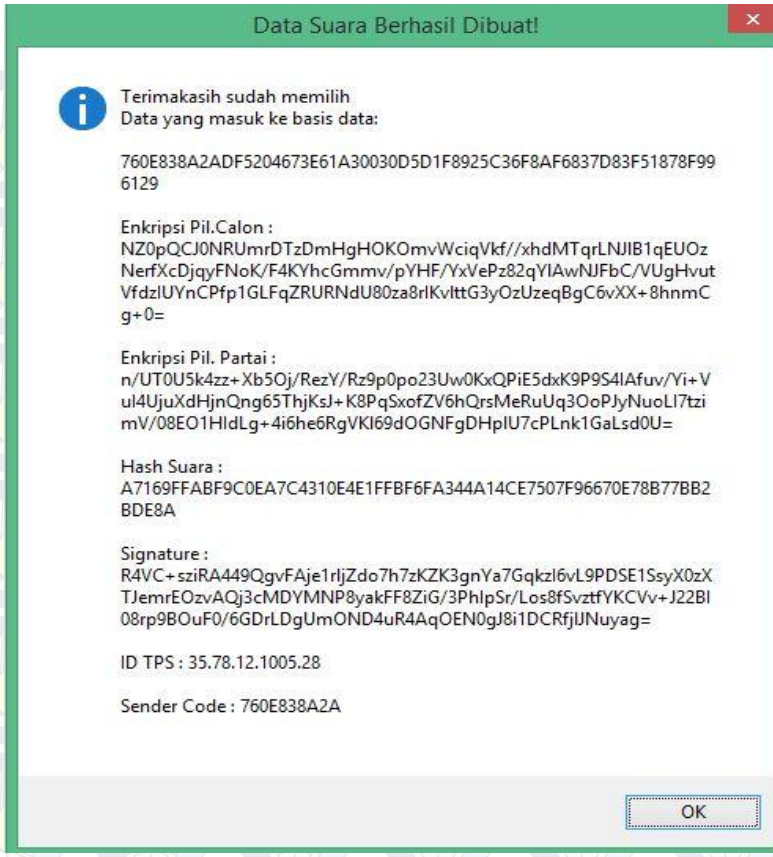
Gambar 1.15 Tampilan basis Data (a) Sebelum Data Suara Masuk dan (b) Setelah Data Suara Masuk

Data suara juga diamankan dengan cara menerapkan teknik enkripsi RSA dan *hash* SHA256. Data suara yang diproes berisi:

1. Nomor urut calon pilihan
2. Nomor urut partai pilihan
3. Nomor surat suara
4. *Hash* suara
5. ID TPS

Sebelum menyimpan data suara ke basis data, *admin* harus memastikan bahwa kunci publik dan privat sudah dimuat kedalam aplikasi.

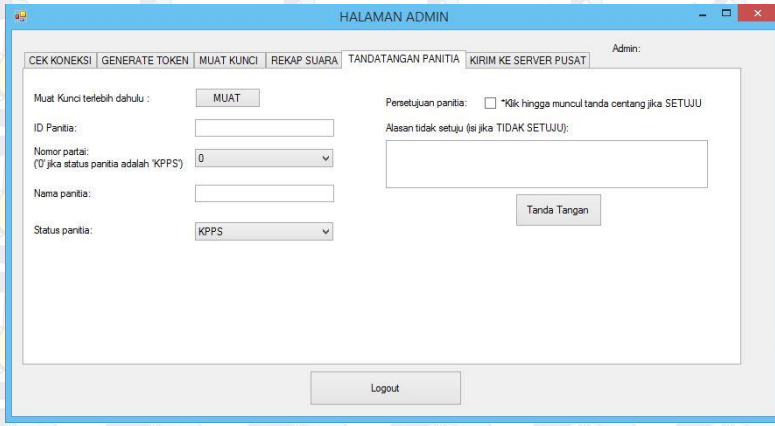
Setelah *admin* memuat kedua kunci tersebut, *admin* lalu menekan tombol simpan ke basis data. Aplikasi memproses data suara tesebut dan menampilkan pesan seperti pada Gambar 1.16. Setelah tombol OK pada kotak pesan, barulah data suara tersebut masuk ke dalam halaman pemilih.



Gambar 1.16 Pesan berisi data suara enkripsi yang ditambahkan ke basis data

1.2.1.5 Skenario Pengujian 5: Tanda Tangan Panitia

Dalam skenario ini, dilakukan uji coba tanda tangan panitia. Gambar 1.17 berikut menunjukkan bahwa panitia yang menyetujui dan juga yang tidak menyetujui data suara dapat melakukan tanda tangan pada aplikasi.

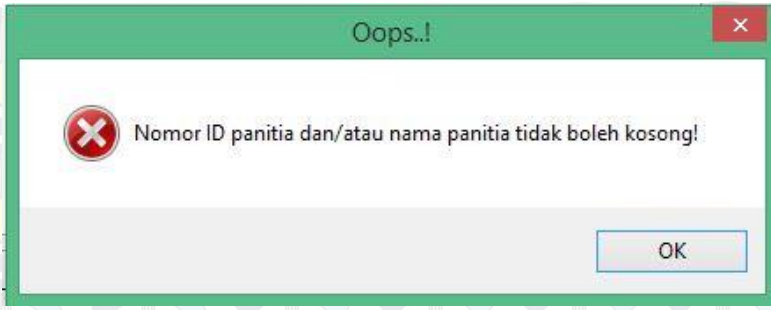


Gambar 1.17 Pengisian Tanda Tangan Panitia yang Menyetujui Data Suara

Perlu diingat bahwa dalam melakukan tanda tangan panitia, kunci privat TPS mutlak diperlukan. Apabila sebelumnya *admin* belum memuat kunci privat tersebut, maka setelah panitia mengisi datanya untuk tanda tangan dan *admin* mengeklik tombol Tanda tangan, aplikasi akan menampilkan pesan seperti pada Gambar 1.18 berikut.

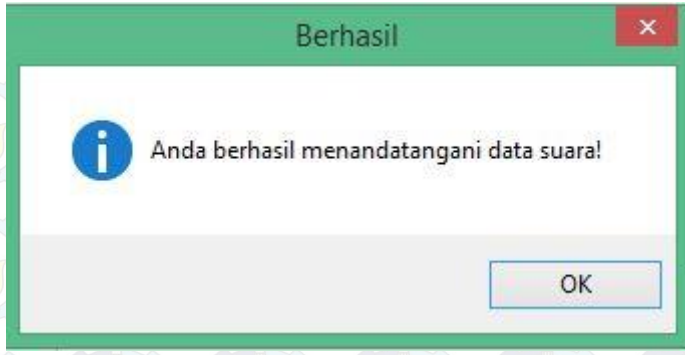
Gambar 1.18 Pesan untuk memuat kunci privat TPS sebelum panitia melakukan tanda tangan

Apabila data panitia berupa nomor ID atau namanya masih kosong, maka setelah *admin* meng-klik tombol tanda tangan, aplikasi akan menampilkan pesan seperti pada Gambar 1.19 berikut.



Gambar 1.19 Pesan untuk mengisi kotak teks yang kosong

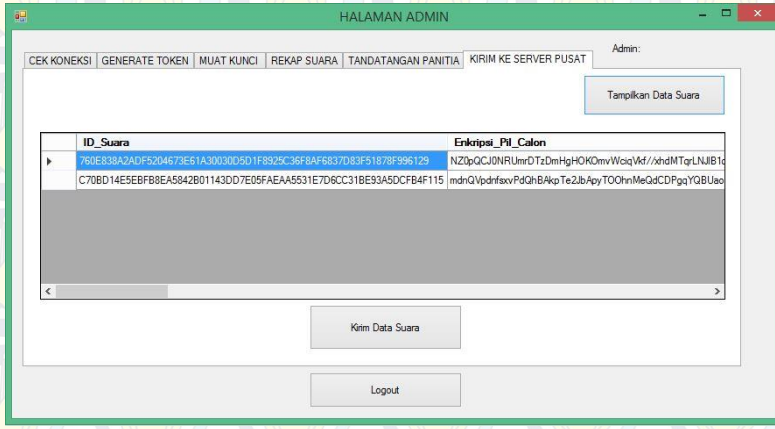
Apabila semua data diisi dan kunci privat TPS telah dimuat, maka setelah *admin* meng-klik tombol tanda tangan, aplikasi akan menampilkan pesan seperti pada Gambar 1.20 berikut



Gambar 1.20Pesan bahwa tanda tangan telah masuk

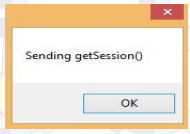
1.2.1.6 Skenario Pengujian 6: Kirim Data Suara ke *Server*Pusat

Dalam skenario ini, dilakukan uji coba mengirim data suara ke *server* pusat. Awalnya *admin* menampilkan data suara yang telah masuk kedalam basis data *server* lokal. Setelah tombol tampilkan data suara diklik, maka aplikasi akan menampilkan data suara seperti pada Gambar 1.21 berikut.

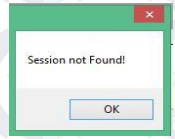


Gambar 1.21 Tampilan halaman *admin* dengan *Tab* Kirim ke *Server* Pusat setelah data suara dari basis data ditampilkan

Apabila data suara telah tampil, maka data suara tesebut siap dikirim ke server pusat dengan cara meng-klik tombol kirim Data Suara. Gambar 1.22 adalah urutan pesan-pesan yang muncul selama proses pengiriman data suara ke *server* pusat.



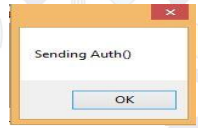
(I)



(II)

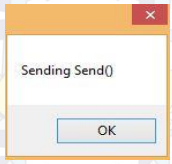
(III)

(IV)



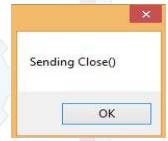
(V)

(VI)



(VII)

(VIII)



(IX)

Gambar 1.22Urutan pesan yang muncul selama proses pengiriman data suara ke *server* pusat

Gambar 1.23 adala tampilan basis data di *server* pusat setelah data suara berhasil dikirim dan diterima

Gambar 1.23 Data Suara di *Server* Pusat

Dari hasil uji coba tersebut, maka data tersebut dapat masuk dan akan diolah *server* pusat untuk mendapatkan data suara mentahan. Dengan metode enkripsi dan proses *hash* yang dilakukan, maka proses yang dilakukan pada uji coba tersebut aman dan proses tersebut tidak dapat dibuka kuncinya karena konsep kriptografi yang dipakai sudah sesuai dengan semestinya.

1.2.2 Uji Coba Keamanan

1.2.2.1 Skenario Pengujian 1: Keamanan database pada field nomer handphone

Pada skenario ini, dilakukan uji coba mengolah kemanan database pada *field* nomor handphone. Untuk menghindari admin yang ingin mengubah nomor handphone.

(a)

JENIS_KELAMIN	ALAMAT_LENGKAP	AGAMA	HP
F	9 Quincy Circle	Buddha	+6281259362430

(b)

Gambar 1.24 Tampilan pada basis data : (a) sebelum admin mengganti nomer handphone dan (b) setelah data suara diganti

Data nomor handphone setelah diganti oleh admin maka terdapat pesan error karena nomor handphone tidak tervalidasi, karena *hash* value berbeda dengan yang diinginkan oleh hashrow Gambar 1.25. Pesan berikutnya yang akan tampil sama seperti padaGambar 1.22 diatas.

Gambar 1.25 Pesan berisi nomer handphone tidak valid.

Pesan tersebut menunjukkan bahwa *signature* data suara yang dikirim tersebut berbeda dengan yang sebenarnya. Saat menghasilkan signature data suara tersebut, aplikasi terlebih daulu akan menghasilkan nilai *hash* dari gambungan data suara calon dan partai yang dienkripsi, ID TPS, dan nomor surat suara. Jika hasil nilai *hash*-nya berbeda, maka *signature* data suaranya juga pasti berbeda sehingga data suara yang diubah tersebut tidak bisa masuk ke *server* pusat.

1.2.2.2 Skenario Pengujian 2: Keamanan database pada field token

Pada skenario ini, dilakukan uji coba menemukan adanya perubahan atau pemalsuan terhadap basis data atau database pada tabel pemilih di *field* token.

AGAMA	HP	WAKTU_MEMILIH	TOKEN
Buddha	+6281259362431	*NULL*	b789ad3222oiasw22a78272812321sddsasd1212esadsd1

(a)

AGAMA	HP	WAKTU_MEMILIH	TOKEN
Buddha	+6281259362431	*NULL*	1234

(b)

Gambar 1.26 Perubahan terhadap data token: (a) data token sebelum diubah (b) data token setelah diubah

Cara paling mudah untuk menemukan perubahan tersebut adalah ketika admin merubah data token tersebut maka asumsi admin dapat memasukkan token tersebut di bilik pemilih dan admin dapat memilih partai dan anggota legislatif sesuai dengan yang admin inginkan.

Sistem aplikasi e-voting ini mempunyai keamanan data dari sisi database yaitu mengamankan agar supaya database tersebut tidak bisa diganti oleh admin. Yang kita amankan adalah menampilkan di *field* token hasil *hash* kedua dari token aslinya. Karena *hash* value nya berbeda dengan total *hashrow* maka akan muncul pesan menolak, seperti Gambar 1.27 berikut.

Gambar 1.27 Pesan yang ditampilkan saat memasukkan data token yang diubah atau dipalsukan

Pesan pada Gambar 5.27 menunjukkan bahwa *hash* value data token yang dikirim tersebut berbeda dengan yang sebenarnya. Saat menghasilkan token, aplikasi terlebih dahulu akan menghasilkan nilai *hash* dari gabungan data nama calon dan NIK yang dienkripsi. Jika hasil nilai *hash*-nya berbeda, maka data *hash* value token juga berbeda sehingga data suara yang diubah tersebut tidak bisa masuk ke *server* lokal.

1.3 Evaluasi

Dari sejumlah skenario uji coba fungsionalitas yang telah dilakukan, kontrol-kontrol aplikasi yang dibuat dapat berjalan dengan baik. Proses *e-voting* dapat berjalan dengan semestinya karena terdapat dua aplikasi yang saling berhubungan, yaitu halaman *admin* dan halaman *voter*.

Dari sejumlah skenario uji coba keamanan yang telah dilakukan, terbukti bahwa data suara yang diubah atau dipalsukan bisa mudah diketahui yaitu dengan cara ketika melakukan pengiriman menuju *server* pusat yang mana hasil dari proses pengirmannya adalah tidak masuknya data suara disebabkan karena perbedaan nilai *signature* data suara. Lalu, data suara yang sebelumnya sudah pernah dikirim ke *server* pusat tidak bisa masuk lagi karena adanya mekanisme pengecekan dari server pusat yang hasilnya bahwa nomer ID data suara yang telah dikirim sebelumnya dan yang akan diterima adalah sama. Dan skenario pengamanan serangan dari *admin* yang akan melakukan kecurangan, dengan mengubah basis data lokal token, sesuai dengan apa yang diinginkan. Namun sistem aplikasi akan menolak karena *hash-value* yang didapatkan tidak sesuai dengan token yang aslinya.

BAB VI
KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diambil selama pengerjaan tugas akhir serta saran-saran tentang pengembangan yang dapat dilakukan terhadap tugas akhir ini di masa yang akan datang.

1.1 Kesimpulan

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

1. Dengan melihat hasil uji coba fungsionalitas, semua kontrol aplikasi yang dibuat dapat berjalan dengan baik.
2. Rancangan keamanan data didalam database dapat berjalan dengan baik dari sisi admin melakukan kecurangan mengganti kolom token dan nomer handphone.
3. Data suara yang diubah atau dipalsukan tidak bisa dikirim ke *server* pusat karena nilai *hash* atau *signature* yang dihasilkan antara data yang asli dan data yang diubah adalah berbeda.
4. Data suara yang sebelumnya telah dikirim ke *server* pusat tidak bisa dikirim lagai karena nomor id data suara yang dikirim sebelumnya dengan yang dikirim lagi adalah sama.

1.2 Saran

Beberapa saran terkait tugas akhir ini yang diharapkan bisa membuat tugas akhir ini menjadi lebih baik antara lain:

1. Perlu adanya peralatan peralatan khusus dalam pemilihan umum di Indonesia, seperti laptop dan Wifi. Agar data tersebut dapat dikirim menuju *server* lokal maupun pusat.
2. Untukproses autentikasi aplikasi*E-Voting*menggunakan nama dan nomorKTP, kedepannya untuk proses otentikasi untuk pemilihan umum yang berada di Indonesia dengan perkembangan teknologi yang semakin maju maka penulis memberi saran kepada KPU (Komisi Pemilihan Umum) agar proses autentikasi menggunakan pemindaian barcode *e-KTP* atau dengan menggunakan fingerprint.

DAFTAR PUSTAKA

[1] A. Rokhman, "Prospek Penerapan E-Voting di Indonesia," Universitas Jendral Soedirman, 2011. [Online]. Available: http://map.unsoed.ac.id/2011/11/29/prospek-penerapan-e-voting-diindonesia.

[2] "BPPT Sukses Uji Coba Evoting Berbasis E-KTP di Jembrana, Bali," Sekretariat Kabinet Republik Indonesia, 18 Desember 2013. [Online].

[3] C. Utama, *CodeIgniter Framework. Bandung: Universitas Pasundan.*, 2011.

[4] I. Sommerville, *Software Engineering, 9th edition, AddisonWesley*, 2011.

[5] "PHP adalah - Hypertext Preprocessor," 10 Desember 2013. [Online]. Available: http://agiptek.com/index.php/php/101-php.html. [Diakses 23 12 2014].

[6] P. Mansyurin, "Debian Web *Server* with OpenSSL (HTTPS)," 9 Desember 2013. [Online]. Available: http://lebaksono.wordpress.com/2010/12/20/debian-web-*server*-withopenssl-https. [Diakses 23 12 1014].

[7] M. Mogollon, "Cryptography and Security Sevices: Mechanisms and Applications," 2007.

[8] U. P. Nasional, 8 Desember 2013. [Online]. Available: http://www.library.upnvj.ac.id/pdf/2s1teknikinformasi/205511014/bab2.pdf. [Diakses 23 12 2014].

[9] H. Saputro, “Pembelajaran Praktek Basis Data (MySQL),” 2012.

[10] H. d. Pan, “Ensuring Voters and Candidates’ Confidentiality in E-voting Systems,” pp. 1-6, 2011.

BIODATA PENULIS

Ishom Muhammad Drehem, penulis dari buku Tugas Akhir ini lahir di kota Sumenep pada tanggal 29 Agustus 1992. Penulis adalah anak kedua dari empat bersaudara. Penulis telah menempuh pendidikan di SD Al Irsyad Surabaya (1999-2005), SMP Al Hikmah Surabaya (2005-2008), SMA Al Hikmah Surabaya (2008-2011) dan terakhir di Teknik Informatika ITS Surabaya (2011-2015). Selama masa perkuliahan, penulis pernah menjadi asisten pada mata kuliah Organisasi Komputer, Riset Operasional dan Sistem Operasi . Penulis juga aktif sebagai anggota organisasi Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) ITS, Sekretaris Jendral BEM ITS, Ketua Mahakamah Mahasiswa ITS, Forum Indonesia Muda 15, Gerakan Melukis Harapan dan Young Leadership For Indonesia. Penulis memilih bidang minat Komputasi Berbasis Jaringan (KBJ) dan tertarik pada topik jaringan dan Manajemen Basis Data. Penulis dapat dihubungi melalui surel : ishomdrehem@hotmail.com.