DESAIN DAN IMPLEMENTASI SISTEM KEAMANAN *E-VOTING* DENGAN JAMINAN *CONFIDENTIALITY DATA*

SKRIPSI

LABORATORIUM REKAYASA PERANGKAT LUNAK

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh :

TIKA RAHMADIAN

NIM. 0910680034

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2014

Kata Pengantar

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena berkat karunia-Nya penulis dapat menyelesaikan proposal skripsi ini dengan lancar. Penyusunan skripsi ini adalah dengan maksud untuk memenuhi salah satu persyaratan dalam mengikuti ujian akhir Sarjana Program Studi Teknik Informatika pada Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya, dengan judul "Desain dan Implementasi Sistem Keamanan *E-Voting* dengan Jaminan *Confidentiality Data*".

Banyak hambatan yang telah penulis hadapi dalam penyelesaian skripsi ini, tetapi berkat bantuan dan dukungan dari berbagai pihak hambatan-hambatan tersebut bisa dilewati. Melalui kesempatan ini, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan dan dukungan selama penulisan skripsi, diantaranya:

1. Bapak Aswin Suharsono, S.T., M.T., selaku dosen pembimbing I yang telah memberikan ide dan saran yang berguna dalam penyelesaian proposal skripsi ini.
2. Bapak Ahmad Afif Supianto, S.Si., M.Kom selaku dosen pembimbing II yang telah memberikan pengarahan dalam pembuatan proposal skripsi ini.
3. Ibu Ari Kusyanti, S.T., M.Sc., yang telah memberikan inspirasi dalam penyelesaian proposal skripsi ini.
4. Bapak Drs. Marji, M.T, selaku Ketua Program Studi Teknik Informatika Universitas Brawijaya Malang.
5. Ir. Sutrisno, MT., selaku dosen pembimbing akademik yang telah memberikan pengarahan selama penulis menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
6. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada penulis selama menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
7. Ayahanda Machmudi dan Ibunda Rusiawati, kedua orangtua, atas segala kasih sayang dan dukungan moral, serta yang senantiasa tiada hentinya selalu

memberikan do'a dalam penyelesaian skripsi ini. Terima kasih untuk Boma Randiaz dan Nadia Feranika, kedua adik tersayang, yang selalu memberikan dorongan semangat untuk saya.

8. Bunga Mayangsari, Dian Novita W, Sangkan Phinandita G, Krisnha Bramantya, sahabat tersayang semenjak SD dan SMP yang selalu mendengarkan cerita dan keluhan saya, terima kasih atas persahabatan, *support* dan pengertian yang selalu diberikan.
9. Ika Kusumaning P., D. Rangga Maulana, M. N. Wibisono, Ardy Purnama P., sahabat terbaik, terima kasih sudah selalu mendengarkan cerita dan keluhan serta memberikan semangat kepada saya. Terima kasih juga untuk Aulia Meitika K. yang telah membantu dan memberikan saran serta semangat. Dyah Ayu M.G.W, *partner* dalam penyelesaian *e-voting*, terima kasih atas saran, semangat dan bantuannya.
10. Teman-teman rumpik di lab *game* (Anisa Aini, Austin Buya, Arianty Anggraini, Silvihanni Vionita, Fuad, Bagus, Muri, Tian dll), terima kasih atas saran, bantuan dan semangat yang selalu diberikan.
11. Seluruh teman-teman TIF angkatan 2009 dan khususnya kelas C, terima kasih telah menjadi bagian dari perjalanan kehidupan saya di kampus ini, sukses untuk kita semua !
12. Seluruh pihak yang telah membantu yang tidak dapat saya sebutkan satu per satu.

Semoga Allah SWT dengan rahmat dan karunia-Nya membalas kebaikan dan ketulusan semua pihak yang telah membantu menyelesaikan skripsi ini. Penulis menyadari bahwa proposal skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun penulis ucapkan terima kasih. Penulis mengharapkan semoga proposal skripsi ini dapat berguna bagi yang membutuhkannya.

Malang, April 2014

Penulis

ABSTRAK

Tika Rahmadian. 2014. : Desain dan Implementasi Sistem Keamanan E–*voting* dengan Jaminan *Confidentiality Data.* Skripsi Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Dosen Pembimbing : Aswin Suharsono, S.T,.M.T. dan Ahmad Afif Supianto, S.Si., M.Kom.

Proses pemungutan suara dilakukan dalam bentuk informasi digital dengan menggunakan *e-voting*. Dengan *e-voting* proses pemungutan suara dapat menghemat biaya, cepat dan akurat dalam penghitungan suara, kemudahan, dan aman dalam penggunaan sistem. *Confidentiality* data merupakan aspek penting dalam menjaga kerahasiaan dan menjamin keamanan data. Penerapan *confidentiality* dalam *e-voting* yaitu untuk menjamin keamanan dalam hal kerahasiaan data hasil *voting* masing-masing pemilih. Jaminan *confidentiality* data pada *e-voting* menggunakan metode enkripsi dengan algoritma kriptografi kunci-publik *RivestShamirAdleman* (RSA) untuk memastikan keamanan data pada saat proses *voting*. Berdasarkan dari hasil pengujian validasi dengan metode *Black-box* testing menunjukan bahwa aplikasi telah berjalan sesuai rancangan yang telah dibangun. Proses jaminan *confidentiality* data pada sistem *e-voting* telah berjalan dengan baik dan telah valid karena tidak ada perubahan data sebelum proses dan sesudah proses enkripsi dan dekripsi pada data hasil *voting* oleh *voter*. Sistem *e-voting* ini telah di implementasikan sesuai dengan perancangan dan dapat digunakan untuk melakukan proses *voting* dan dapat menjamin *confidentiality* data hasil *voting*.

Kata Kunci : *e-voting, confidentiality*, kriptografi kunci-publik

ABSTRACT

Tika Rahmadian. 2014. : *Design and Implementation of E-Voting Security System with the Assurance of Confidentiality Data.* Skripsi Program Studi Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Supervisors : Aswin Suharsono, S.T,.M.T. dan Ahmad Afif Supianto, S.Si., M.Kom.

Voting process is conducted in the shape of digital information by using e-voting. With e-voting the voting process can save cost, fast and accurate in vote count, easy, and secure in the usage of the system. Confidentiality of data is an important aspect to keep secrecy and ensuring data security. The applying of confidentiality in the e-voting is to ensure security in terms confidentiality data of voting results on each voter. Ensuring of confidentiality data on e-voting using an encryption method with a public key cryptographic algorithm RivestShamirAdleman (RSA) to ensure the data security during the process of voting. Based on the results of the validation tests with Black-box testing method indicates that the application has been going according to plan which has already been built. The process ensuring the confidentiality data on e-voting system has been running very well and has been valid since there is no change to the data before the process and after the process of encryption and decryption on the voting results by voter data. E-voting system has been implemented in accordance with the design and can be used to do the voting process and can ensuring the confidentiality of the data results of the voting.

Keywords: *e-voting, confidentiality, public-key cryptography*

DAFTAR ISI

Daftar Gambar

Daftar Tabel

BAB I
PENDAHULUAN

1.1 Latar Belakang

Proses pemungutan suara saat ini masih menggunakan cara sederhana yaitu dengan media kertas dan penghitungan hasil suara secara *manual*. Cara tersebut memiliki kendala pada proses penghitungan suara yang berjalan lambat dan sulitnya penghitungan ulang apabila ada yang tidak percaya pada hasil penghitungan suara. Dengan memanfaatkan teknologi komputer, proses pemungutan suara dilakukan dalam bentuk informasi digital. Pemungutan suara elektronik (*e-voting*) merupakan suatu sistem pemilihan yang bersifat rahasia secara elektronik [KTE-10]. Dengan *e-voting* proses pemungutan suara dapat menghemat biaya, cepat dan akurat dalam penghitungan suara, kemudahan dan keamanan dalam penggunaan sistem. Sistem *e-voting* memiliki beberapa aspek untuk jaminan keamanan data antara lain *authentication, confidentiality, non-repudiation* dan *integrity* [ACO-96].

E-voting mulai banyak digunakan oleh beberapa negara yaitu Amerika Serikat, Australia, Brazil, Estonia, Filipina, India, dan Perancis [ALP-11]. Di Indonesia penggunaan *e-voting* masih dalam wacana akan tetapi sudah dilakukan di Jembrana, Bali [KTE-10]. Pemilu di Indonesia menurut UU No. 12 tahun 2003 berasaskan langsung, umum, bebas, rahasia, jujur, dan adil. Rahasia artinya pemilih telah memberikan suara dan tidak ada orang lain yang mengetahui apa isi suara tersebut. Asas rahasia pada sistem *e-voting* diterapkan untuk mengamankan data hasil *voting* oleh *voter*. Tetapi dalam pelaksanaan asas rahasia pada sistem *e-voting* dapat terjadi manipulasi pada total penghitungan suara yang sudah disimpan dan menyebabkan tidak adanya kepercayaan pada data hasil *e-voting*. Oleh karena itu *e-voting* membutuhkan metode penyimpanan data hasil pilihan masing-masing pemilih yang menjamin keamanan data hasil pilihan pemilih. Sistem *e-voting* pada penelitian sebelumnya yang menjamin *confidentiality* telah dibahas pada *An Analysis and Recommendations for an E-voting System* [AJM-04] dengan kriptografi simetris. Tetapi jaminan *confidentiality* menggunakan kriptografi

simetris memungkinkan kunci dapat diketahui oleh *Administrator*. Sehingga *Administrator* dapat merubah data hasil *voting*. Selain itu sistem *e-voting* pada penelitian sebelumnya dengan jaminan *confidentiality* dan *authentication* juga telah di bahas *Pada E-Voting Protocol Based On Public-Key Cryptography* [EPC-11] dengan kriptografi asimetris. Tetapi pada sistem tersebut membutuhkan *Administrator* untuk mendekripsi hasil *voting* yang telah di enkripsi sebelum dicetak dan diumumkan hasilnya. Hal tersebut memungkinkan *Administrator* dapat melakukan manipulasi data hasil *voting*.

Confidentiality memegang peranan penting dalam menjaga kerahasiaan dan menjamin keamanan data. Penerapan *confidentiality* dalam *e-voting* yaitu untuk menjamin keamanan dalam hal kerahasiaan data hasil *voting* masing-masing pemilih. Jaminan *confidentiality* data pada *e-voting* dapat diberikan dengan metode enkripsi algoritma RSA untuk memastikan keamanan data pada saat proses *voting*. RSA yang diciptakan oleh Ron Rivest, Adi Shamir dan Len Adleman merupakan algoritma kriptografi asimetri, dimana kunci yang digunakan untuk mengenkripsi data berbeda dengan yang digunakan untuk mendekripsi data (*public-key* dan *private-key*). Kekuatan keamanan algoritma RSA terletak pada tingkat kesulitan memfaktorkan modulus *n* menjadi *p* dan *q* (faktorisasi bilangan prima). Hal tersebut dikarenakan dapat ditentukannya besar nilai *p* dan *q* pada saat proses *generate* pasangan kunci [EMR-09]. Dengan RSA keamanan data hasil *e-voting* dapat dijamin kerahasiaannya.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dijabarkan, maka dirumuskan beberapa permasalahan :

1. Bagaimana merancang dan mengimplementasikan sistem keamanan *e-voting* yang menjamin *confidentiality* data?
2. Bagaimana menerapkan sistem keamanan *e-voting* yang menjamin *confidentiality* data?
3. Bagaimana pengujian keamanan pada *e-voting*?

1.3 Batasan Masalah

Berdasarkan rumusan masalah yang telah diuraikan, maka berikut dibuat beberapa batasan masalah :

1. Sistem pemungutan suara elektronik (*e-voting*) diakses dalam bentuk *website*.
2. Keamanan sistem hanya fokus pada aspek *confidentiality*.
3. Jaminan *confidentiality data* di fokuskan pada pengamanan data hasil *voting* oleh *voter*.
4. Keamanan data hasil *voting* menggunakan kriptografi kunci publik yaitu RSA.
5. Pengujian sistem *e-voting* dilakukan dengan validasi menggunakan metode *Black-box*.
6. Pengujian kriptografi pada sistem dilakukan dengan validasi data yang terenkripsi dengan RSA.
7. Pengujian keamanan dilakukan dengan skenario manipulasi data oleh Administrator dengan cara mengubah (*update*) data pilihan *voter*.

1.4 Tujuan

Berdasarkan uraian rumusan masalah di atas, maka dapat diketahui tujuan pada tugas akhir ini yaitu sebagai berikut:

1. Merancang dan mengimplementasikan sistem keamanan pada *e-voting*.
2. Menerapkan sistem keamanan pada *e-voting* menggunakan kriptografi kunci-publik yaitu RSA.
3. Melaksanakan pengujian validalitas pada sistem keamanan *e-voting* yang menggunakan kriptografi kunci-publik yaitu RSA.

1.5 Manfaat

Manfaat dari penelitian ini adalah sebagai berikut :

- Bagi Penulis :
  1. Menerapkan ilmu yang telah diperoleh dari Informatika di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
  2. Mendapatkan pemahaman tentang sistem *e-voting*.
  3. Mendapatkan pemahaman tentang kriptografi kunci-publik.

- Bagi Pengguna :
  1. Dapat menerapkan pada pemilihan–pemilihan organisasi mahasiswa di Program Teknologi Informasi dan Ilmu Komputer.
  2. Memberikan kemudahan dan jaminan keamanan data pada saat proses *voting.*

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini sebagai berikut:

BAB I Pendahuluan

Memuat latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II Kajian Pustaka dan Dasar Teori

Berisi kajian tentang penelitian *e-voting* sebelumnya dan berisi tentang teori dasar dan referensi secara luas yang diperlukan unruk implementasi sistem keamanan *e-voting* dengan jaminan *confidentiality* data.

BAB III Metode Penelitian dan Perancangan

Berisi tentang analisis kebutuhan, penjelasan penelitian dan perancangan sistem keamanan dalam mengimplementasi sistem perangkat lunak untuk keamanan *e-voting* dengan jaminan *confidentiality* data.

BAB IV Implementasi Sistem

Berisi mengenai implementasi keamanan pada aplikasi yang dibangun, diantaranya pembuatan aplikasi dengan menggunakan pemrograman PHP dan OpenSSL untuk keamanan data.

BAB V Pengujian dan Analisis

Berisi mengenai proses, hasil pengujian dan analisa hasil pengujian terhadap sistem yang telah dibangun.

BAB VI Penutup

Memuat kesimpulan yang diperoleh dari hasil akhir pembuatan dan pengujian perangkat lunak yang dikembangkan dalam skripsi serta saran-saran untuk pengembangan lebih lanjut.

BAB II
KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Kajian Pustaka

Kajian pustaka pada bab ini membahas tentang penerapan dan penelitian sebelumnya yang berkaitan dengan *e-voting* serta jaminan keamanannya. Penjelasan tentang bagaimana penerapan *e-voting* di beberapa negara, penerapan *e-voting* di Indonesia, dan metode keamanan *e-voting*.

2.1.1 E-Voting di Beberapa Negara

Penelitian dan pemanfaatan elektronik pada proses pemungutan suara (*e-voting*) menggantikan proses pemungutan suara secara *manual*. Penerapan *e-voting* sudah dilakukan oleh beberapa negara. Penerapan tersebut dilakukan pada pelaksanaan pemilihan lokal sampai pemilihan umum berdasarkan metode, strategi dan tahapan yang dimiliki. Berikut ini adalah negara-negara yang telah menerapkan *e-voting* yang dibahas pada Prospek dan Tantangan Penerapan E–*Voting* di Indonesia [ALP-11].

1. Amerika Serikat

Menurut data *Aceproject*, di Amerika e-*voting* baru mencakup sepertiga jumlah pemilih. Pada pemilihan presiden tahun 2004, muncul permasalahan di sejumlah tempat pemungutan suara. Pemilih tidak bisa memverifikasi apakah mesin *e-voting* dapat mencatat suara dengan baik dan petugas pemilu pun tidak mungkin melakukan penghitungan ulang. Sehingga timbul kekhawatiran terhadap keamanan penggunaan mesin e-*voting*. Muncul juga masalah serius bagaimana menjamin integritas hasil pemilihan presiden yang digelar saat itu dimana pada 2004 pemilu presiden diikuti George W Bush dari Republik, dan John Kerry dari Demokrat. Setelah melihat beberapa masalah tersebut, muncul ide untuk melengkapi mesin *e-voting* dengan teknologi tambahan yang memungkinkan suara yang telah diberikan telah diverifikasi. Bentuknya berupa struk yang keluar dari mesin e-*voting* sebagai bukti. Teknologi ini dikenal dengan sebutan *voter verifiable paper audit trail* (VVPAT).

Selain masalah pada mesin *e-voting* di Amerika terdapat masalah lainnya. Seperti yang dilaporkan oleh *Electronic Frontier Foundation* (EFF) yaitu adanya

masalah pada SDM yang tidak terlatih. Selain itu, EFF juga menyatakan teknisi dari vendor mesin *e-voting* pun masih memiliki akses yang tidak terawasi terhadap peralatan e-*voting*. Staf KPU lokal pun, selalu menolak audit data. Masalah juga terjadi pada teknologi *internet voting* (*remote e-voting*). Teknologi ini digunakan oleh warga negara Amerika yang berada di luar negeri (ekspatriat). Tetapi, teknologi yang disebut dengan *Secure Electronic Registration and Voting Experiment* (SERVE) dihentikan pada tahun 2004, setelah petugas dari Departemen Pertahanan AS menemukan suatu masalah bahwa sistem tersebut tidak cukup aman untuk mentransfer suara pemilih.

Menurut data *International Foundation for Electoral System* (IFES), sampai dengan tahun 2004, dari 50 negara bagian di Amerika, 80% diantaranya masih menggunakan surat suara *manual* dan penghitungan suaranya menggunakan pemindai optik (*e-counting*). Negara bagian lainnya menggunakan surat suara *manual* dan *e-voting*. Satu Negara bagian menggunakan surat suara *manual* dan *punch card*, 10 negara bagian menggunakan surat suara *manual* dan teknologi DRE serta VVPAT, 4 negara bagian menggunakan surat suara *manual* dengan teknologi DRE dengan atau tanpa VVPAT, 7 negara bagian menggunakan surat suara *manual* dengan teknologi DRE tanpa VVPAT (antara lain Lousiana, Georgia, dan South Carolina). Yang benar-benar menerapkan teknlogi DRE dengan VVPAT hanya 2 negara bagian, yaitu Nevada dan Utah. Hingga saat ini, Amerika Serikat masih digolongkan sebagai negara yang bermasalah dalam penerapan *e-voting*. Negara lain yang juga masih bermasalah dengan penerapan *e-voting* adalah Jerman, Belanda, dan Irlandia. Adapun negara-negara yang berhasil menerapkan *e-voting* yaitu India dan Brazil.

2. Australia

Oktober 2001 *e-voting* telah diterapkan dan digunakan pertama kali dalam pemilihan anggota parlemen Australia. Pemilu tersebut diikuti oleh 16.559 pemilih yang menggunakan hak pilihnya secara elektronik ditempat pemungutan suara (TPS). Kemudian Pemerintah Negara Bagian *Victoria* memperkenalkan *e-voting* sebagai uji coba pada tahun 2006. Pada tahun 2007 para personil angkatan bersenjata Australia yang ditempatkan di Irak, Afghanistan, Timor Leste, dan Kepulauan Solomon telah diberi kesempatan untuk menggunakan hak pilihnya

melalui jaringan khusus departemen pertahanan sebagai bagian dari proyek kerjasama antara departemen pertahanan dengan komisi pemilu Australia. Setelah mereka menggunakan hak pilih kemudian datanya dienskripsi dan dikirimkan melalui *Citrix server* ke *database*. Sebanyak 2.012 personil terdaftar sebagai pemilih dan dari jumlah tersebut 1.511 orang berhasil menggunakan hak pilihnya.

3. Brazil

E-voting di Brazil diperkenalkan pertama kali pada tahun 1996 yaitu ketika dilakukan uji coba di Negara Bagian Santa Catarina. Sejak tahun 2000 semua pemilu yang diselenggarakan di Brasil telah dilakukan secara elektronik. Pada tahun 2002 lebih dari 400.000 mesin *e-voting* telah digunakan di seluruh wilayah Brazil. Data hasil pemilu di Brazil dihitung secara elektronik yang hasilnya dapat diketahui dengan cepat dalam hitungan menit setelah pemilu selesai.

4. Estonia

E-voting di Estonia telah diterapkan mulai bulan Oktober 2005 pada pemilu lokal. Estonia menjadi negara pertama yang menyelenggarakan pemilu melalui *Internet* dan telah dinyatakan berhasil oleh pejabat pemilu Estonia. Sebanyak 9.317 orang telah menggunakan hak pilihnya secara *online*. Pada tahun 2007 Estonia dinobatkan sebagai negara yang menyelenggarakan *e-voting* melalui *Internet* secara nasional. Pemilu yang dilakukan melalui *Internet* telah dilaksanakan selama dua hari pada bulan Februari dan telah berhasil membuat 30.275 orang yang menggunakan hak pilihnya melalui *Internet*. Tahun 2009 pada pemilu lokal telah berhasil memfasilitasi 104.415 orang yang menggunakan hak pilihnya melalui *Internet*. Data menyebutkan 9,5% dari total pemilih telah menggunakan hak pilihnya melalui *Internet*. Tahun 2011 pada pemilihan anggota parlemen pada tanggal bulan Februari-Maret, sebanyak 2.140.846 orang telah memilih secara *online*. 95% pemilih menggunakan hak pilihnya di dalam negeri dan sisanya memilih dari luar negeri yang tersebar di 106 negara.

5. Filipina

Pada bulan Mei 2010 Pemerintah Filipina telah merencanakan untuk menerapkan penyelenggaraan pemilu secara eletronik untuk pertama kali dengan menggunakan *optical scan voting system*. Penerapan *e-voting* secara nasional dimaksudkan untuk meningkatkan akurasi dan kecepatan dalam penghitungan

suara. Serta diharapkan dapat mengurangi kecurangan dan korupsi sebagaimana ditemukan pada pemilu-pemilu di Filipina yang telah diadakan sebelumnya. Pada tanggal 3 Mei 2010, Filipina telah melakukan pre-*test* terhadap sistem *e-voting*. Komisi Pemilu (Comelec) telah menemukan 76.000 dari total 82.000 mesin *scan* optik terdapat kegagalan dalam kartu memori.

Mesin bermasalah dalam menghitung dan memberikan suara kepada kandidat lawan. Kemudian setelah dilakukan penyesuaian antara penghitungan *manual* dan elektronik, kartu memori kemudian diganti untuk seluruh wilayah. Hal tersebut menyebabkan banyak pemilih yang meragukan penerapan e-*voting* setelah kejadian tersebut. Rakyat Filipina telah memilih presiden menggunakan e-*voting* untuk pertama kalinya pada pemilu bulan Mei 2010. KPU Filipina melaporkan bahwa hanya 400 dari 82.000 mesin e-*voting* yang tidak berfungsi. Terdapat beberapa masalah yaitu pemilih mengeluhkan panjangnya antrian dan butuh waktu lama untuk mempelajari teknologi baru.

6. India

E-voting diperkenalkan pertama kali pada tahun 1982 dan digunakan pada waktu uji coba untuk pemilihan Majelis Bort Parur di Negara Bagian Kerala. Namun Mahkamah Agung India membatalkan hasil pemilu tersebut karena tidak sesuai dengan hukum yang berlaku di sana. Atas dasar masalah hukum tersebut kemudian dilakukan amandemen terhadap Undang-undang Perwakilan Rakyat untuk mengesahkan pemilu yang diselenggarakan melalui *Electronic Voting Machine* (EVMs). Pada tahun 2003 semua pemilu di negara bagian telah menggunakan EVMs. EVMs juga telah digunakan pada pemilu nasional untuk memilih anggota parlemen India pada tahun 2004 dan 2009.

Menurut data statistik yang bersumber dari media massa utama di India, lebih dari 400 juta pemilih yaitu 60% dari pemilih yang terdaftar telah menggunakan hak mereka melalui EVMs pada pemilu tahun 2009. Keberhasilan penerapan e-*voting* di India karena sistem pemilunya yang sederhana. India menggunakan *system first past the post* atau sistem distrik yang merupakan bentuk alternatif paling sederhana dan mudah. Yaitu, hanya ada satu kandidat dari setiap partai di surat suara (*single member distric*).

7. Perancis

Pada Januari 2007 Partai *Union for a Popular Movement* (UMP) telah menerapkan dan menyelenggarakan pemilihan presiden dengan menggunakan *remote e-voting* dan menyediakan layar sentuh pada 750 tempat pemilihan. Pemilihan telah diikuti 230.000 suara yang mewakili hampir 70% dari daftar pemilih. Pemilu di Perancis diselenggarakan pada tahun 2003 secara *online* melalui *Internet* dan untuk warga negara Perancis yang tidak berada di Perancis, memilih wakil mereka yang akan duduk dalam Majelis Warga Perancis dari luar negeri. Lebih dari 60% pemilih menggunakan haknya untuk pemilu melalui *Internet*.

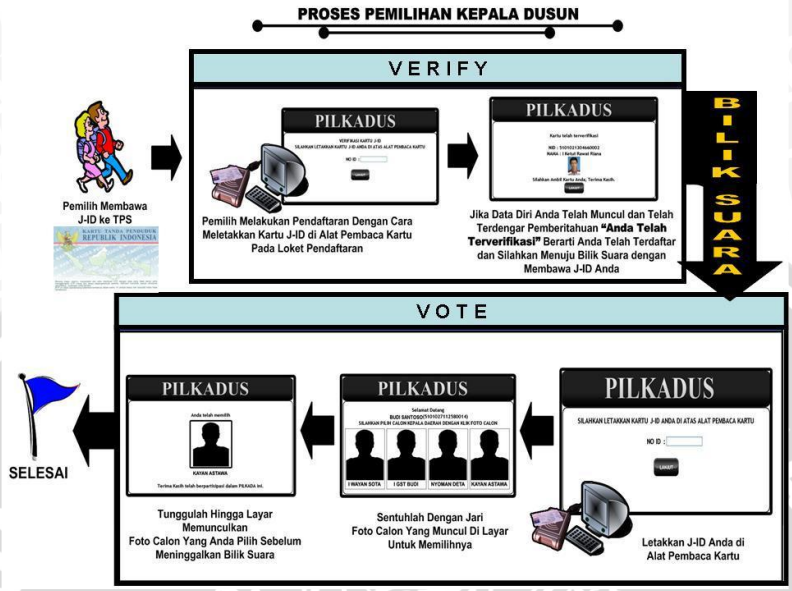
Dari penerapan *e-voting* yang dilakukan diberbagai negara, *e-voting* memiliki beberapa masalah yang harus diperbaiki. Masalah-masalah tersebut antara lain:

- Penerapan *e-voting* harus dilengkapi dengan adanya perlindungan hukum yang jelas tentang penerapan *e-voting* mulai dari tahap persiapan sampai dengan pengesahan hasil pemungutan suara
- Keamanan dan kehandalan *e-voting* merupakan aspek penting dalam menjaga kevalidan data selain kecepatan penghitungan dan pendistribusian hasil penghitungan suara
- Pemahaman teknologi akan *e-voting* yang dilakukan dengan *online* atau menggunakan DRE diperlukan agar tidak menghambat pelaksanaan
- Para saksi dan pengawas juga harus siap terhadap *e-voting*. Untuk itu perlu dilakukan pelatihan untuk saksi dan pengawas agar dapat memiliki kompentensi apabila terjadi permasalahan dalam pelaksanaan *e-voting*
- Resiko politik pada penerapan *e-voting* yang berkaitan dengan keabsahan hasil pemilu. Apabila pemilu gagal menyebabkan ketidakstabilan politik suatu negara
- Penerapan *e-voting* yang membutuhkan tenaga ahli dalam keamanan teknologi informasi dan pemahaman pada sistem pemilihan

2.1.2 E-Voting di Indonesia

Penerapan *e-voting* di Indonesia masih dalam wacana dan perlu dilakukan pertimbangan yang baik dan benar dari beberapa aspek dalam pelaksanaannya. Akan tetapi sudah ada provinsi di Indonesia yang menerapkan *e-voting* untuk Pemilihan Kepala Dusun yaitu Jembrana, Bali. Pemerintah Kabupaten Jembrana, Bali telah melakukan Pilkadus dengan sistem *e-voting* sejak tahun 2009. Data

pemilih diperoleh dari *Database* Sistem Informasi Administrasi Kependudukan (SIAK) yang dimasukkan ke dalam komputer *e-voting*. Hingga tahun 2010, telah diselenggarakan *e-voting* pilkadus sebanyak 60 kali [KTE-10].



Gambar 2.1 Proses Pilkadus di Kabupaten Jembrana, Bali
Sumber: [KTE-10]

Pada gambar 2.1 proses pemungutan suara pada Pilkadus di Kabupaten Jembrana berlangsung sebagai berikut:

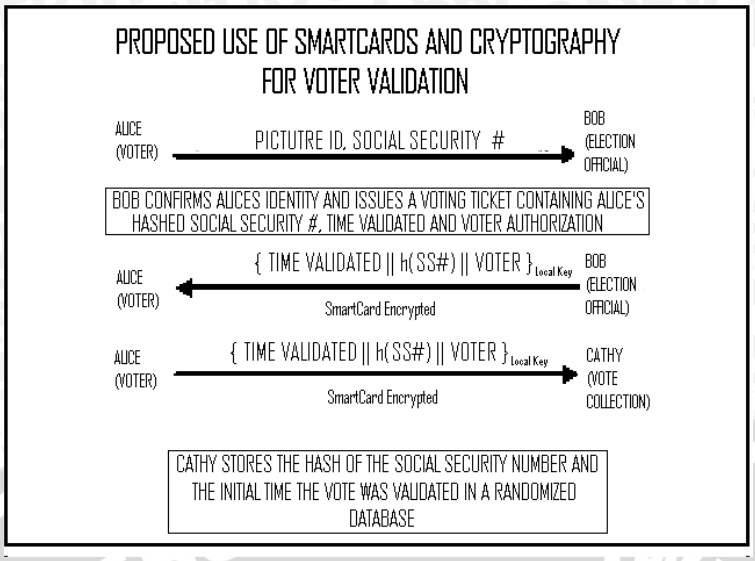
1. Pemilih melakukan pendaftaran dengan cara meletakkan kartu J-ID pada alat pembaca kartu pada loket pendaftaran.
2. Apabila data diri pemilih telah muncul dan telah muncul pemberitahuan "Anda telah terverifikasi" berarti pemilih telah terdaftar dan disilahkan menuju bilik suara dengan membawa kartu J-ID.
3. Letakkan kartu J-ID pada alat pembaca kartu.
4. Sentuh dengan jari foto calon yang muncul pada layar untuk memilihnya.
5. Menunggu hingga layar memunculkan foto calon yang dipilih sebelum meninggalkan bilik suara.

2.1.3 Metode Keamanan E-Voting

Berikut ini beberapa persyaratan keamanan yang harus dipenuhi dalam suatu sistem *e-voting* menurut *A Survey of Modern Electronic Voting Technologies* [MSE-10]:

1. *Eligibility and authentication* : pemilih hanya diperbolehkan melakukan proses *vote* setelah proses otentikasi.
2. *Uniqueness* : pemilih hanya memiliki kesempatan memilih satu kali (memiliki satu suara yang berlaku).
3. *Accuracy* : hasil *vote* dicatat dan disimpan dengan benar.
4. *Integrity* : hasil *vote* tidak dapat dirubah dan dihapus.
5. *Verifiability* dan *accountability* : sistem memiliki fasilitas untuk memverifikasi hasil *vote* dihitung dengan benar.
6. *Reliability* : sistem pemilihan harus dapat bekerja dengan baik dalam menghadapi beberapa masalah seperti kehilangan hasil *vote*, kegagalan mesin *voting* dan kerugian komunikasi *internet.*
7. *Secrecy* dan *non-coercibility* : pilihan yang dilakukan oleh pemilih harus rahasia, tidak ada yang mengetahui pilihan yang dipilih oleh pemilih, dan pemilih tidak memiliki catatan pilihannya.
8. *Flexibility* : sistem dapat digunakan oleh berbagai jenis pemilih.
9. *Convenience* : pemilih mendapatkan kenyamanan pada saat melakukan proses *voting.*
10. *Certifiability* : sistem *e-voting* yang akan digunakan harus diuji oleh petugas pemilu terlebih dahulu.
11. *Transparency* : pemilih harus mampu memahami sistem secara umum.
12. *Cost* : sistem yang digunakan tidak boleh terlalu mahal.

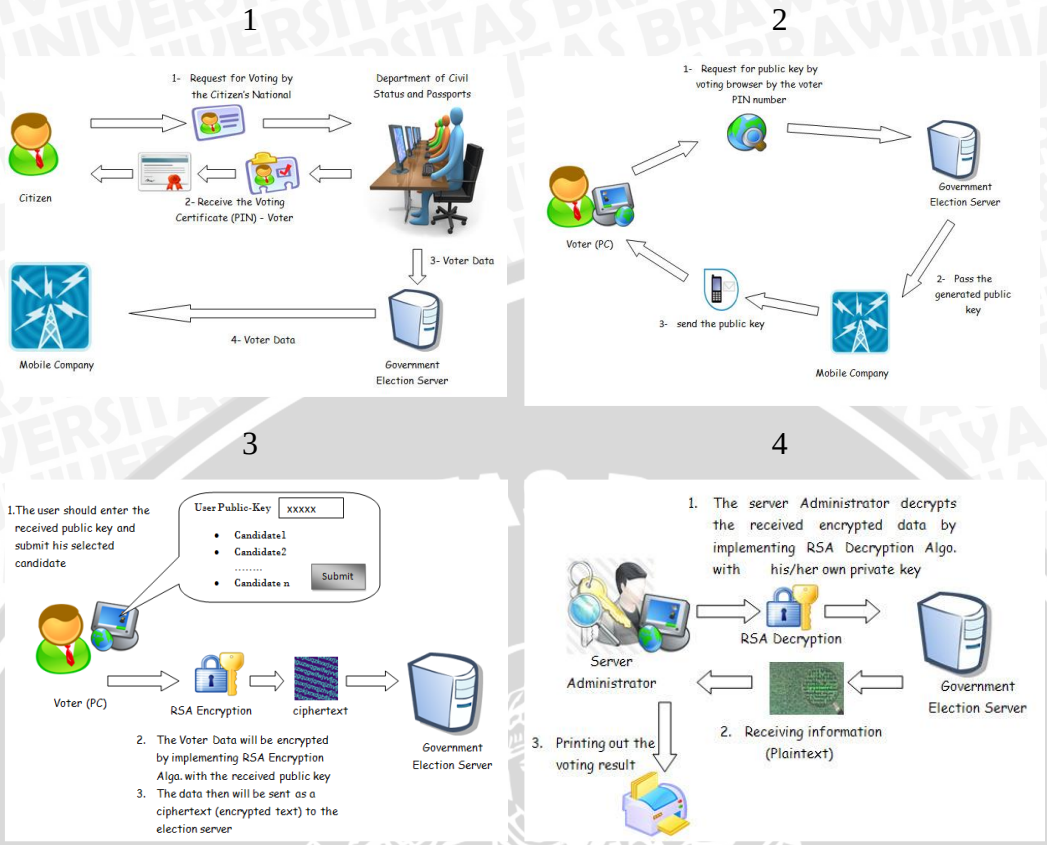
Metode keamanan *e-voting* yang pada penelitian sebelumnya yang dibahas pada *An Analysis and Recommendations for an E-voting System* [JAM-04] yaitu menggunakan enkripsi kriptografi simetris yaitu AES dan DES untuk jaminan *confidentiality*. Jaminan *confidentiality* dengan kriptografi simeteris yaitu DES digunakan untuk mengamankan informasi pemilih yang disimpan di *database* dan AES digunakan untuk mengamankan data hasil *voting* oleh pemilih yang ditransfer ke server.



Gambar 2.2 Manajemen Kunci Untuk Proses *Voting*
Sumber: [AJM-04]

Pada gambar 2.2 menunjukan proses pemilih melakukan proses *voting* dengan *smartdcard*. *Smartcard* memverivikasi identitas pemilih untuk melakukan pemungutan suara. Setelah pemilih melakukan *voting*, hasil *voting* sudah terenkripsi disimpan di *database*. Pada saat proses pengiriman hasil *voting* pada *database*, hasil *voting* diberi keamanan menggunakan *hash*.

Metode keamanan *sistem e-voting* pada penelitian sebelumnya yang dibahas pada *E-Voting Protocol Based On Public-Key Cryptography* [EPC-11] memiliki *authentication* pada saat proses pendaftaran pemilih dan *confidentiality* pada proses *voting*. Pada gambar 2.3 nomor 1 merupakan proses *authentication* sistem *e-voting* untuk pendaftaran calon pemilih, pemilih dipastikan verivikasi identiasnya untuk mendapatkan PIN yang digunakan untuk *voting*.



Gambar 2.3 Sistem Keamanan *E-Voting* pada *E-Voting Protocol Based On Public-Key Cryptography*
Sumber: [EPC-11]

Pada gambar 2.3 nomor 2, pemilih sebelum melakukan proses *voting* harus melalui proses *authentication*. Dengan merequest PIN ke server untuk mendapatkan kunci-publik. Selanjutnya pada gambar 2.3 nomor 3 proses *voting* dilakukan setelah pemilih mendapatkan kunci-publik. Pemilih menentukan pilihan, lalu mensubmit. Data pilihan oleh pemilih tersebut terenkripsi oleh kunci-publik dan masuk *database*. Pada gambar 2.3 nomor 4 Proses penghitungan suara yaitu dengan mendekripsi hasil pilihan masing-masing pemilih. *Administrator* memiliki kunci privat kemudian mendekripsi hasil *voting* sebelum dicetak dan diumumkan hasilnya.

2.2 Kriptografi

Kriptografi adalah ilmu dan teknik yang berhubungan dengan aspek keamanan data informasi. Aspek kriptografi dalam keamanan informasi yang dibahas pada *A Handbook Of Applied Cryptography* [ACO-96] ada 4, yaitu :

1. *Confidentiality*

Confidentiality adalah layanan yang digunakan untuk menjaga isi suatu informasi dari semua pihak, akan tetapi hanya untuk mereka yang memiliki kewenangan untuk melihat dan mengaksesnya.

2. *Data Integrity*

Data integrity adalah layanan yang berhubungan dengan perubahan data yang tidak sah. Untuk menjamin integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak yang tidak berhak. Manipulasi data mencakup hal-hal seperti penyisipan, penghapusan, dan substitusi.

3. *Authentication*

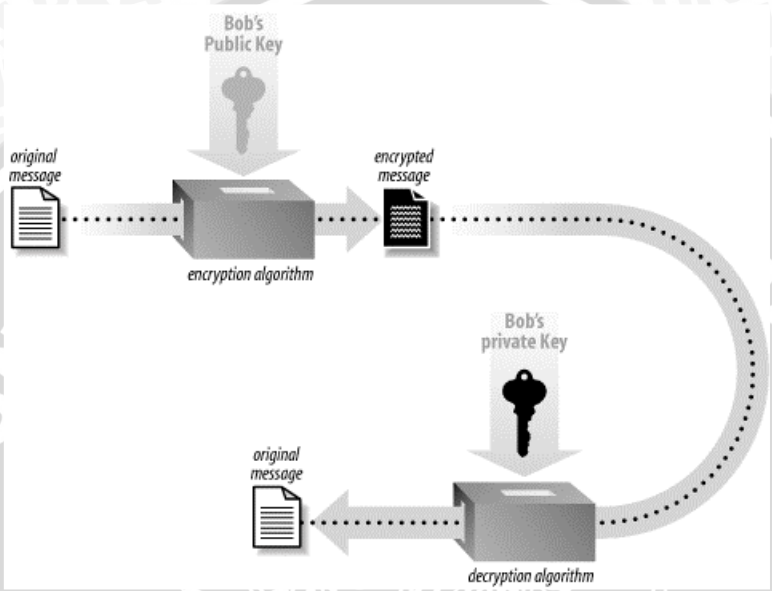
Authentication adalah layanan yang berhubungan dengan identifikasi. Fungsi ini berlaku untuk kedua entitas dan informasi itu sendiri. Dua informasi yang masuk ke dalam komunikasi harus mengidentifikasi satu sama lain. Informasi disampaikan melalui saluran harus dikonfirmasi dengan tanggal asal, isi data, waktu dikirim, dan lain-lain. Dalam hal ini aspek kriptografi biasanya dibagi menjadi dua kelas utama: otentikasi entitas dan otentikasi data asal. Otentikasi data asal secara langsung memberikan integritas data. Hal tersebut terjadi karena jika pesan dimodifikasi atau sumber telah berubah.

4. *Non-repudiation*

Non-repudiation adalah layanan yang mencegah suatu entitas dari penyangkalan komitmen atau tindakan sebelumnya. Ketika permasalahan timbul karena suatu entitas menyangkal bahwa tindakan tertentu diambil, diperlukan sarana untuk mengatasinya. Contohnya, satu entitas dapat mengizinkan pembelian properti oleh entitas lain dan kemudian menolak otorisasi tersebut diberikan. Hal tersebut membutuhkan sebuah prosedur yang melibatkan pihak ketiga yang terpercaya untuk menyelesaikan masalah.

2.2.1 Algoritma Kunci Publik

Algoritma kunci publik atau yang lebih dikenal dengan kunci asimetris merupakan algoritma yang memiliki sepasang kunci yaitu *public key* dan *private key*. Disebut kunci asimetris karena kunci enkripsi tidak sama dengan kunci dekripsi [RSK-04]. Kunci untuk enkripsi (*public key*) diumumkan kepada publik sedangkan kunci untuk dekripsi (*private key*) bersifat rahasia.



Gambar 2.4 Sistem Kriptografi Kunci-Publik
Sumber: [NSO-02]

Pada gambar 2.4 merupakan proses pengiriman pesan atau data oleh Alice kepada Bob. Alice terlebih dahulu harus memiliki kunci publik Bob. Alice mengenkripsi pesan atau data yang akan dikirimkan kepada Bob dengan kunci publik lalu mengirimkannya. Pesan Alice yang dienkripsi kunci publik dan telah sampai kepada Bob dan dapat dibuka (di dekripsi) oleh kunci privat yang hanya dimiliki oleh Bob.

2.2.1.1 RSA

RSA termasuk algortima kriptografi kunci-publik. RSA di temukan oleh R. Rivest, A. Shamir dan L. Adleman. Merupakan cryptosystem kunci publik yang banyak digunakan saat ini. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor bilangan prima. Pemfaktoran tersebut dilakukan untuk mendapatkan *private-key*. Keamanan dengan algoritma RSA tetap terjamin apabila pemfaktoran bilangan besar tidak dapat

menjadi faktor bilangan prima [RSK-04]. RSA juga dikombinasikan dengan skema *padding* untuk menjaga pesan agar aman. Ukuran rata-rata nilai *n* harus meningkatkan dengan waktu sebagai algoritma anjak lebih efisien dibuat dan sebagai komputer semakin cepat. Para peneliti menyarankan nilai panjang *n* RSA lebih dari 200-digit (663bit). RSA dengan panjang nilai *n* 512bit dapat dipecahkan kurang lebih dalam 5 bulan. Saat ini, kunci RSA yang digunakan bernilai antara 1024 dan 2048 bit panjang. Panjang nilai *n* RSA yang tertinggi adalah 4096bit dan sejauh ini tidak terpecahkan. Sebuah nilai *n* tidak lebih dari 300 bit dapat difaktorkan pada PC dalam beberapa jam [EMR-09].

Selain itu RSA RSA dapat digunakan untuk *key distribution* dan *key exchange*, serta *digital signature*. Karena merupakan sistem pertama yang dapat digunakan untuk *key distribution* dan *digital signature*, RSA menjadi sistem kriptografi *public key* yang terpopuler. Semua *standard* protokol kriptografi menggunakan RSA, termasuk SSL/TL (pengamanan http) dan SSH (*secure shell*) [SKT-09].

1. *Generate Key* (Pembangkitan Kunci)

Generate key untuk membuat kunci privat dan kunci publik dilakukan dengan langkah-langkah berikut [RSK-04] :

1. Pilih 2 bilangan prima secara acak untuk masing-masing p dan q, $p \neq q$
2. Hitung n = p . q
3. Hitung $\varphi = (p - 1)(q - 1)$
4. Pilih bilangan bulat (*integer*) antara 1 dan $\varphi(1 < e < \varphi)$ yang juga relatif prima terhadap φ
5. Hitung d hingga $d\ e \equiv 1 \pmod{\varphi}$

Keterangan :

- Bilangan prima dapat diuji probabilitasnya menggunakan *Fermat's little theorem*- a^(n – 1) mod n = 1 jika n adalah bilangan prima.
- Langkah 2 dapat menggunakan $\varphi = (p - 1, q - 1)$, (gcd,φ)=1.
- Langkah nomor 3 dapat dihasilkan dengan algoritma *extended Euclidean.*
- Langkah nomor 4 dapat dihasilkan dengan menemukan bilangan x sehingga $d = (x\ \varphi + 1)/e$ menghasilkan bilangan bulat, kemudian menggunakan nilai dari $d \pmod{\varphi}$.

Pada *public key* diberikan :

- *n*, modulus yang digunakan
- *e*, eksponen publik (eksponen enkripsi)

Pada *private key* diberikan :

- *n*, modulus yang digunakan (digunakan juga pada *public key*)
- *d*, eksponen privat (eksponen dekripsi) yang bersifat rahasia

2. Enkripsi dan Dekripsi Pesan

Enkripsi :

1. Mengambil kunci publik penerima pesan (*e*) dan modulus (*n*)
2. Membagi *plaintext* menjadi blok-blok m_1, m_2, ..., sehingga setiap blok merepresentasikan nilai [0, $n - 1$]
3. Setiap blok m_i dienkripsi menjadi blok c_i dengan $c_i = m_i^e \bmod n$

Dekripsi :

1. Setiap blok *ciphertext* c_i didekripsi kembali menjadi blok m_i dengan $m_i = c_i^d \bmod n$
2. *d* merupakan kunci privat

2.3 OpenSSL

OpenSSL adalah impelementasi *open source* dari protokol SSL (*Secure Socket Layer)* dan TLS (*Transport Layer Security*). OpenSSL dikembangkan berdasarkan pada SSLeay oleh Eric A. Young dan Tim J. Hudson. OpenSSL merupakan *library* kriptografi yang digunakan untuk implementasi algoritma kriptografi seperti DES, AES, RSA dan lain-lain. OpenSSL juga merupakan alat yang menyediakan akses ke banyak fungsionalitas dari *command-line*. *Command-line* memudahkan untuk melakukan operasi umum, seperti menghitung kunci enkripsi, *hash*, dan MD5 dari isi file. *Command-line* menyediakan kemampuan untuk mengakses fungsi OpenSSL tingkat tinggi dari skrip *shell* di Unix atau *batch file* pada *Windows*. Alat bantu menjalankan *command-line* pada Unix bernama openssl dan pada Windows bernama openssl.exe [NSO-02].

2.3.1 Algoritma Public Key pada OpenSSL

OpenSSL mencakup dukungan untuk algoritma *public key* yaitu RSA. Tidak seperti algoritma yang lain, algoritma RSA tidak membutuhkan parameter yang dihasilkan untuk menyederhanakan jumlah pekerjaan yang diperlukan pada saat menghasilkan kunci, autentikasi dan mengenkripsi komunkasi. OpenSSL memiliki *Command-line tool* yang memiliki tiga perintah untuk menghasilkan, memeriksa, memanipulasi, dan menggunakan kunci RSA.

OpenSSL menentukan nilai *p* dan *q*. Nilai *p* dan *q* merupakan bilangan prima besar yang dipilih secara acak. Dua angka dari *p* dan *q* dikalikan untuk mendapatkan *n* (modulus umum). Refernsi untuk kekuatan RSA ditunjukan pada panjang *bit* modulus umum. Nilai *e* dikenal sebagai eksponen publik. Menentukan nilai e harus *integer*, dipilih secara acak, relatif prima dan memiliki faktor 1 menggunakan (*p*–1)(*q*–1). Eksopnen publik biasanya angka yang memiliki jumlah kecil dan dalam prakteknya biasanya angka tersebut bernilai 3 atau 65,537 (disebut sebagai angka Fermat F4). Nilai *d* dihitung menggunakan *e*, *p*, dan *q*. Jadi *e* dan *n* adalah kunci publik, *d* dan *n* adalah kunci privat [NSO-02]. Untuk menghasilkan sepasang kunci RSA, OpenSSL menyediakan satu fungsi :

```
RSA *RSA_generate_key(int num, unsigned long e,
                 void (*callback)(int, int, void *), void
*cb_arg);
```

num

`int num`	Untuk menentukan panjang bit modulus umum (*n*)
`e`	Nilai umum eksponen. OpenSSL tidak menghasilkan nilai tersebut secara acak tapi dapat ditentukan oleh pengguna. Ditentukan dengan menggunakan salah satu konstanta, RSA_3 atau RSA_F4
`callback`	Pointer ke fungsi akan dipanggil selama proses *generate* untuk melaporkan status *generate*.
`Cb_arg`	Pointer ke *application-specific data*. OpenSSL tidak menggunakan nilai ini untuk apapun. Digunakan hanya saat melewati sebagai argumen untuk fungsi `callback` tertentu

Fungsi diatas di *execute* menggunakan perintah genrsa pada *command-line* OpenSSL digunakan untuk menghasilkan *private key* RSA yang baru. *Command-line* pada OpenSSL untuk menghasilkan *private key* RSA :

```
$ openssl genrsa -out rsaprivatekey.pem -passout pass:trousers -
des3 1024
```

Hasil *private key* RSA pada *command-line* OpenSSL melibatkan penemuan dua bilangan prima besar yang masing-masing berukuran sekitar setengah panjang kunci yang ditentukan. Kunci privat yang dihasilkan dapat di enkripsi menggunakan DES, DES3, atau IDEA. Komponen hasil kunci privat dapat dicetak dengan *output* standar untuk mengetahui proses yaitu `$openssl rsa -in rsaprivatekey.pem -noout -text`. Perintah rsa juga digunakan untuk mengasilkan kunci publik RSA dari kunci privat RSA. Untuk mendapatkan *public key* RSA dilakukan dengan mengekstrak *private key* RSA :

```
$openssl rsa -in rsaprivatekey.pem -passin pass:trousers –pubout -
out rsapublickey.pem
```

Komponen hasil kunci publik juga dapat dicetak dengan *output* standar untuk mengetahui hasil proses kunci publik yaitu `$openssl rsa -in rsapublickey.pem -pubin -noout –text`. Hasil *generate private key* dan *public key* disimpan dalam *file* dengan format .pem. PEM (*Privacy Enchanced Mail*) merupakan format standar pada OpenSSL. Menyimpan semua data *private key* dan *public key* dengan dikodekan dalam bentuk Base64 DER (*Distinguished Encoding Rules*) dan dikelilingi oleh *header* ASCII sehingga sangat cocok untuk model teks transfer antar sistem.

2.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah disiplin ilmu yang berkaitan dengan semua aspek produksi perangkat lunak dari tahap awal spesifikasi sistem sampai pemeliharaan sistem setelah sistem digunakan. Secara umum, perekayasa perangkat lunak mengadopsi pendekatan yang sistematis dan terorganisir untuk bekerja, hal tersebut merupakan cara yang paling efektif untuk menghasilkan perangkat lunak berkualitas tinggi. Akan tetapi, rekayasa adalah segala sesuatu tentang memilih metode yang paling sesuai untuk suatu set keadaan dan pendekatan yang lebih

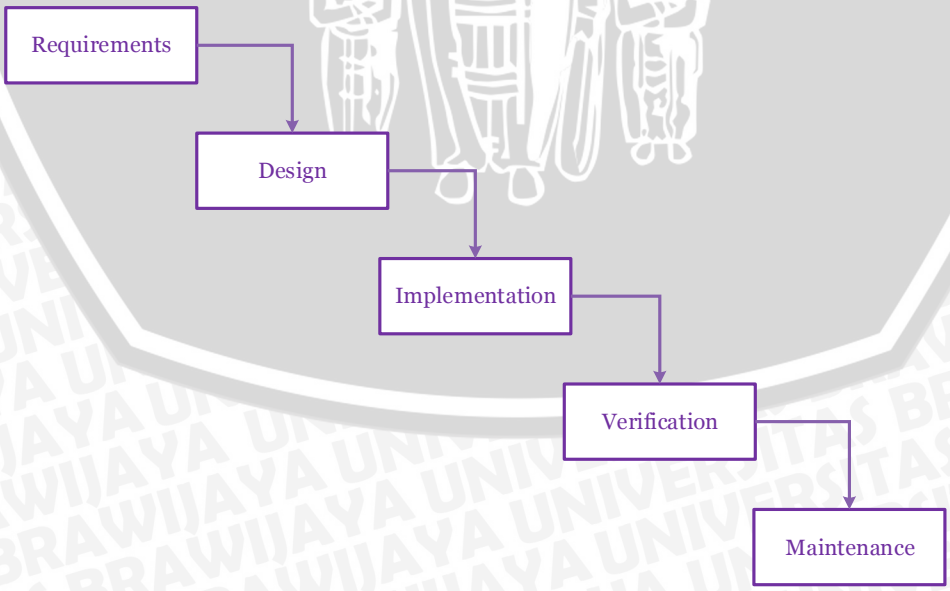
kreatif, informal terhadap pengembangan yang mungkin efektif pada beberapa keadaan [ISE-11].

2.4.1 Software Processs Model

Software process model merupakan representasi sederhana dari sebuah proses perangkat lunak. Pada rekayasa perangkat lunak, terdapat model-model yang dikembangkan untuk membantu proses pengembangan perangkat lunak. Model-model tersebut umumnya mengacu pada model proses pengembangan sistem yang disebut *System Development Life Cycle* (SDLC). Masing-masing model proses mewakili sebuah proses dari perspektif tertentu, dan menyediakan hanya sebagian informasi tentang proses itu [ISE-11].

- Model Waterfall

Desain sistem *e-voting* menggunakan model waterfall. Model *waterfall* merupakan model yang melakukan pendekatan secara sistematis dan sesuai urutan mulai dari kebutuhan sistem kemudian ke tahap desain, implementasi, *verivication*, dan *maintenance*. Dalam model *watefall*, setiap aktivitas pada tahapan harus diselesaikan sebelum menuju tahap pengembangan selanjutnya. *Review* dapat terjadi sebelum menuju ke tahap berikutnya yang memungkinkan untuk perubahan (yang mungkin melibatkan proses pengendalian perubahan formal). *Review* juga dapat digunakan untuk memastikan bahwa fase memang lengkap (bahwa proyek harus melewati untuk menuju tahap berikutnya) [STE-11].



Gambar 2.5 Model Waterfall
Sumber: [STE-11]

Tahap-tahap dari model waterfall sistem *e-voting* pada gambar 2.5 :

1. *Requirements*

   Merupakan tahapan untuk melakukan pengumpulan kebutuhan secara lengkap kemudian di analisis lalu di definisikan kebutuhan-kebutuhan proses produk yang harus dipenuhi oleh perangkat lunak yang akan dibangun.

2. *Design*

   Proses desain sistem digunakan untuk mengubah kebutuhan-kebutuhan tersebut menjadi desain sistem dan aplikasi meliputi desain konseptual, logikal dan fisikal. Desain perangkat lunak melibatkan mengidentifikasi dan menggambarkan dan menghubungkan abstraksi sistem perangkat lunak dasar.

3. *Implementation*

   Pada tahapan ini desain perangkat lunak yang telah dirancang, diwujudkan dengan menerjemahkan ke dalam kode-kode program menggunakan bahasa pemrograman. Dan melakukan pengujian terhadap unit-unit program yang telah di buat untuk memverifikasi bahwa setiap unit memenuhi spesifikasi.

4. *Verification*

   Tahapan dimana semua unit-unit program dijadikan satu kemudian melakukan pengujian sistem perangkat lunak secara keseluruhan. Setelah pengujian, sistem perangkat lunak akan dikirim ke pelanggan.

5. *Maintenance*

   Pengoperasian program. Dilakukan perawatan yang melibatkan pengoreksian kesalahan yang tidak ditemukan dalam tahap awal, meningkatkan kinerja sistem dan meningkatkan perbaikan sistem.

2.4.2 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk menunjukan bahwa program dapat melakukan apa yang menjadi tujuan perangkat lunak tersebut sebelum digunakan. Dalam melakukan pengujian perangkat lunak dapat dilakukan dengan mengeksekusi program dengan data percobaan. Kita dapat mengetahui dan memeriksa hasil uji coba untuk mendapatkan kesalahan, anomali, atau informasi tentang atribut nonfungsional dalam program. Tujuan pengujian mengarah pada pengujian validasi, di mana sistem diharapkan untuk melaksanakan dengan benar dalam satu set uji kasus yang mencerminkan penggunaan sistem yang diharapkan

dan mengarahkan pada pengujian terhadap kecacatan, di mana uji kasus dirancang untuk mengetahui kecacatan [ISE-11].

1. *Black-box Testing*

Setiap produk rekayasa perangkat lunak dapat diuji dalam pengujian *Black-box*. Pengujian *Black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan hal tersebut, pengujian *Black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *Black-box* berupaya menemukan kesalahan dalam kategori sebagai berikut :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisisalisasi dalam kesalahan terminasi

Proses pengujian *Black-box* dilakukan selama tahap akhir pengujian. Hal tersebut dikarenakan pengujian *Black-box* memperlihatkan struktur kontrol yang berfokus pada domain informasi. Keuntungan dalam penggunaan pengujian dengan metode *Black-box testing* yaitu :

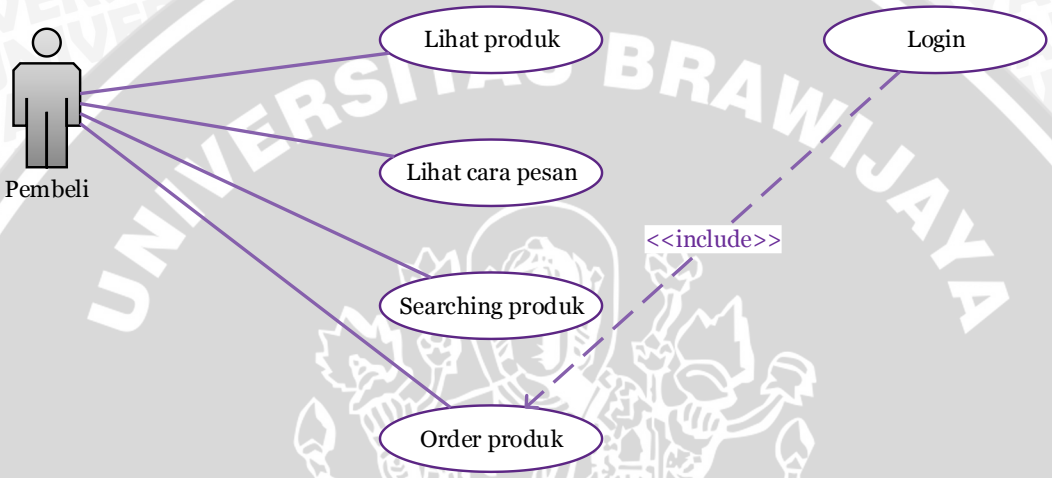
1. Dapat melakukan pengujian sesuai dengan spesifikasi yang telah ditentukan.
2. Dapat memaksimalkan pengujian dengan berfokus pada uji kasus yang berada dalam aplikasi.
3. Pengujian memastikan *output* yang didapat telah sesuai dengan tujuan dan berfokus pada masukan valid dan tidak valid.

2.5 Unified Modelling Language

Unified Modelling Language (UML) merupakan sebuah bahasa untuk visualisasi yang digunakan merancang dan mendokumentasikan sistem piranti lunak. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti lunak yang dapat berjalan pada piranti keras, sistem operasi dan jaringan, serta ditulis dalam bahasa pemrograman. UML mendefinisikan diagram-diagram yang memiliki fungsi dan informasi yang berbeda-beda [SDI-03].

2.5.1 Use case Diagram

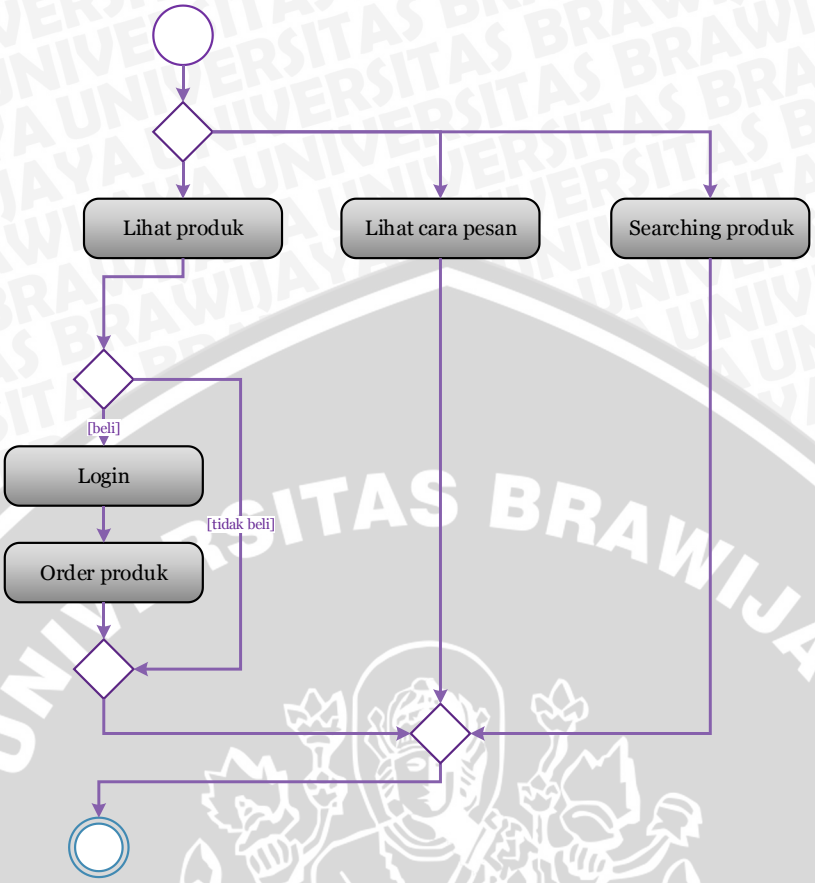
Use case merupakan diagram yang menggambarkan fungsionalitas dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, meng-*create* sebuah daftar belanja dan lain-lain. Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu seperti pada gambar 2.6 [SDI-03].



Gambar 2.6 Contoh *Use Case* Diagram

2.5.2 Activity Diagram

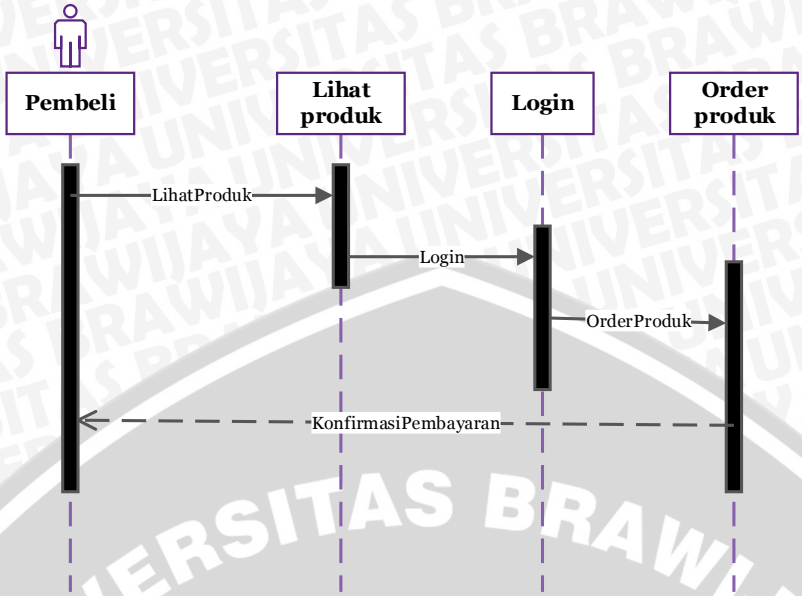
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. A*ctivity diagram* hanya menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. *Activity diagram* dapat dibagi menjadi beberapa object *swimline* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu seperti pada gambar 2.7 [SDI-03].



Gambar 2.7 Contoh *Activity Diagram*

2.5.3 Sequence diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* digunakan untuk menggambarkan rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah kasus untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan seperti pada gambar 2.8 [SDI-03].



Gambar 2.8 Contoh *Sequence Diagram*

2.6 Code Igniter Framework

CodeIgniter (CI) adalah sebuah *web application framework* yang bersifat *open source* untuk membangun aplikasi php dinamis. Tujuan utama pengembangan *CodeIgniter* adalah untuk membantu *developer* untuk mengerjakan aplikasi lebih cepat dan praktis daripada menulis semua *code* dari awal. *CodeIgniter* menyediakan berbagai macam *library* yang dapat mempermudah dalam pengerjaan dan pengembangan aplikasi [IDI-11]. *CodeIgniter* sangat ringan, terstruktur, dan mudah dipelajari. Selain itu *CodeIgniter* juga memiliki fitur-fitur lainnya yang sangat bermanfaat, antara lain :

- **Menggunakan pattern MVC**

  Dengan menggunakan pattern MVC ini, struktur kode yang dihasilkan menjadi lebih terstruktur dan memiliki standar yang jelas.

- **URL Friendly**

  URL yang dihasilkan sangat *url friendly*. Pada *CodeIgniter* diminimalisasi penggunaan `$_GET` dan diganti dengan URL.

- **Kemudahan**

  Kemudahan dalam mempelajari, membuat *library* dan *helper*, memodifikasi serta meng-integrasikan *library* dan *helper*.

- **Kecepatan**

  Berdasarkan hasil *benchmark, CodeIgniter* merupakan salah satu framework PHP tercepat saat ini.

- **Mudah dimodifikasi dan beradaptasi**

  Sangat mudah memodifikasi *behavior framework* ini. Tidak membutuhkan *server requirement* yang macam-macam serta mudah mengadopsi *library* lainnya.

- **Dokumentasi lengkap dan jelas**

  CodeIgniter telah menyediakan sebuah panduan yang lengkap didalamnya.

2.6.1 Model-View-Controller (MVC)

CodeIgniter menggunakan konsep MVC serta menyediakan banyak *library* dan *helper* untuk digunakan. MVC adalah sebuah *pattern* atau teknik pemrograman yang memisahkan bisnis *logic* (alur pikir), data *logic* (penyimpanan data) dan *presentation logic* (antarmuka aplikasi) atau secara sederhana yaitu memisahkan antara desain, data dan proses [IDI-11]. Komponen – komponen MVC antara lain:

1. Model

   Merepresentasikan struktur data. Biasanya *class model* akan berisi fungsi-fungsi untuk mengambil data, *insert data*, dan *update data* ke *database*.

2. View

   Merupakan informasi atau halaman yang ditampilkan ke pengguna. Sebuah *view* biasanya adalah sebuah *web page*, tapi di *CodeIgniter view* juga dapat berbentuk bagian-bagian halaman *web*, seperti *header* dan *footer*. Bahkan *view* juga dapat berupa halaman RSS.

3. Controller

   Berfungsi sebagai penghubung antara *Model, View* dan dengan sumber daya lain yang digunakan untuk memproses HTTP *request*. *Controller* juga biasanya berfungsi sebagai inti pemrosesan *logic* aplikasi.

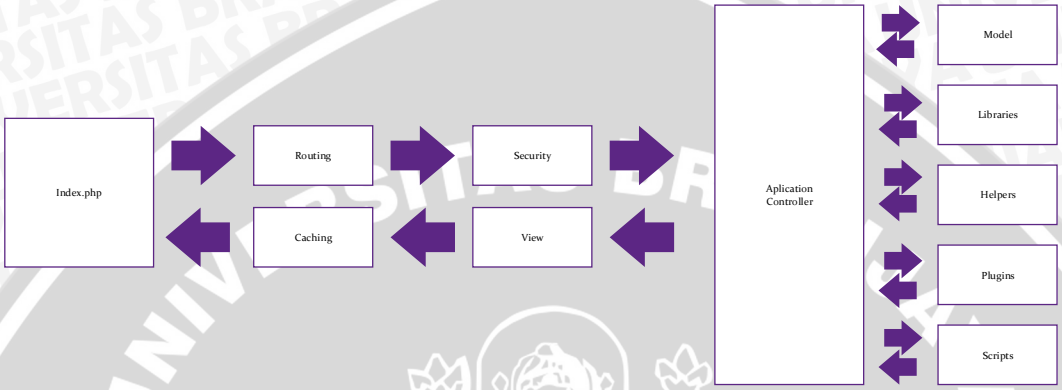
4. Libraries

   Merupakan macam-macam *class* yang masing-masing mempunyai fungsi khusus yang dapat digunakan untuk mengembangkan aplikasi. Contoh *library database, email*, validasi *form*, dan lain-lain.

5. *Helper*

Setiap *file helper* terdiri dari kumpulan fungsi. Contoh URL *Helper* yang berfungsi untuk membuat *link*, *form helper* untuk membuat elemen – elemen *form*. *Helper* tidak menggunakan *format Object Oriented*, sehingga dapat digunakan dimanapun, baik itu di *model*, *view*, *controller* dan *library* [IDI-11].

2.6.2 Alur Proses Data CodeIgniter



Gambar 2.9 Alur Proses Data pada *CodeIgniter*

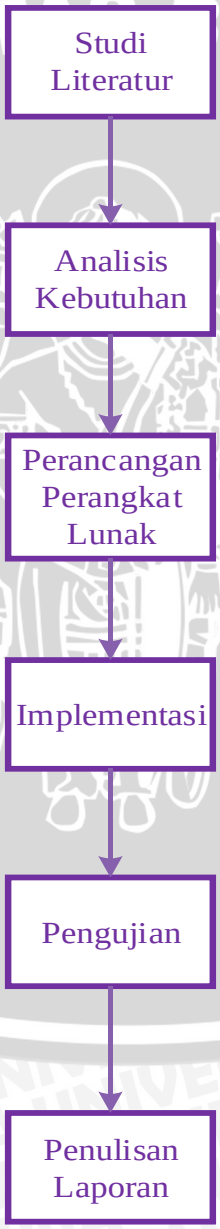
Alur proses data pada *CodeIgniter* pada gambar 2.9 menurut *CodeIgniter Framework* [CAN-11] :

1. Index.php berfungsi sebagai pengendali awal, menginisialisasi sumber daya utama yang dibutuhkan *CodeIgniter*.
2. *Router* memeriksa paket HTTP *request* untuk menentukan aksi apa yang harus dilakukan oleh sistem.
3. Jika *cache* tersedia, maka halaman langsung dikirim ke *browser*, eksekusi sistem dengan normal akan dilewati.
4. *Security*. Sebelum *Application Controller* dieksekusi, paket HTTP *request* dan semua data yang dikirimkan pengguna akan disaring terlebih dahulu oleh *Security Class*.
5. *Application Controller* menginisialisasi *model*, *library* utama, *helpers* dan semua sumberdaya yang dibutuhkan untuk setiap *request*.
6. Antarmuka aplikasi (*view*) yang sudah disiapkan dikirimkan ke *browser*. Jika *caching* diaktifkan, maka view akan disimpan sementara untuk *request* yang sama berikutnya.

BAB III
METODOLOGI PENELITIAN DAN PERANCANGAN

3.1 Metode Penelitian

Pada bab ini dijelaskan mengenai prosedur-prosedur dan kegiatan-kegiatan yang akan dilakukan dalam pengerjaan skripsi, yaitu studi literatur, analisa kebutuhan, perancangan perangkat lunak, implementasi, pengujian dan penulisan laporan.



Gambar 3.1 Perancangan

3.1.1 Studi Literatur

Studi literatur merupakan penelusuran literatur yang bertujuan dalam menyusun dasar teori yang digunakan untuk menunjang skripsi. Teori-teori pendukung tersebut diperoleh dari buku, website, ebook, dan jurnal. Data pada studi literatur didapatkan dari sumber yang terpercaya dan dapat dijadikan dasar teori dalam pembuatan aplikasi ini.

3.1.2 Analisis Kebutuhan

Analisa kebutuhan bertujuan untuk menganalisa dan mendapatkan semua kebutuhan dari sebuah perangkat lunak yang akan di bangun. Langkah-langkah pada analisis kebutuhan tersebut antara lain :

1. **Identifikasi Aktor**

Aktor merupakan pengguna sistem, dapat berupa orang atau sistem. Tahap identifikasi aktor dilakukan untuk mengetahui siapa saja yang berinteraksi dengan sistem. Dalam hal ini, aktor yang berinteraksi dengan sistem adalah *voter* dan *Administrator*.

2. **Identifikasi Kebutuhan**

Identifikasi kebutuhan ada 2, yaitu:

- Kebutuhan fungsional merupakan fungsionalitas atau layanan yang disediakan oleh sistem dan bagaimana sistem berinteraksi dengan input tertentu. Contohnya *user* melakukan proses pencarian data, dan sistem menampilkannya.
- Kebutuhan nonfungsional merupakan pendukung layanan yang disediakan oleh sistem. Seperti pendukung pengembangan proses yang dimiliki oleh sistem.

3. **Pembuatan *Use Case Diagram***

Sistem memiliki 2 aspek, yaitu siapa pengguna sistem dan apa kegiatan yang dapat dilakukan dalam sistem. Pengguna dari sistem ini adalah *voter* dan *Administrator*. Pengguna dapat melakukan kegiatan dengan mengakses sistem ini melalui *browser*. Kegiatan yang dapat dilakukan dalam sistem yaitu :

- Calon *voter* dapat melakukan pendaftaran.
- *Voter* dapat melakukan proses *voting*.

- Administrator dapat melakukan pengolahan data kandidat, pengolahan data *voter*, dan melihat hasil *voting*.
- Super Administrator dapat melakukan *generate* kunci RSA dan melihat data hasil *voting* per-*voter* pada *log* pemilihan dengan menggunakan kunci privat.

3.1.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem *e-voting* menggunakan model *waterfall*. Model *Waterfall* pada sistem *e-voting* memiliki tahapan :

1. *Reqiurements* :

   Requirements pada sistem *e-voting* didukung oleh perangkat *voting* yang digunakan, siapa saja pengguna *e-voting*, langkah-langkah untuk melakukan *voting*, menu dan transfer data pada perangkat serta antarmuka *standart* yang digunakan.
2. *Design* dan *Implementation* :

   Mendesain sistem *e-voting* pada perangkat yang digunakan meliputi komponen *software* dan proses keamanan transfer data pada *server*.
3. *Verification*

   Melakukan pengujian terhadap sistem *e-voting* yang telah dibangun.
4. *Maintenance*

   Merupakan tahapan perbaikan *bug* pada sistem yang merupakan bagian dari rencana yang dilakukan untuk *e-voting*. Hal tersebut akan mencakup *e-voting* pada pemilih baru atau pemilih dan perangkat *e-voting* versi baru dari perangkat lunak *browser web* dan pemilih baru atau perangkat pemilih dengan sistem operasi baru.

Selanjutnya perancangan perangkat lunak dilakukan setelah mendapatkan semua kebutuhan dari proses analisis kebutuhan. Perancangan perangkat lunak terdiri dari:

1. Perancangan kriptografi kunci-publik pada sistem.
2. Perancangan *use case* untuk aktor-aktor dan kegiatan apa saja yang dilakukan dalam sistem.
3. Perancangan diagram aktivitas untuk menggambarkan alur kegiatan sistem.
4. Perancangan diagram interaksi untuk menggambarkan antar elemen dalam sistem.

5. Perancangan tampilan antar muka aplikasi yaitu :
   - Menu aplikasi Calon *Voter*.
   - Menu aplikasi User *Voter*.
   - Menu aplikasi User Administrator.
   - Menu aplikasi User Super Administrator.

3.1.4 Implementasi Perangkat Lunak

Setelah melakukan tahap perancangan, tahap berikutnya adalah implementasi (pembuatan aplikasi). Dalam implementasinya, aplikasi ini menggunakan bahasa pemrograman PHP, *library* OpenSSL dan *database* MySQL.

3.1.5 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk memastikan aplikasi yang di bangun sesuai dengan yang diinginkan. Hal tersebut bertujuan agar tidak terjadi kesalahan (*error*) pada saat aplikasi digunakan oleh pengguna. Pengujian yang dilakukan pada perangkat lunak ini yaitu pengujian validasi dengan metode *Black-box*.

3.1.6 Pengambilan Kesimpulan

Pembuatan kesimpulan dilakukan setelah selesai melakukan tahapan analisis kebutuhan, perancangan, implementasi dan pengujian sistem. Kesimpulan diambil dari hasil pengujian dan analisis dari sistem yang dibangun. Tahap akhir dari penulisan yaitu saran yang dimaksudkan untuk memberikan pertimbangan untuk tahap pengembangan selanjutnya.

3.2 Perancangan

Perancangan perangkat dilakukan setelah proses pengumpulan semua kebutuhan dari analisa kebutuhan. Proses perancangan terdiri dari analisa kebutuhan perangkat lunak/keras dan perancangan perancangan perangkat lunak/keras.

3.2.1 Analisa Kebutuhan Perangkat Lunak/Keras

Analisa kebutuhan bertujuan untuk mendapatkan semua daftar kebutuhan yang dibutuhkan oleh sistem yang akan dibangun. Proses analisa kebutuhan dilakukan dengan mengidentifikasi kebutuhan sistem dan siapa saja yang terlibat dalam sistem tersebut. Proses analisa kebutuhan ini dimulai dengan identifikasi

aktor-aktor yang terlibat dalam sistem, penjelasan pada daftar kebutuhan lalu memodelkannya ke dalam diagram *use case*.

1. Identifikasi Aktor

Pada tabel 3.1 adalah tahap identifikasi terhadap aktor-aktor yang akan berinteraksi dengan sistem *e-voting*.

Tabel 3.1 Identifikasi Aktor

Aktor	Deskripsi Aktor
Super Administrator	Super Administrator merupakan aktor yang melakukan proses *generate* kunci RSA dan proses dekripsi (menggunakan kunci privat) data hasil *voting*
Administrator	Administrator merupakan aktor yang mengoperasikan dan memelihara sistem *e-voting*
Voter	*Voter* merupakan aktor pengguna sistem *e-voting* melalui proses autentikasi terlebih dahulu
Calon *Voter*	Calon *voter* merupakan aktor pengguna sistem *e-voting* yang belum melakukan proses autentikasi dan otorisasi

2. Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional. Spesifikasi kebutuhan fungsional ditunjukan pada tabel 3.2 dan kebutuhan non-fungsional pada tabel 3.3.

Tabel 3.2 Daftar Kebutuhan Fungsional

ID	Kebutuhan	Aktor
SRS_001_01	Sistem dapat melakukan pendaftaran *voter* baru	Calon *Voter*
SRS_002_01	Sistem dapat melakukan vote sekarang	*Voter*
SRS_003_01	Sistem dapat melihat data kandidat	Administrator
SRS_003_02	Sistem dapat menambah data kandidat	Administrator
SRS_003_03	Sistem dapat menghapus data kandidat	Administrator
SRS_003_04	Sistem dapat melihat data *voter*	Administrator

SRS_003_05	Sistem dapat menambah data *voter*	Administrator
SRS_003_06	Sistem dapat menghapus data *voter* yang belum melakukan *voting*	Administrator
SRS_003_07	Sistem dapat menyimpan dan menampilkan data hasil *voting*	Administrator
SRS_003_08	Sistem dapat menampilkan data *log* pemilihan	Administrator
SRS_004_01	Sistem dapat melakukan *generate* kunci RSA	Super Administrator
SRS_004_02	Sistem dapat melakukan proses dekripsi data *log* pemilihan	Super Administrator

Tabel 3.3 Daftar Kebutuhan Non-Fungsional

Parameter	Deskripsi Kebutuhan
Security	Perangkat lunak harus dapat menyimpan data yang terdeteksi dengan benar, adanya hak akses yang membatasi hak akses masing-masing *user*

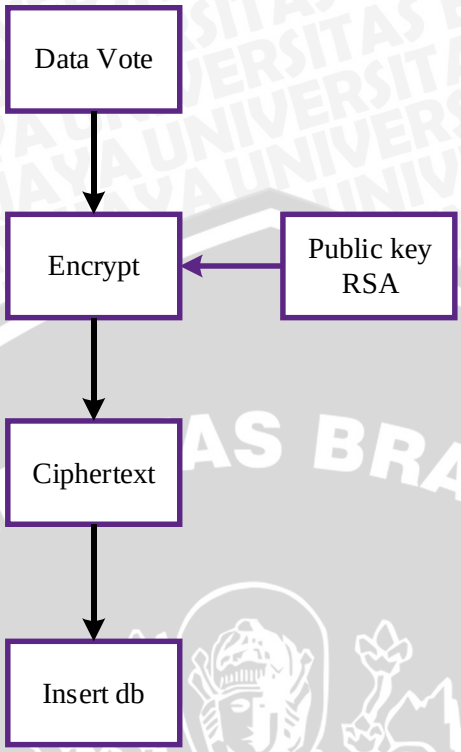
3.2.2 Perancangan Perangkat Lunak

Tahapan ini dilakukan setelah semua proses penentuan kebutuhan telah di dapatkan dari proses analisis. Kemudian dimodelkan dalam bentuk diagram *use case*, diagram aktivitas, diagram interaksi, perancangan basis data dan perancangan antarmuka.

3.2.2.1 Perancangan Kriptografi Kunci-Publik

Perancangan kriptografi kunci-publik menggunakan algoritma RSA pada proses pengamanan data hasil *voting* yaitu enkripsi dengan kunci publik dan dekripsi dengan kunci privat.

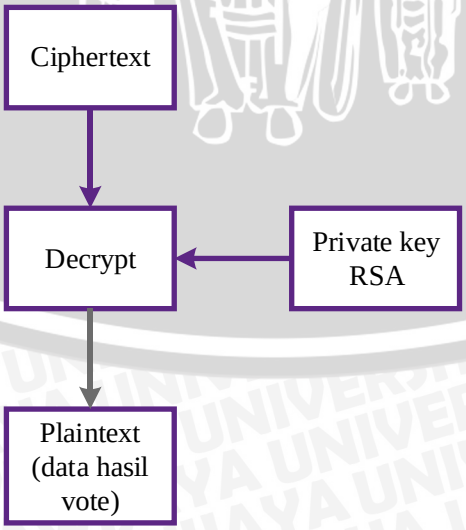
- Perancangan enkripsi data *e-voting* dengan kunci publik RSA



Gambar 3.2 Perancangan proses enkripsi data hasil *voting*

Pada gambar 3.2 menunjukan perancangan proses enkripsi data hasil *voting*. Data hasil *voting* yang sebelum disimpan oleh sistem dienkripsi menggunakan kunci publik RSA. Data *voting* menjadi *ciphertext* lalu disimpan di *database*.

- Perancangan dekripsi data *e-voting* dengan kunci privat RSA

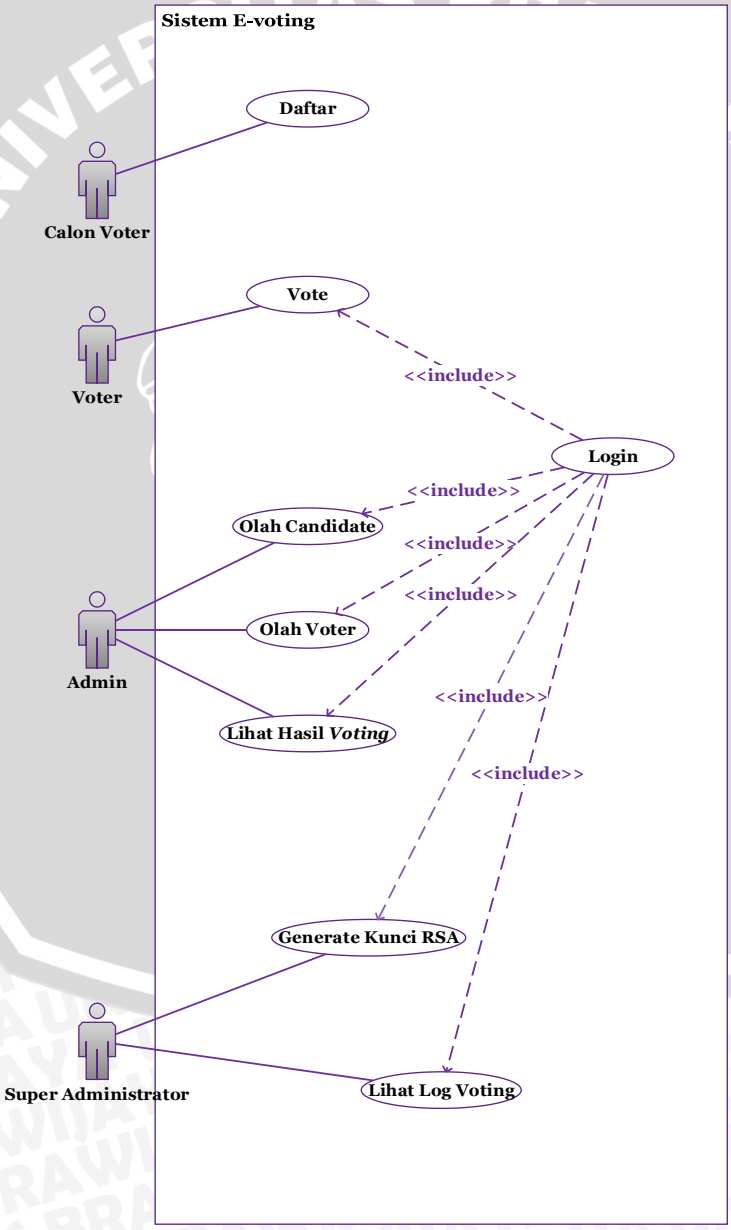


Gambar 3.3 Perancangan proses dekripsi data hasil *voting*

Pada gambar 3.3 menunjukan perancangan proses dekripsi data hasil *voting*. Data hasil *voting* yang sudah disimpan di *database* dalam *ciphertext*, di dekripsi menggunakan kunci privat RSA dan menjadi *plaintext* data hasil *voting*.

3.2.2.2 Perancangan Use Case

Perancangan *use case* dari sistem *e-voting* akan diberi penjelasan dengan nama *use case*, aktor yang berhubungan dengan *use case*, tujuan dari *use case*, deskripsi tentang *use case*, kondisi awal yang harus dipenuhi, kondisi akhir yang diharapkan setelah berjalannya fungsional *use case* dan tanggapan sistem akan satu aksi yang dilakukan oleh aktor.



Gambar 3.4 *Use case* Sistem *E-voting*

Penjelasan mengenai masing-masing *use case* pada gambar 3.4 akan dijabarkan dalam tabel-tabel dibawah ini:

1. *Use case* calon *voter* pada tabel 3.4.

Tabel 3.4 Tabel *Use Case* Calon *Voter*

Skenario Kasus Pada Sistem	
Nama	Calon *voter*
Tujuan	Calon *voter* dapat melakukan pendaftaran
Deskripsi	*Use case* ini menjelaskan bagaimana calon *voter* melakukan pendaftaran
Aktor	Calon *voter*
Skenario Utama	
Kondisi awal	Calon *voter* telah memasuki halaman *contact* atau menekan tautan daftar pada halaman utama
Aksi Aktor	**Reaksi Sistem**
• Calon *voter* memasuki halaman *contact* • Calon *voter* memasukan data pendaftaran dan menekan tombol *send*	Sistem menampilkan halaman *contact* Sistem melakukan *input* data pendaftaran yang dimasukan oleh calon *voter*
Kondisi Akhir	*Voter* kembali ke halaman *contact*

2. *Use case* vote sekarang untuk *voter* pada tabel 3.5.

Tabel 3.5 Tabel *Use Case* Vote Sekarang untuk *Voter*

Skenario Kasus Pada Sistem	
Nama	Vote sekarang
Tujuan	*Voter* dapat melakukan vote
Deskripsi	*Use case* ini menjelaskan bagaimana *voter* melakukan vote
Aktor	*Voter*
Skenario Utama	

Kondisi awal	*Voter* telah melalui proses autentikasi dan otorisasi pada sistem dan telah masuk ke halaman *voter*
Aksi Aktor	**Reaksi Sistem**
• *Voter* menekan tautan vote sekarang	Sistem menampilkan halaman vote
• *Voter* melihat daftar kandidat	Sistem menampilkan data kandidat
• *Voter* memilih *radio* pada kandidat yang dipilih dan menekan tombol *vote*	Sistem melakukan *input* data *voting* yang dimasukan oleh *voter* ke dalam sistem
Kondisi Akhir	*Voter* kembali ke halaman *vote* sekarang dan tidak bisa melakukan *voting* lagi

3. *Use case* olah data kandidat pada tabel 3.6.

Tabel 3.6 Tabel *Use Case* Olah Data Kandidat

Skenario Kasus Pada Sistem	
Nama	Olah data kandidat
Tujuan	Administrator dapat mengolah dan menampilkan data kandidat
Deskripsi	*Use* case ini menjelaskan bagaimana Administrator dapat mengolah dan menampilkan data kandidat
Aktor	Administrator
Skenario Utama	
Kondisi awal	Administrator telah melalui proses autentikasi dan otorisasi pada sistem dan telah memasuki halaman menu lalu menekan tombol kandidat
Aksi Aktor	**Reaksi Sistem**
• Administrator melihat data kandidat	Sistem menampilkan data kandidat
• Administrator menambah data kandidat	Sistem menampilkan *form input* data kandidat dan menyimpannya

• Administrator menghapus data kandidat	Sistem melakukan penghapusan data kandidat
Kondisi Akhir	Sistem menampilkan data kandidat

4. *Use case* olah data *voter* pada tabel 3.7.

Tabel 3.7 Tabel *Use Case* Olah Data *Voter*

Skenario Kasus Pada Sistem	
Nama	Olah data *voter*
Tujuan	Administrator dapat mengolah dan menampilkan data *voter*
Deskripsi	*Use case* ini menjelaskan bagaimana Administrator dapat mengolah dan menampilkan data *voter*
Aktor	Administrator
Skenario Utama	
Kondisi awal	Administrator telah melalui proses autentikasi dan otorisasi pada sistem dan telah memasuki halaman menu lalu menekan tombol pemilih
Aksi Aktor	**Reaksi Sistem**
• Administrator melihat data *voter* • Administrator menambah data *voter* • Administrator menghapus data *voter* yang belum melakukan proses *voting*	Sistem menampilkan data *voter* Sistem menampilkan *form input* data *voter* dan menyimpannya pada sistem Sistem melakukan penghapusan data *voter*
Kondisi Akhir	Sistem menampilkan data *voter*

5. *Use case* lihat hasil *voting* pada tabel 3.8.

Tabel 3.8 Tabel *Use Case* Lihat Hasil *Voting*

Skenario Kasus Pada Sistem	
Nama	Lihat hasil *voting*
Tujuan	Administrator dapat melihat data hasil *voting*
Deskripsi	*Use case* ini menjelaskan bagaimana Administrator dapat melihat data hasil *voting*
Aktor	Administrator
Skenario Utama	
Kondisi awal	Administrator telah melalui proses autentikasi dan otorisasi pada sistem dan telah memasuki halaman menu lalu menekan tombol hasil pemilihan
Aksi Aktor	**Reaksi Sistem**
• Administrator melihat data hasil *voting*	Sistem menampilkan data hasil *voting*
Kondisi Akhir	Sistem menampilkan halaman data hasil *voting*

6. *Use case generate* kunci RSA pada tabel 3.9.

Tabel 3.9 Tabel *Use Case Generate* Kunci

Skenario Kasus Pada Sistem	
Nama	*Generate* kunci RSA
Tujuan	Super Administrator melakukan *generate* kunci RSA
Deskripsi	*Use case* ini menjelaskan bagaimana Super Administrator melakukan *generate* kunci RSA
Aktor	Super Administrator
Skenario Utama	
Kondisi awal	Super Administrator telah melalui proses autentikasi dan otorisasi pada sistem dan memasuki halaman proses *generate* kunci RSA

Aksi Aktor	Reaksi Sistem
• Super Administrator melakukan proses *generate* kunci RSA	Sistem menampilkan hasil *generate* pasangan kunci RSA
Kondisi Akhir	Sistem menampilkan halaman *generate*

7. *Use case* lihat *log* pemilihan pada tabel 3.10.

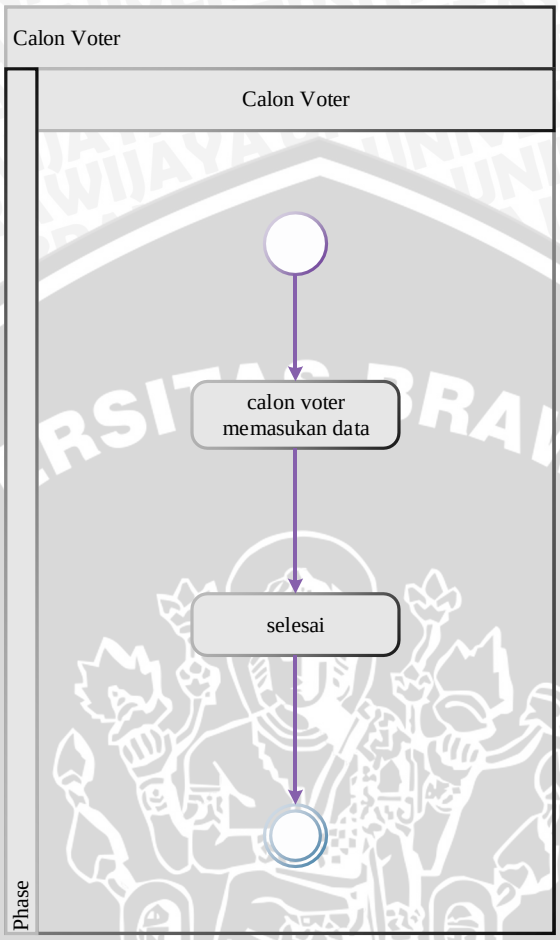
Tabel 3.10 Tabel *Use case* Lihat *Log* Pemilihan

Skenario Kasus Pada Sistem	
Nama	Lihat *log* pemilihan
Tujuan	Super Administrator dapat melihat data hasil *voting* dari *voter*
Deskripsi	*Use case* ini menjelaskan bagaimana Super Administrator dapat melihat data hasil *voting* dari *voter*
Aktor	Super Administrator
Skenario Utama	
Kondisi awal	Super Administrator telah melalui proses autentikasi dan otorisasi pada sistem dan telah memasuki halaman menu lalu menekan tombol *log* pemilihan
Aksi Aktor	**Reaksi Sistem**
• Super Administrator memiliki dan melakukan *upload private key* • Super Administrator melihat data hasil *voting* dari *voter* dan melakukan *sorting*	Sistem menampilkan halaman *upload private key* Sistem menampilkan halaman data hasil *voting* dari *voter*
Kondisi Akhir	Sistem menampilkan halaman data hasil *voting* masing-masing *voter*

3.2.2.3 Perancangan Aktivitas

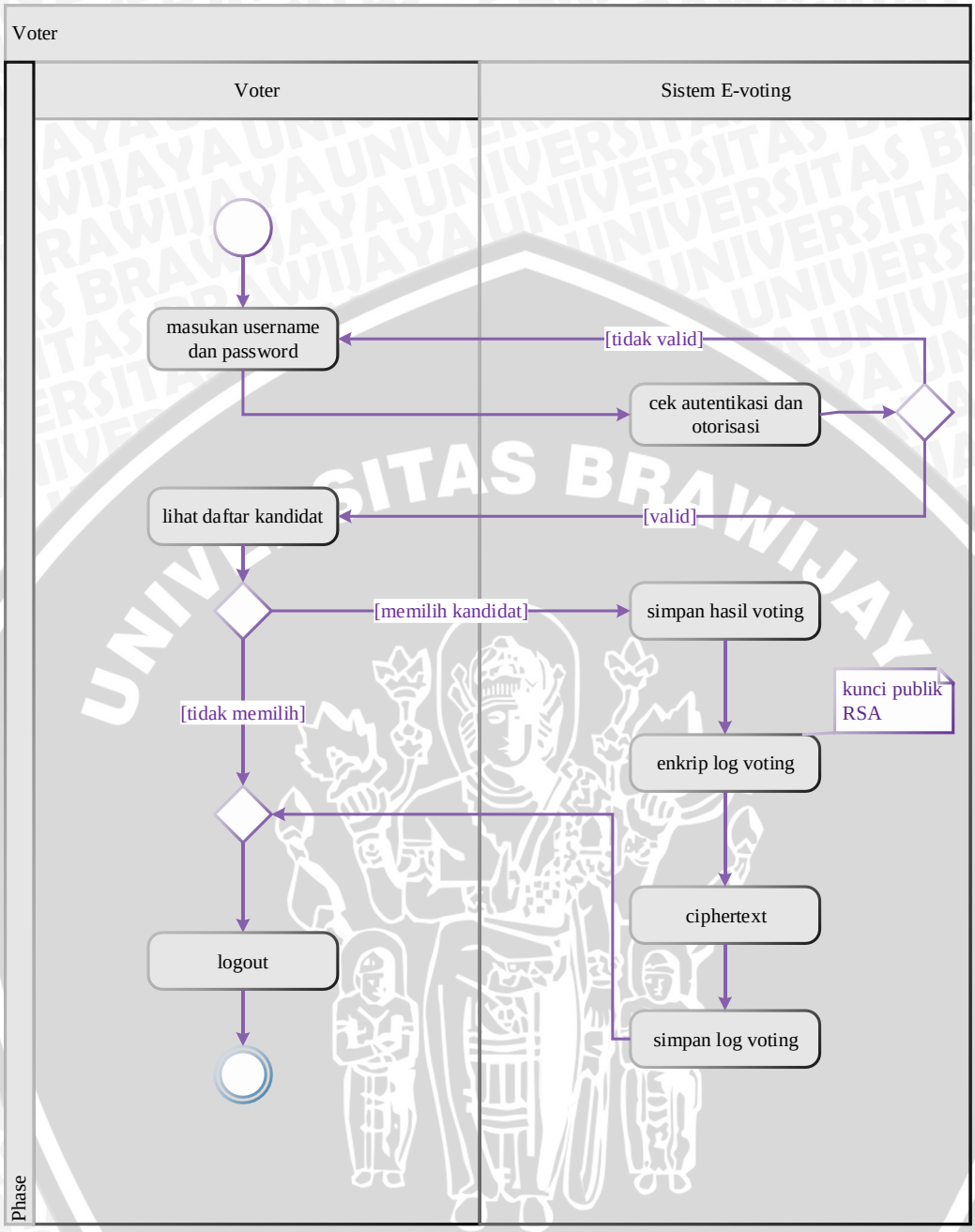
Activity diagram digunakan untuk menggambarkan dan menjelaskan alur proses kegiatan sistem. *Activity diagram* dikelompokkan menjadi empat bagian,

yaitu *activity diagram* untuk pendaftaran calon *voter*, proses vote sekarang, olah data untuk Administrator dan Super Administrator.



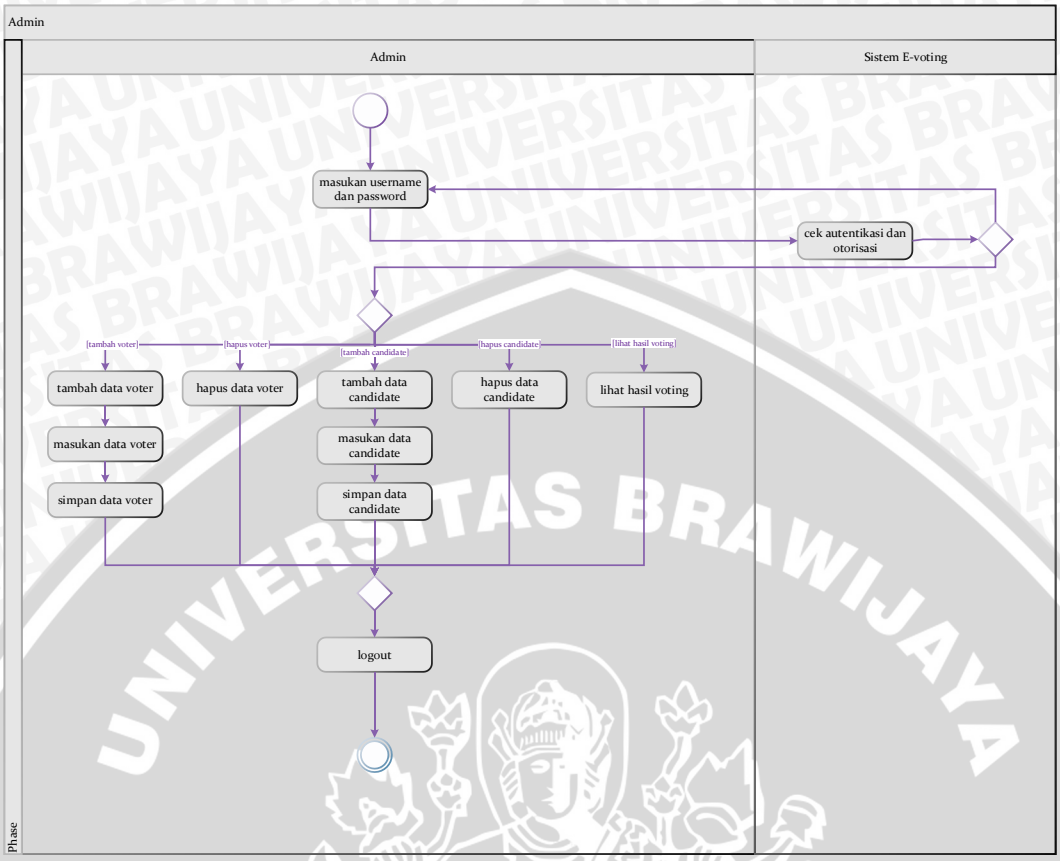
Gambar 3.5 Diagram Aktivitas Pendaftaran Calon *Voter*

Pada gambar 3.5 menunjukan diagram aktivitas pendaftaran calon *voter*. Diagram aktivitas ini mencakup *use case* calon *voter*. Calon *voter* harus melakukan pendaftaran terlebih dahulu apabila belum terdaftar pada saat akan melakukan proses autentikasi dan otorisasi untuk *vote* sekarang. Setelah melakukan pendaftaran, calon *voter* yang sudah menjadi *voter* melakukan autentikasi dan otorisasi untuk *vote* sekarang.



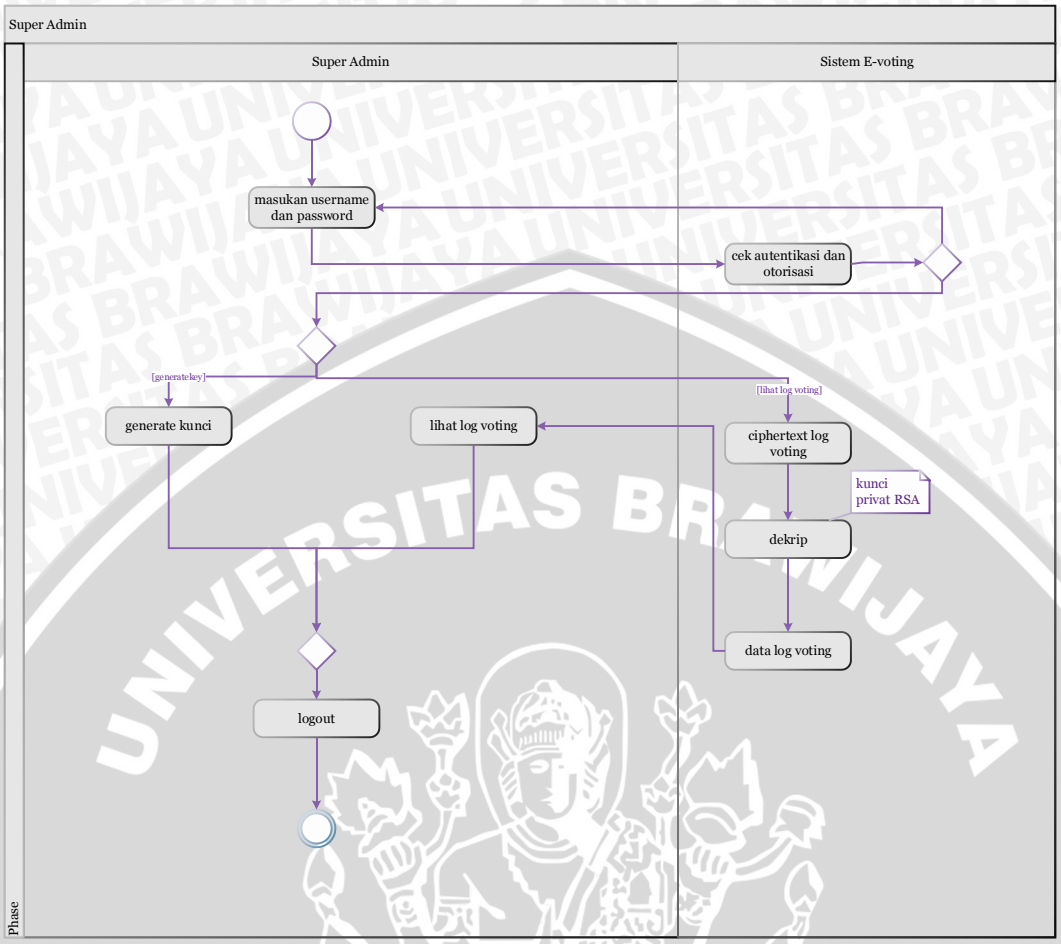
Gambar 3.6 Diagram Aktivitas *Vote* Sekarang

Pada gambar 3.6 menunjukan diagram aktivitas *vote* sekarang untuk *voter*. Sebelum melakukan *vote* sekarang, *voter* harus melakukan autentikasi dan otorisasi. Setelah melakukan autentikasi dan otorisasi, *voter* dapat melihat daftar kandidat yang akan dipilih. Setelah *voter* memilih kandidat, sistem akan memproses data hasil *voting* oleh *voter* dengan enkripsi menggunakan kunci publik RSA dan menyimpannya di *database*.



Gambar 3.7 Diagram Aktivitas Olah Data untuk Administrator

Gambar 3.7 menunjukan diagram aktivitas olah data untuk Administrator. Sebelum melakukan olah data, Administrator harus melakukan autentikasi dan otorisasi. Setelah melakukan autentikasi dan otorisasi, Administrator dapat melihat menu olah data. Untuk olah data Administrator dapat melakukan olah data kandidat, olah data *voter*, dan melihat hasil *voting*. Untuk olah data kandidat, Administrator dapat melakukan lihat data kandidat, tambah data kandidat dan hapus data kandidat. Untuk olah data *voter*, Administrator dapat melakukan lihat data *voter*, tambah data *voter*, dan hapus data *voter*. Untuk melihat hasil *voting*, adminitrator dapat melakukan lihat hasil *voting*.

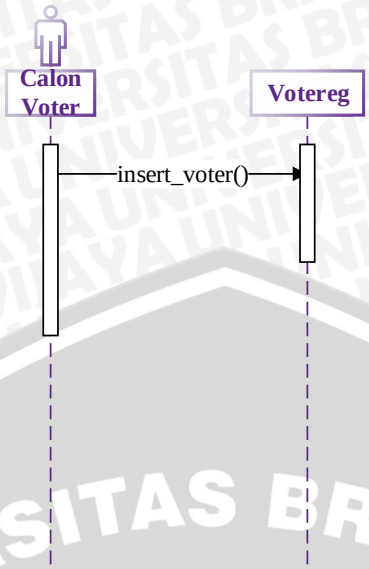


Gambar 3. 8 Diagram Aktivitas Super Administrator

Gambar 3.8 menunjukan diagram aktivitas Super Administrator. Super Administrator dapat melakukan *generate* kunci RSA dan untuk melihat *log* pemilihan, Super Administrator dapat melakukan dengan memiliki dan meng-*upload* kunci privat RSA untuk mendekripsi serta mensorting data hasil *voting* dari *voter*.

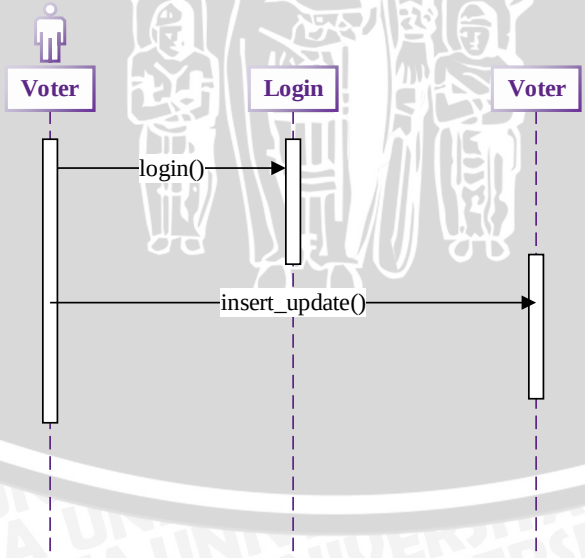
3.2.2.4 Perancangan Interaksi

Sequence diagram digunakan untuk menjelaskan interaksi antar objek pada sistem. *Sequence diagram* biasa digunakan untuk menggambarkan skenario dari sebuah kejadian untuk mendapatkan keluaran tertentu. Skenario dari *sequence diagram* ini dibagi menjadi empat bagian, yaitu *sequence diagram* untuk pendaftaran calon *voter*, proses *vote* sekarang, olah data untuk adminitrator dan super administrator.



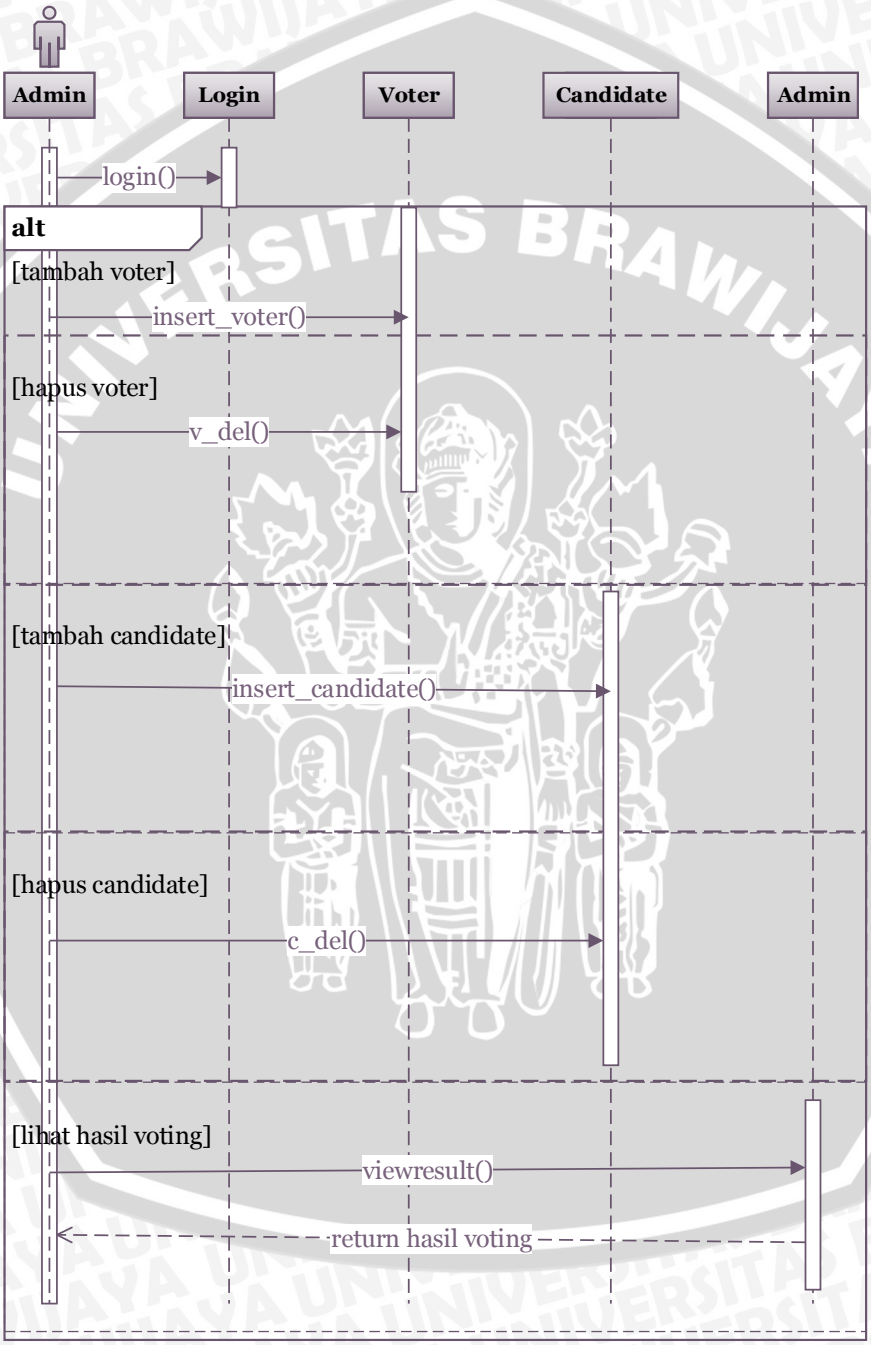
Gambar 3.9 Diagram Interaksi Pendaftaran Calon *Voter*

Gambar 3.9 menunjukan diagram interaksi pendaftaran calon *voter*. Calon *voter* harus melakukan pendaftaran terlebih dahulu apabila belum terdaftar untuk melakukan proses *voting*. Pada gambar 3.10 menunjukan diagram interaksi proses *vote* sekarang. *Voter* melakukan proses *voting* dengan melihat daftar kandidat dan menekan tombol *vote*.



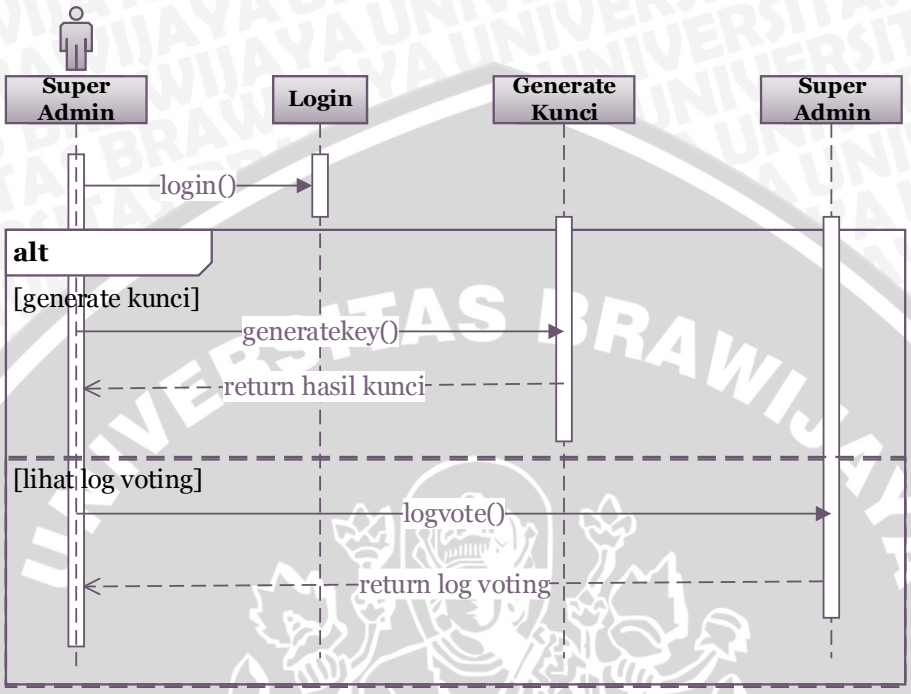
Gambar 3.10 Diagram Interaksi Proses *Vote* Sekarang

Pada gambar 3.11 menunjukan diagram interaksi olah data untuk Administrator. Administrator dapat memilih menu untuk melakukan olah data. Administrator dapat melihat data *voter*, menambah data *voter*, dan menghapus data *voter*. Administrator dapat melihat data kandidat, menambah data kandidat dan menghapus data kandidat serta melihat hasil *voting*.



Gambar 3.11 Diagram Interaksi Olah Data untuk Administrator

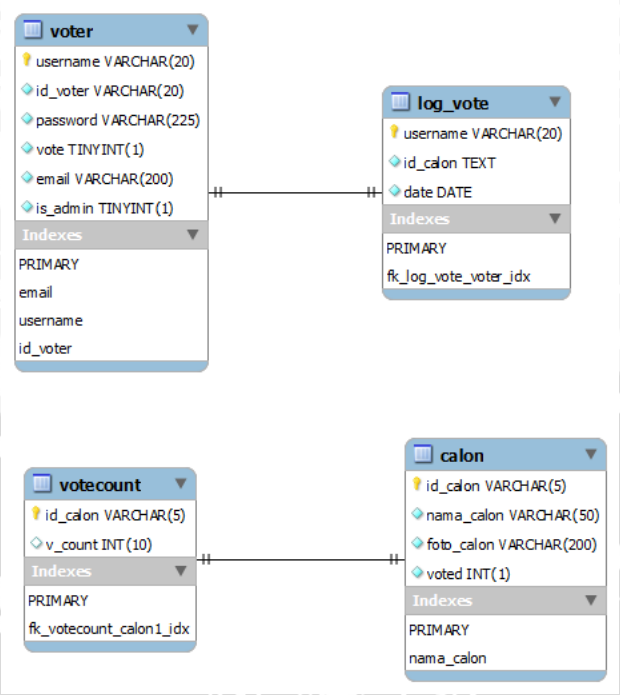
Gambar 3.12 menunjukan diagram interaksi Super Administrator. Super Administrator dapat melakukan *generate* kunci RSA dan *log* pemilihan. *Log* pemilihan meliputi data hasil *voting* dari *voter* dan *sorting* data hasil *voting*.



Gambar 3.12 Diagram Interaksi Super Administrator

3.2.2.5 Perancangan Basis Data

Perancangan basis data merupakan perancangan manajemen data yang akan digunakan sistem. Perancangan basis data pada sistem ini digambarkan dengan *entity relationship diagram*. *Entity relationship diagram* sistem ditunjukan pada gambar 3.13.



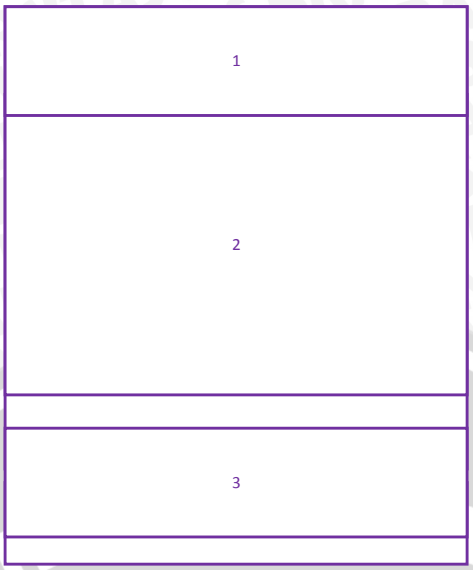
Gambar 3.13 Rancangan Database Sistem

3.2.2.6 Perancangan Antarmuka

Pada bagian ini akan dijelaskan tentang perancangan antarmuka sistem perangkat lunak untuk *e-voting* dan keamanan data *voting*. Perancangan antarmuka mewakili keadaan sebenarnya dari sistem yang akan dibangun. Antarmuka perangkat lunak bertujuan digunakan oleh pengguna untuk berinteraksi dengan sistem yang dibangun.

1. Perancangan Halaman Utama

Perancangan halaman utama merupakan perancangan antarmuka pengguna yang berfungsi sebagai akses *log in* untuk *voter*, *Administrator* dengan hak akses yang telah ditentukan dan menampilkan gambar kandidat.



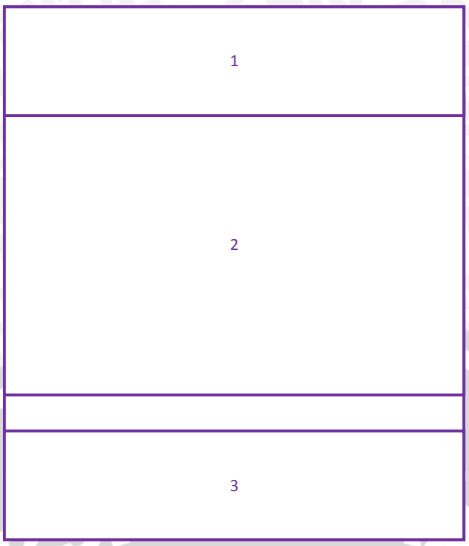
Gambar 3.14 Perancangan Antarmuka Halaman Utama

Halaman utama menampilkan sekilas tentang gambar para kandidat dan menu *log in*. Gambar 3.14 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi gambar para kandidat dan tempat untuk proses *login* untuk pengguna yang memiliki akses sebagai *Administrator* dan *voter*.
3. Footer berisi nama dari sistem.

2. Perancangan Halaman Pendaftaran Calon *Voter*

Perancangan halaman pendaftaran calon *voter* juga merupakan perancangan antarmuka pengguna dalam melakukan pendaftaran identitas sebelum melakukan proses *voting*.



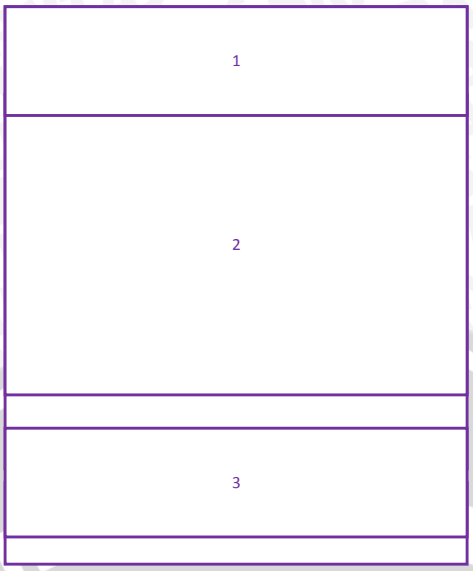
Gambar 3.15 Perancangan Antarmuka Halaman Pendaftaran Calon *Voter*

Halaman pendaftaran calon *voter* menampilkan *form* pendaftaran. Gambar 3.15 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi *form* pendaftaran untuk calon *voter*.
3. Footer berisi nama dan tahun dari sistem.

3. Perancangan Halaman Vote Sekarang untuk *Voter*

Perancangan halaman vote sekarang untuk *voter* merupakan perancangan antarmuka pengguna untuk melakukan proses *voting* setelah *login* pada halaman utama.



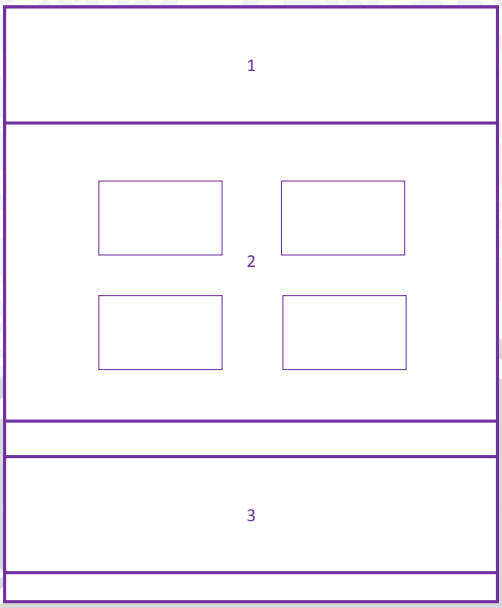
Gambar 3.16 Perancangan Antarmuka Halaman Vote Sekarang

Halaman *vote* sekarang menampilkan daftar kandidat yang akan dipilih. Gambar 3.16 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi daftar kandidat yang akan dipilih dan ikon *vote*.
3. Footer berisi nama dari sistem.

4. Perancangan Halaman Olah Data untuk Administrator

Perancangan halaman olah data untuk Administrator merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Administrator. Administrator dapat melakukan olah data *voter*, kandidat, dan *view result.*



Gambar 3.17 Perancangan Antarmuka Halaman Olah Data

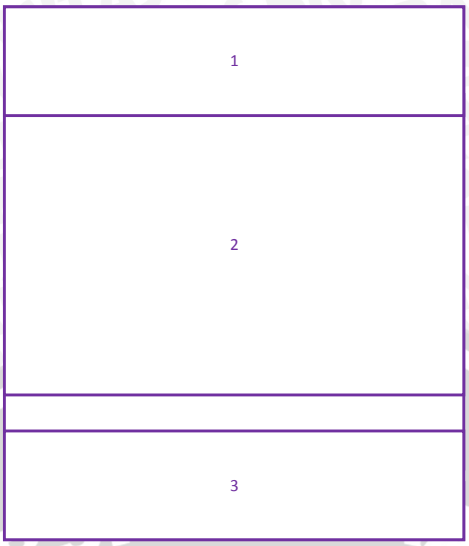
Halaman olah data untuk Administrator menampilkan menu-menu yang digunakan oleh admin. Gambar 3.17 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna
2. Konten berisi 4 menu yang digunakan admin untuk olah data
3. Footer berisi nama dari sistem

Berikut perancangan halaman olah data Administrator, yaitu olah data *voter*, olah data kandidat dan *view result* :

- **Perancangan Halaman Olah Data Kandidat untuk Administrator**

Perancangan halaman olah data kandidat merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Administrator. Pada halaman olah data kandidat ini Administrator memiliki fasilitas melihat data kandidat dan melakukan penambahan data kandidat baru.



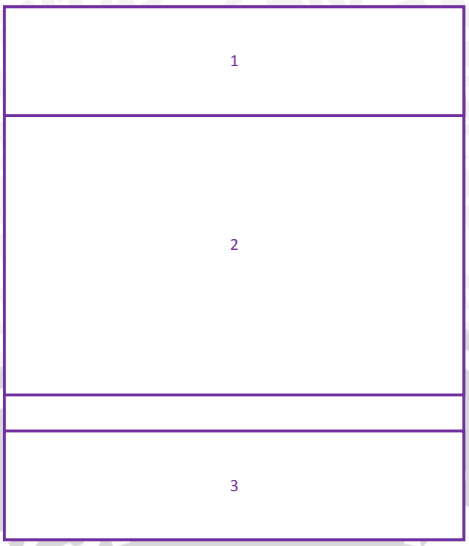
Gambar 3.18 Perancangan Antarmuka Halaman Olah Data Kandidat

Halaman olah data kandidat untuk Administrator menampilkan data kandidat yang terdaftar. Gambar 3.18 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi tabel data kandidat dan ikon tambah kandidat. Ikon tambah kandidat akan menampilkan konten yang berisi *form input* data kandidat baru.
3. Footer berisi nama dari sistem.

- **Perancangan Halaman Olah Data *Voter* untuk Administrator**

Perancangan halaman olah data *voter* merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Administrator. Pada halaman olah data kandidat ini Administrator memiliki fasilitas melihat data *voter* dan melakukan penambahan data calon *voter*.



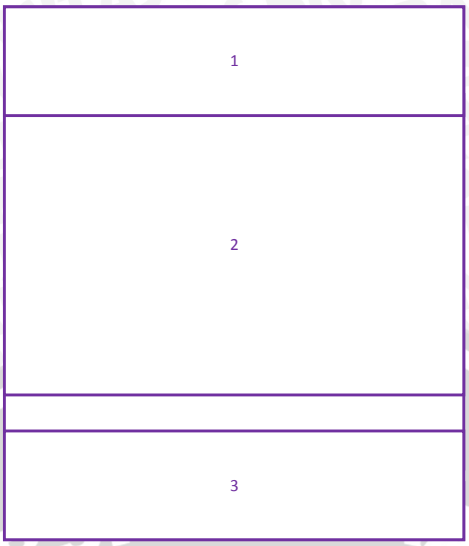
Gambar 3.19 Perancangan Antarmuka Halaman Olah Data *Voter*

Halaman olah *voter* untuk Administrator menampilkan data *voter* yang terdaftar. Gambar 3.19 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi tabel data *voter* dan ikon *add voter*. Ikon *add voter* akan menampilkan konten yang berisi *form input* data calon *voter*.
3. Footer berisi nama dari sistem.

- **Perancangan Halaman *View Result***

Perancangan halaman *view result* merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Administrator. Pada halaman *view result* ini Administrator dapat melihat hasil *voting*.



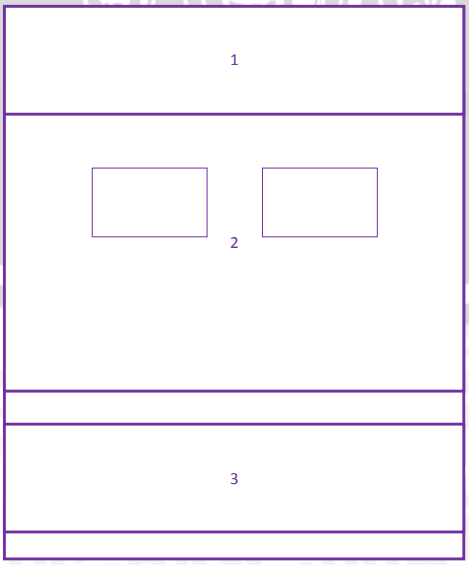
Gambar 3.20 Perancangan Antarmuka Halaman *View Result*

Halaman *view result* untuk Administrator menampilkan hasil data *voting*. Gambar 3.20 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi tabel data hasil *voting* per-kandidat.
3. Footer berisi nama dari sistem.

5. Perancangan Halaman Super Administrator

Perancangan halaman olah data untuk *Administrator* merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Super *Administrator*. *Administrator* dapat melakukan *generate* kunci dan *log* pemilihan.



Gambar 3.21 Perancangan Antarmuka Halaman Super Administrator

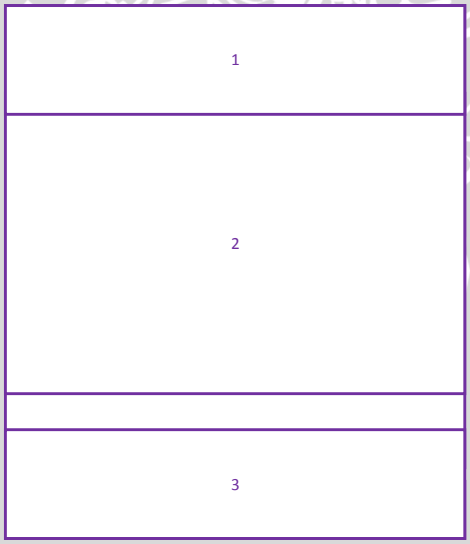
Halaman Super Administrator menampilkan menu olah data untuk Super Administrator. Gambar 3.21 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem e-voting dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi ikon menu untuk olah data yaitu generate kunci dan log pemilihan.
3. Footer berisi nama dari sistem

Berikut perancangan antarmuka halaman Super Administrator yaitu *generate* kunci dan *log* pemilihan :

- **Perancangan Halaman *Generate* Kunci**

Perancangan halaman *generate* kunci merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Super *Administrator* yang dapat melakukan *generate* kunci RSA.



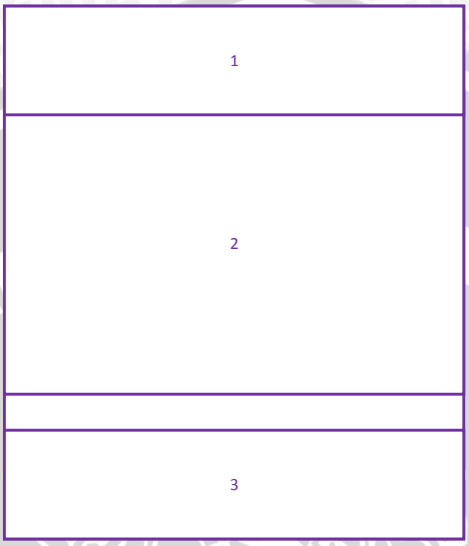
Gambar 3.22 Perancangan Antarmuka Halaman *Generate* Kunci RSA

Halaman *generate kunci* untuk Super *Administrator* menampilkan hasil *generate* kunci RSA. Gambar 3.21 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi ikon *generate* dan hasil *generate* kunci RSA.
3. Footer berisi nama dari sistem.

- **Perancangan Halaman *Log* Pemilihan**

Perancangan halaman olah data *log* pemilihan merupakan perancangan antarmuka pengguna yang memiliki hak akses sebagai Super *Administrator* yang berhak memiliki kunci privat dan dapat melakukan dekripsi hasil *voting*.



Gambar 3.23 Perancangan Antarmuka Halaman Olah Data *Log* Pemilihan

Halaman *log* pemilihan untuk Super *Administrator* menampilkan data hasil *voting* pada masing-masing *voter*. Gambar 3.22 memiliki keterangan sebagai berikut :

1. Header disebelah kiri berisi identitas logo sistem *e-voting* dan disebelah kanan berisi menu untuk fasilitas pengguna.
2. Konten berisi ikon *upload* kunci privat. Setelah meng-*upload* kunci privat akan memasuki halaman yang konten berisi tabel hasil data *voting* pada masing-masing *voter*.
3. Footer berisi nama dari sistem

BAB IV
IMPLEMENTASI

Bab ini membahas mengenai tahapan implementasi sistem perangkat lunak untuk *e-voting* dan keamanan data *e-voting* berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak. Implementasi terdiri atas penjelasan spesifikasi sistem, batasan-batasan dalam implementasi, implementasi basis data. Implementasi tiap *class* pada *file* program, implementasi algoritma, dan implementasi antarmuka.

4.1 Spesifikasi Lingkungan Sistem

Sistem perangkat lunak untuk *e-voting* dan keamanan data *e-voting* dikembangkan dalam lingkungan implementasi yang terdiri dari perangkat lunak dan perangkat lunak.

4.1.1 Spesifikasi Lingkungan Perangkat Keras

Spesifikasi lingkungan perangkat keras yang digunakan dalam proses pengembangan sistem perangkat lunak untuk *e-voting* dan keamanan data *e-voting* dijelaskan pada tabel 4.1.

Tabel 4.1 Spesifikasi Lingkungan Perangkat Keras Komputer

Dell Inspiron 5420	
Processor	Inter(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz
Memory (RAM)	4 GB
Harddisk	1TB HDD

4.1.2 Spesifikasi Lingkungan Perangkat Lunak

Spesifikasi lingkungan perangkat lunak yang digunakan dalam proses pengembangan sistem perangkat lunak untuk *e-voting* dan keamanan data *e-voting* dijelaskan pada Tabel 4.2.

Tabel 4.2 Spesifikasi Lingkungan Perangkat Lunak Komputer

Dell Inspiron 5420	
Operating System	Microsoft Windows 8 Pro 64-bit
Programming Language	*HyperText Markup Language (HTML), HyperText Preprocessor (PHP), Javascript*
Software Development Kit	Mozilla Firefox 26.0
Basis Data *Management System*	MySQL
Integrated Development Environment	*Adobe Dreamweaver CS6*, OpenSSL *Library* 1.0.1

4.2 Batasan-Batasan Implementasi

Beberapa batasan-batasan dalam mengimplementasikan sistem perangkat lunak untuk *e-voting* dan keamanan data *e-voting* adalah sebagai berikut :

1. Sistem perangkat lunak untuk *e-voting* dikerjakan dengan bahasa pemrogramanan PHP dengan *Framework CodeIgniter* dan basis data MySQL.
2. Sistem keamanan perangkat lunak untuk data *e-voting* dikerjakan dengan menggunakan Library OpenSSL pada PHP.
3. Algoritma kriptografi kunci-publik yang digunakan adalah RSA.
4. Pengujian sistem *e-voting* hanya dilakukan pada *browser*.

4.3 Implementasi Kriptografi Kunci-Publik

Sistem *e-voting* memiliki fitur utama yaitu Implementasi jaminan *confidentiality data e-voting* menggunakan salah satu algoritma kriptografi kunci-publik yaitu RSA. Implementasi tersebut akan direpresentasikan dalam bentuk *code* dengan bahasa pemrograman PHP pada *framework CodeIgniter*.

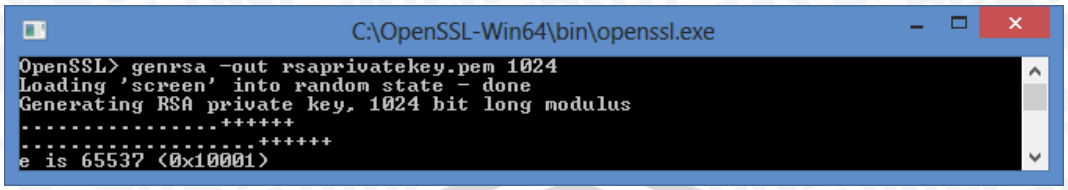
4.3.1 Proses Generate Kunci RSA

Proses *generate* kunci RSA menggunakan OpenSSL :

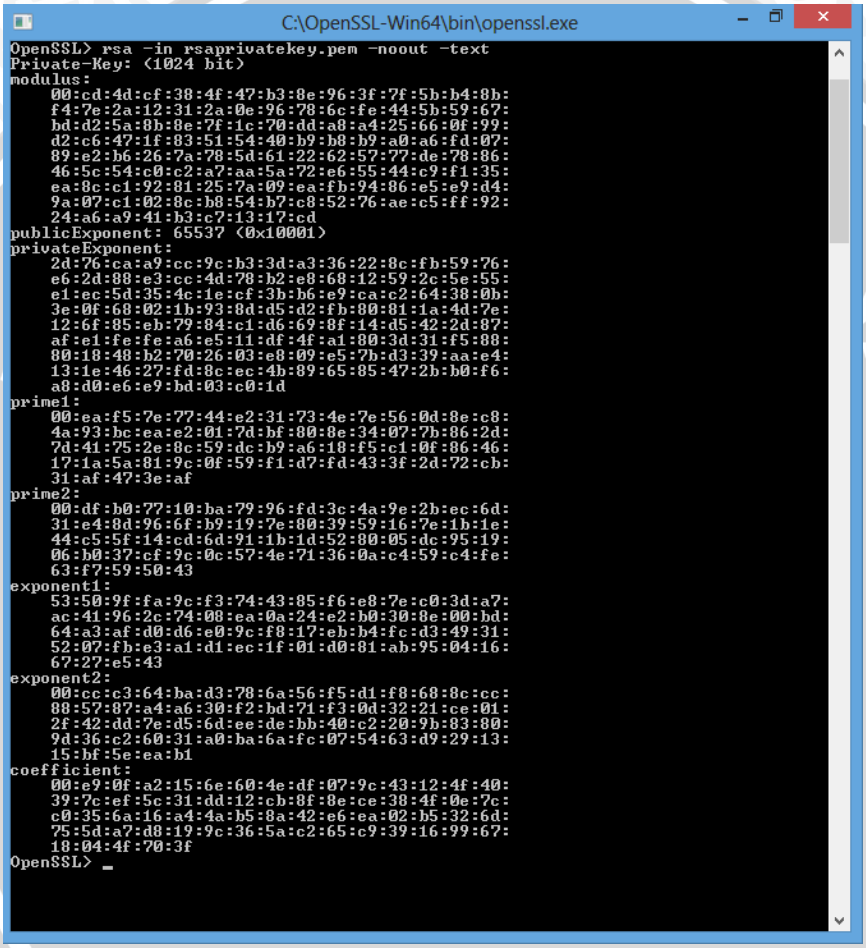
- *generate private key*

Pada gambar 4.1 *generate* kunci privat dengan panjang kunci dan *filename*.pem yang menyimpan hasil *output* kunci privat. Pada gambar 4.2

menampilkan abstrak nilai eksponen kunci privat. Gambar 4.3 menunjukan hasil kunci private dari *generate key*.



Gambar 4.1 Proses Generate Private Key

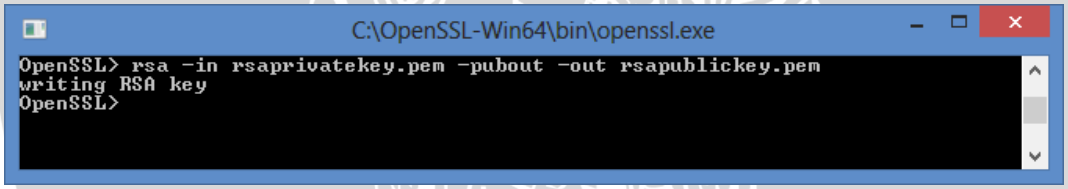


Gambar 4.2 Menampilkan Proses *Private Key*

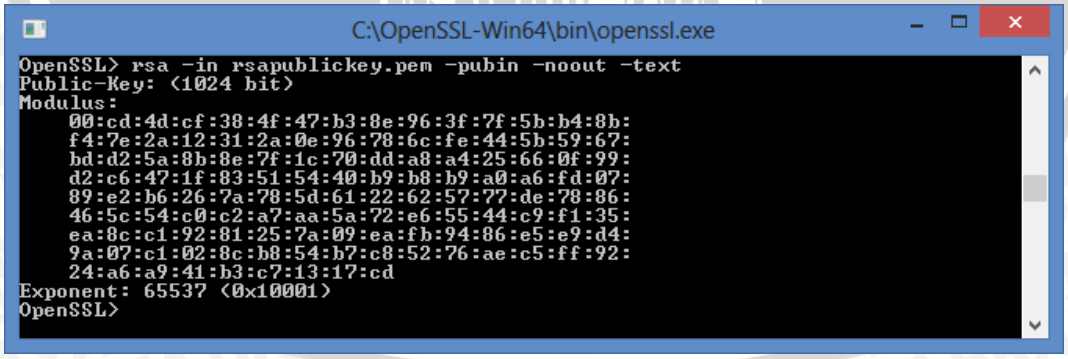
Gambar 4.3 Hasil Generate Private Key

- *generate public key*

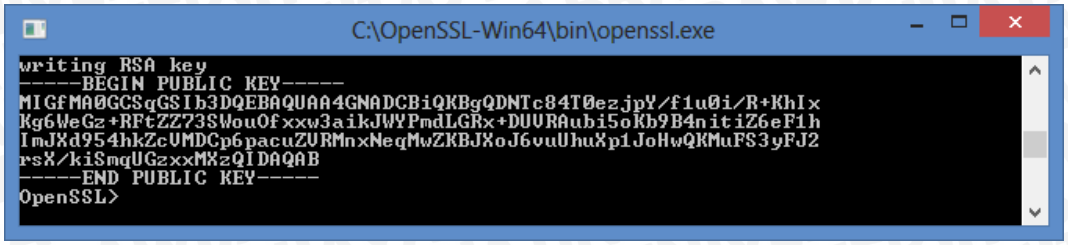
Pada gambar 4.4 *generate* kunci publik diperoleh dari kunci privat. Secara *default* kunci privat berisi semua informasi yang dibutuhkan untuk *generate* kunci publik. Informasi yang dibutuhkan meliputi nilai modulus dan eksponen. Gambar 4.5 menampilkan abstrak nilai eksponen kunci publik. Gambar 4.6 menunjukan hasil kunci publik dari *generate key*. Hasil *generate private key* dan *public key* pada OpenSSL akan disimpan dalam file dengan format .pem.



Gambar 4.4 Proses Generate Public Key



Gambar 4.5 Proses Menampilkan Proses *Public Key*



Gambar 4.6 Hasil Kunci Publik

4.3.2 Proses Enkripsi dan Dekripsi Data E-Voting

- Proses enkripsi pada data *voting* menggunakan *public key* :

```
1.  $fp=fopen("result/public.pem",'r');
2.            $pubkey=fread($fp,8192);
3.            fclose($fp);
4.       $pkey=openssl_get_publickey($pubkey);
5.
6.  openssl_public_encrypt($username."-
7.  ".$id_calon,$encrypted_id_calon,$pubkey);
8.       $e_id_calon=base64_encode($encrypted_id_calon);
```

Gambar 4.7 Implementasi Proses Enkripsi

Penjelasan implementasi proses enkripsi data *e-voting* pada gambar 4.7 :

1. Baris 1-4 berfungsi memanggil kunci publik RSA untuk melakukan enkripsi.
2. Baris 6-7 berfungsi untuk mengenkripsi data hasil pilihan pemilih dengan kunci-publik RSA.
3. Baris 8 berfungsi untuk mengenkripsi data hasil pemilihan dengan diencoding (diformat dalam bentuk karakter ASCII).

- Proses dekripsi pada data *voting* dilakukan dengan meng-*upload private key*:

```
1.  $data = array('upload_data' => $this->upload->data());
2.        $file = $this->upload->data();
3.
4.       $fp=fopen("temp/".$file['file_name'],'r');
5.        $privkey=fread($fp,8192);
6.        fclose($fp);
7.         unlink('././temp/'.$file['file_name']);
```

```
8.
9.      $log_vote = $this->m_vote->logVote();
10.     $rows = 0;
11.                 foreach($log_vote as $row){
12. openssl_private_decrypt(base64_decode($row-
13. >id_calon),$decrypted_id_calon,$privkey);
14. $usr=explode("-", $decrypted_id_calon);
15. if($row->username == $usr[0])
16.     $keterangan="Valid";
17.   else
18.     $keterangan="Tidak Valid";
```

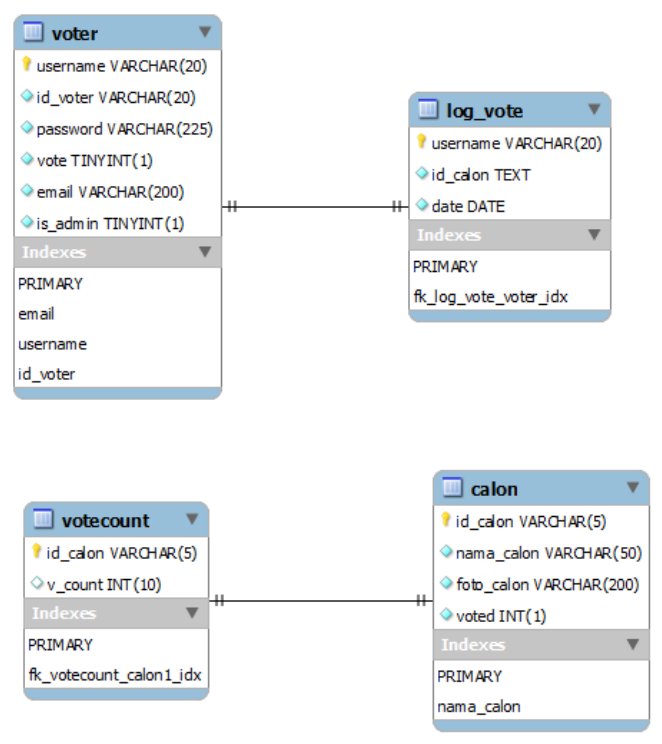
Gambar 4.8 Implementasi Proses Dekripsi

Penjelasan implementasi proses dekripsi data *e-voting* pada gambar 4.8:

1. Baris 1-2 berfungsi untuk meng-upload data dalam *array*.
2. Baris 4-6 berfungsi untuk membuka dan membaca file *private key* yang telah diupload.
3. Baris 7 berfungsi untuk menghapus *file.*
4. Baris 9-11 perintah *query* di model *vote.*
5. Baris 12-13 berfungsi untuk mendekripsi data *voting* pada log vote.
6. Baris 14 berfungsi untuk mengeksplode username pada data dekripsi id calon.
7. Baris 15-18 berfungsi untuk mencocokan username dengan username yang telah di eksplode pada data dekripsi id calon.

4.4 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan basis data *management system* MySQL. Hasil implementasi penyimpanan data ini berupa *script* SQL. Hasil implementasi SQL pada basis data ini dimodelkan dalam diagram konseptual *entity relationship*. Gambar 4.9 menunjukan diagram konseptual *entity relationship* dari sistem perangkat lunak untuk *e-voting* dan keamanan data *e-voting*.



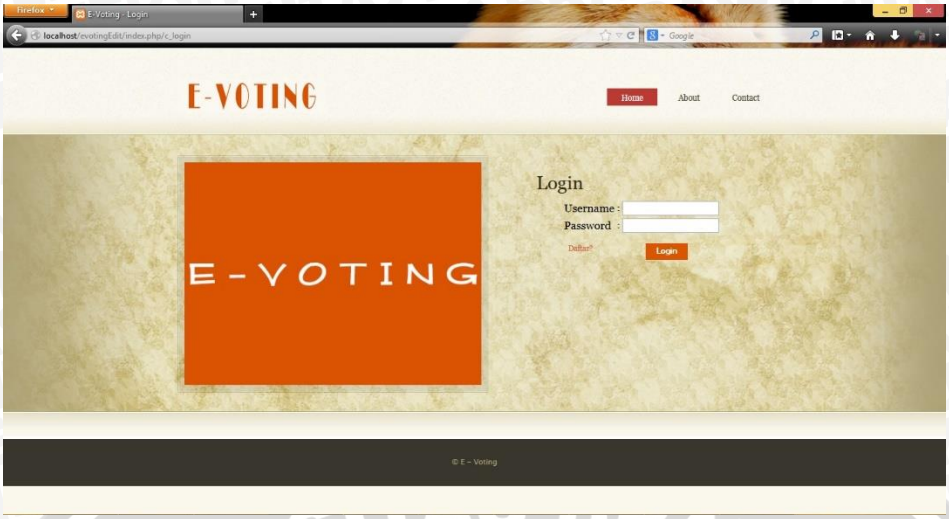
Gambar 4.9 Diagram ER Konseptual Dari Sistem

4.5 Implementasi Antar Muka

Antarmuka aplikasi *e-voting* digunakan oleh pengguna untuk berinteraksi dengan sistem perangkat lunak. Antarmuka perangkat lunak ini dibagi menjadi 9, yaitu antarmuka halaman utama, halaman pendaftaran *voter* baru, halaman olah vote sekarang, halaman olah data untuk Administrator, halaman olah data kandidat untuk adminitrator, halaman olah data *voter* untuk Administrator, halaman *result* untuk Administrator, halaman *generate* kunci untuk Super Administrator dan olah data log pemilihan untuk Super Administrator.

4.5.1 Antar Muka Halaman Utama

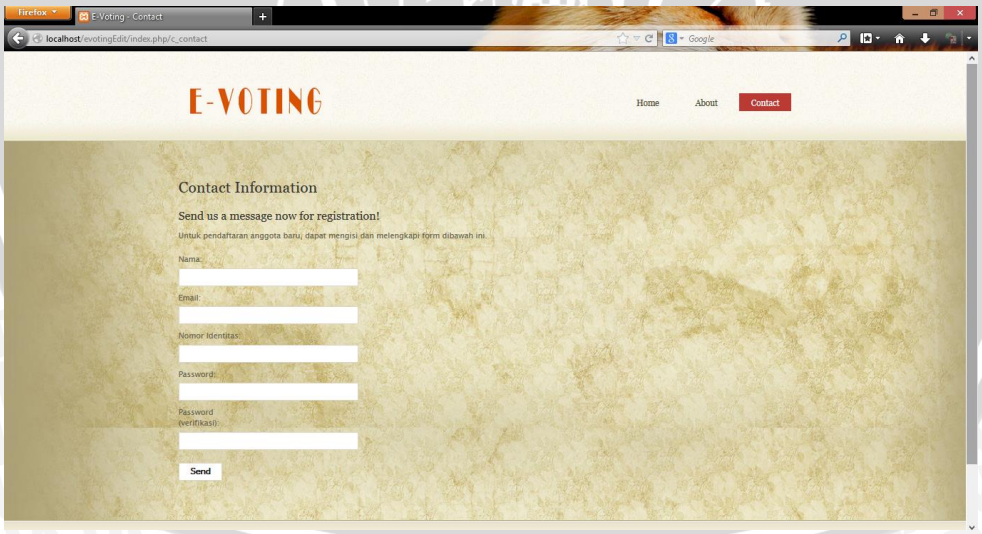
Halaman utama merupakan halaman pertama yang akan dibuka ketika sistem dijalankan. Dari halaman utama pengguna mendapatkan gambar kandidat dan tempat untuk melakukan *login* dan menu untuk melakukan pendaftaran. Gambar 4.10 menunjukkan implementasi tampilan antarmuka dari halaman utama yang mengacu pada perancangan antarmuka halaman utama Sub Bab 3.2.2.6.



Gambar 4.10 Antar Muka Halaman Utama

4.5.2 Antar Muka Halaman Pendaftaran Calon Voter

Halaman pendaftaran *voter* baru merupakan halaman untuk calon *voter* melakukan pendaftaran. Calon *voter* dapat memasuki halaman *contact* dan mengisi *form* pendaftaran. Gambar 4.11 menunjukkan implementasi tampilan antarmuka dari halaman pendaftaran calon *voter* yang mengacu pada perancangan antarmuka halaman pendaftaran calon *voter* Sub Bab 3.2.2.6.



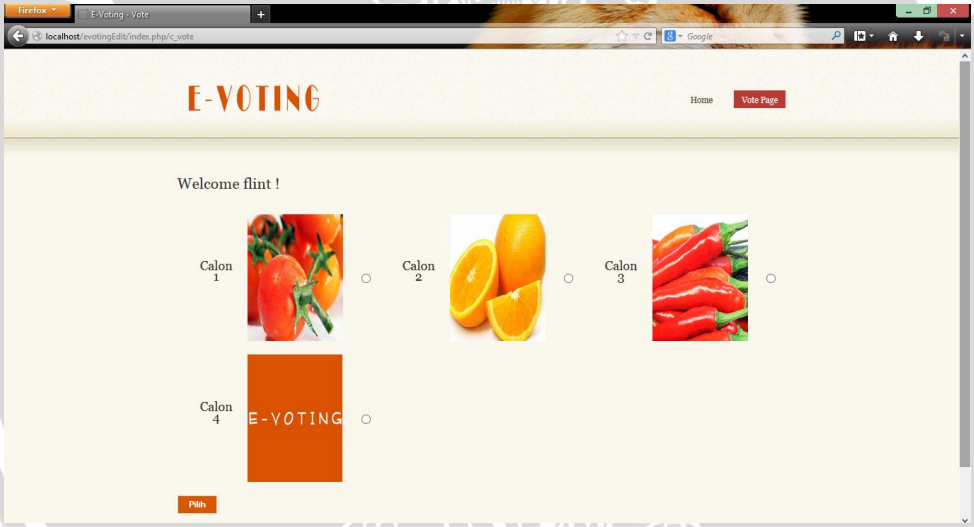
Gambar 4.11 Antar Muka Halaman Pendaftaran Calon *Voter*

4.5.3 Antar Muka Halaman Vote Sekarang untuk Voter

Halaman vote sekarang merupakan halaman untuk melakukan proses *voting*. *Voter* dapat menekan tautan *vote* sekarang lalu memasuki halaman *voting* yang berisi daftar kandidat. Gambar 4.12 dan gambar 4.13 menunjukkan implementasi

tampilan antarmuka dari halaman *vote* sekarang yang mengacu pada perancangan antarmuka halaman *vote* sekarang Sub Bab 3.2.2.6.

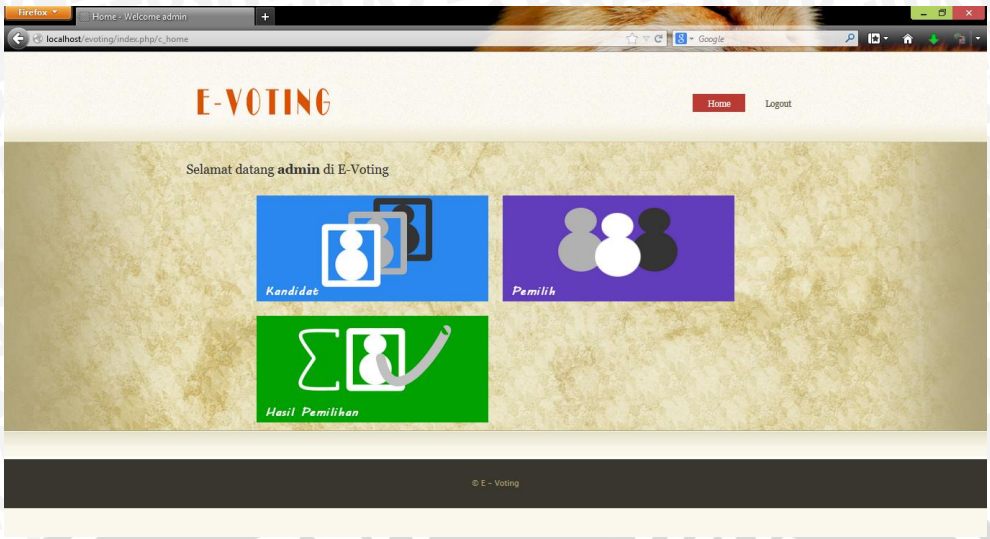
Gambar 4.12 Antar Muka Halaman *Vote* Sekarang



Gambar 4.13 Antar Muka Halaman *Vote* Sekarang

4.5.4 Antar Muka Halaman Olah Data untuk Administrator

Halaman olah data untuk Administrator merupakan halaman untuk melakukan olah data. Administrator dapat melakukan olah data kandidat, olah data *voter*, dan lihat hasil pemilihan. Gambar 4.14 menunjukkan implementasi tampilan antarmuka dari halaman olah data untuk Administrator yang mengacu pada perancangan antarmuka halaman olah data untuk Administrator Sub Bab 3.2.2.6.

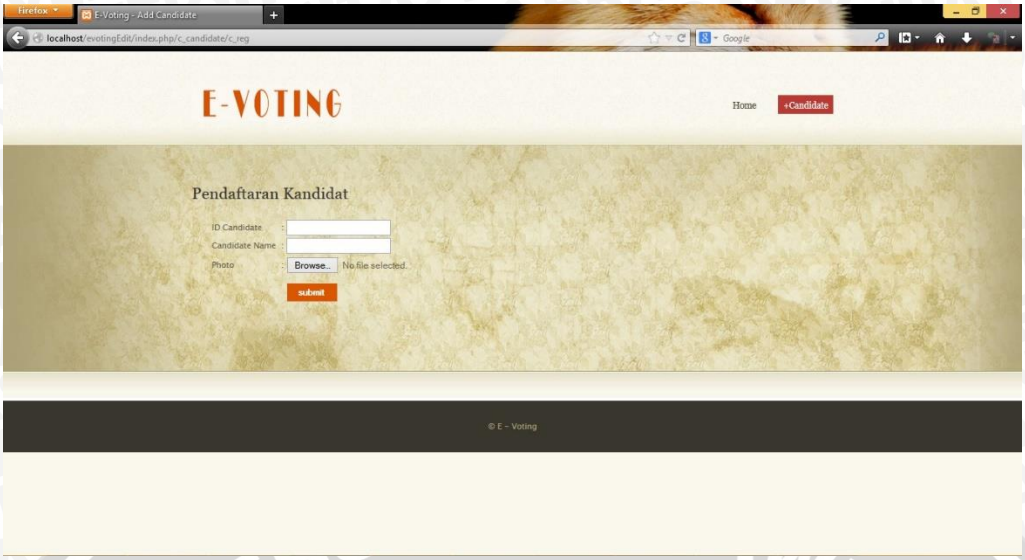


Gambar 4.14 Antar Muka Halaman Olah Data Administrator

- **Antar Muka Halaman Olah Data Kandidat untuk Administrator**

Halaman olah data kandidat untuk Administrator merupakan halaman untuk melakukan olah data kandidat. Administrator dapat melakukan lihat data kandidat, tambah data kandidat, dan hapus data kandidat. Gambar 4.15 dan gambar 4.16 menunjukkan implementasi tampilan antarmuka dari halaman olah data kandidat untuk Administrator yang mengacu pada perancangan antarmuka halaman olah data kandidat untuk Administrator Sub Bab 3.2.2.6.

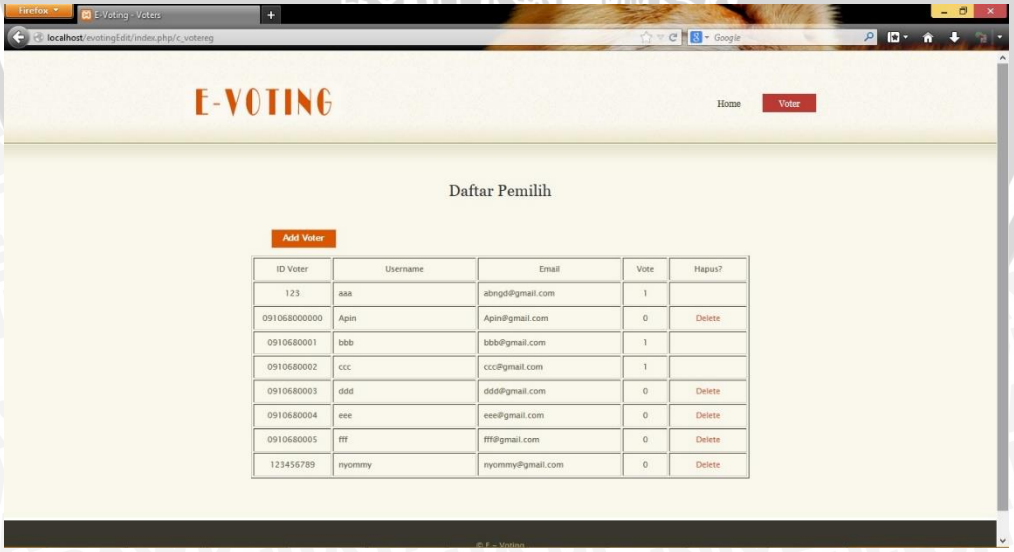
Gambar 4.15 Antar Muka Halaman Olah Data Kandidat



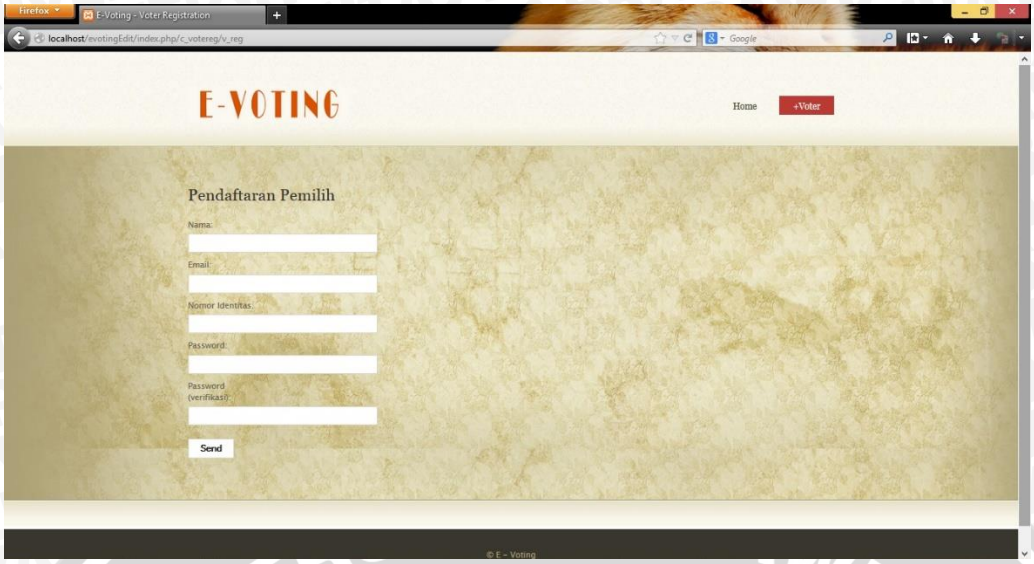
Gambar 4.16 Antar Muka Halaman *Form* Data Kandidat

- **Antar Muka Halaman Olah Data *Voter* untuk Administrator**

Halaman olah data *voter* untuk Administrator merupakan halaman untuk melakukan olah data *voter*. Administrator dapat melakukan lihat data *voter*, tambah data *voter*, dan hapus data *voter*. Gambar 4.17 dan gambar 4.18 menunjukkan implementasi tampilan antarmuka dari halaman olah data *voter* untuk Administrator yang mengacu pada perancangan antarmuka halaman olah data *voter* untuk Administrator Sub Bab 3.2.2.6.



Gambar 4.17 Antar Muka Halaman Olah Data *Voter*



Gambar 4.18 Antar Muka Halaman *Form* Data *Voter*

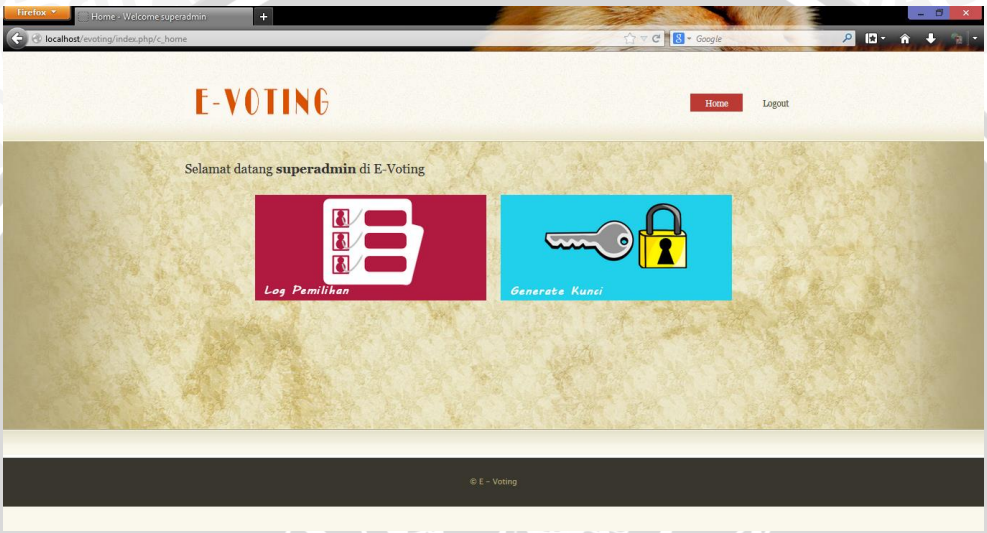
- **Antar Muka Halaman *View Result***

Halaman *view result* untuk Administrator merupakan halaman untuk melakukan lihat data hasil *voting*. Administrator dapat melakukan lihat data hasil *voting*. Gambar 4.19 menunjukkan implementasi tampilan antarmuka dari halaman *view result* untuk Administrator yang mengacu pada perancangan antarmuka halaman *view result* untuk Administrator Sub Bab 3.2.2.6.

Gambar 4.19 Antar Muka Halaman *View Result*

4.5.5 Antar Muka Halaman Super Administrator

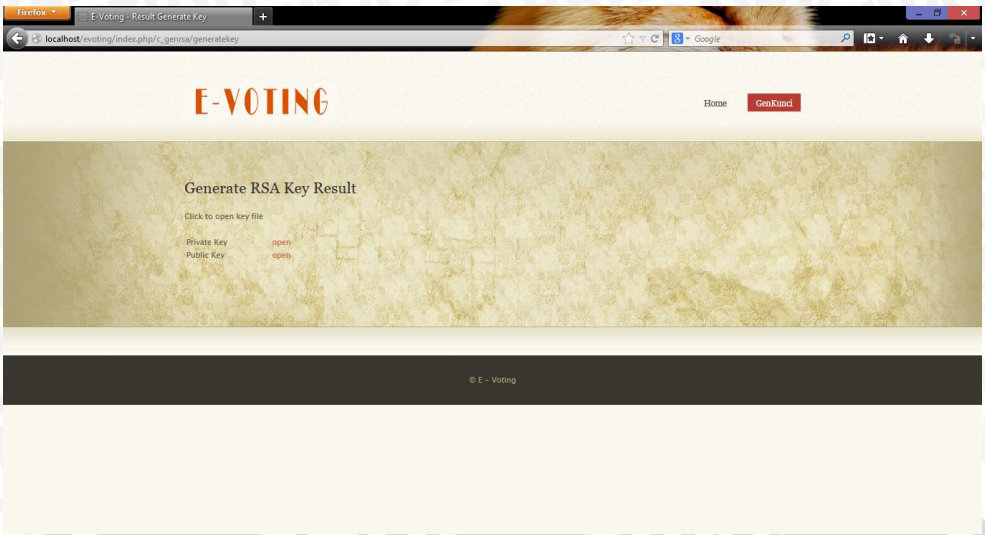
Halaman Super Administrator merupakan halaman untuk melakukan olah data oleh Super Administrator. Super Administrator dapat melakukan *generate* kunci RSA dan lihat *log* pemilihan. Gambar 4.20 menunjukkan implementasi tampilan antarmuka dari halaman olah data untuk Super Administrator yang mengacu pada perancangan antarmuka halaman olah data untuk Super Administrator Sub Bab 3.2.2.6.



Gambar 4.20 Halaman Menu Super Administrator

- **Antar Muka Halaman *Generate* Kunci RSA**

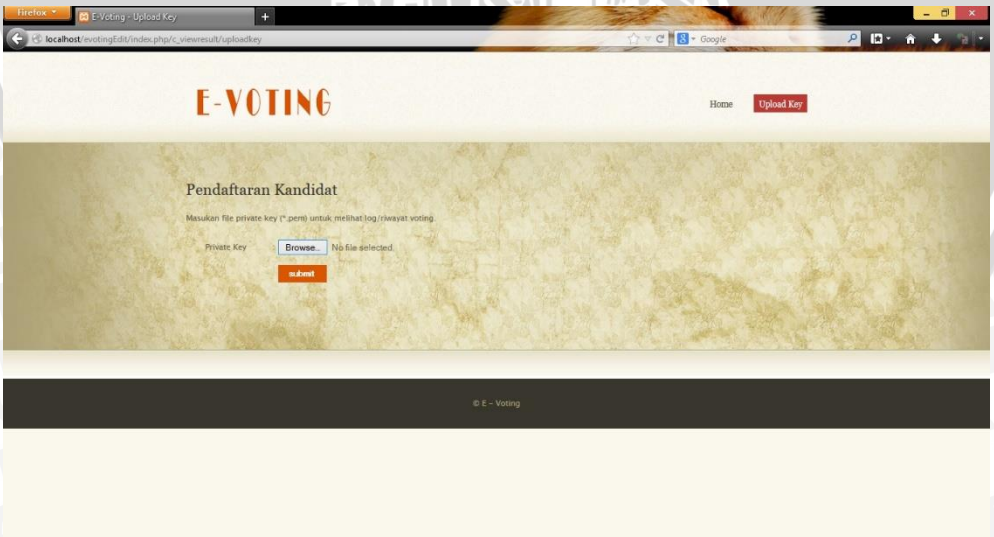
Halaman *generate* kunci untuk Super Administrator merupakan halaman untuk melakukan *generate* kunci. Super Administrator dapat melakukan *generate* kunci RSA. Gambar 4.21 menunjukkan implementasi tampilan antarmuka dari halaman *generate* kunci untuk Super Administrator yang mengacu pada perancangan antarmuka halaman *generate* kunci untuk Super Administrator Sub Bab 3.2.2.6.



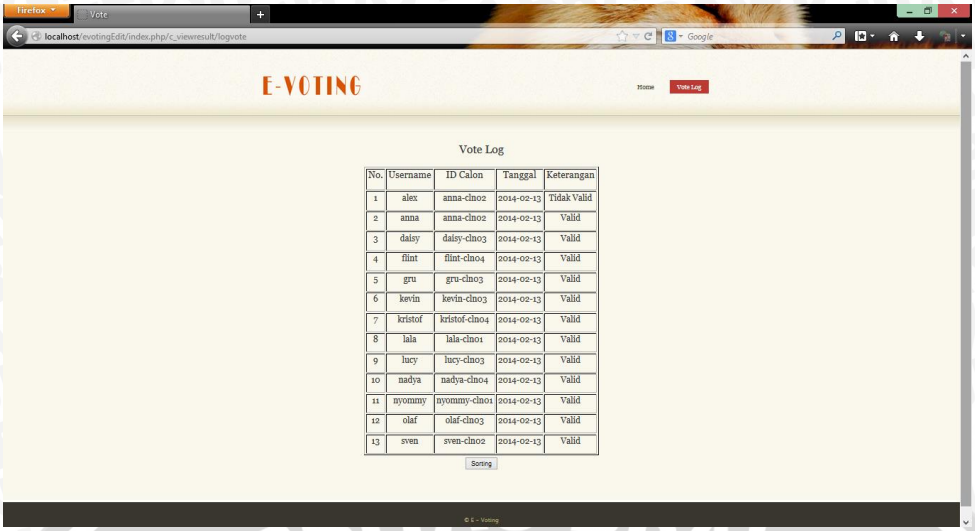
Gambar 4.21 Halaman *Generate* Kunci

- **Antar Muka Halaman Olah Data *Log* Pemilihan**

Halaman olah data *log* pemilihan untuk Super Administrator merupakan halaman untuk melakukan olah data dan melihat data hasil *voting* masing-masing *voter*. Super Administrator dapat melihat data hasil *voting* masing-masing *voter*. Gambar 4.22 dan gambar 4.23 menunjukkan implementasi tampilan antarmuka dari halaman olah data *log* pemilihan untuk Super Administrator yang mengacu pada perancangan antarmuka halaman olah data *log* pemilihan untuk Super Administrator Sub Bab 3.2.2.6.



Gambar 4.22 Antar Muka Halaman Olah Data *Log* Pemilihan



Gambar 4.23 Antar Muka Halaman Lihat Data *Log* Pemilihan

BAB V
PENGUJIAN DAN ANALISIS

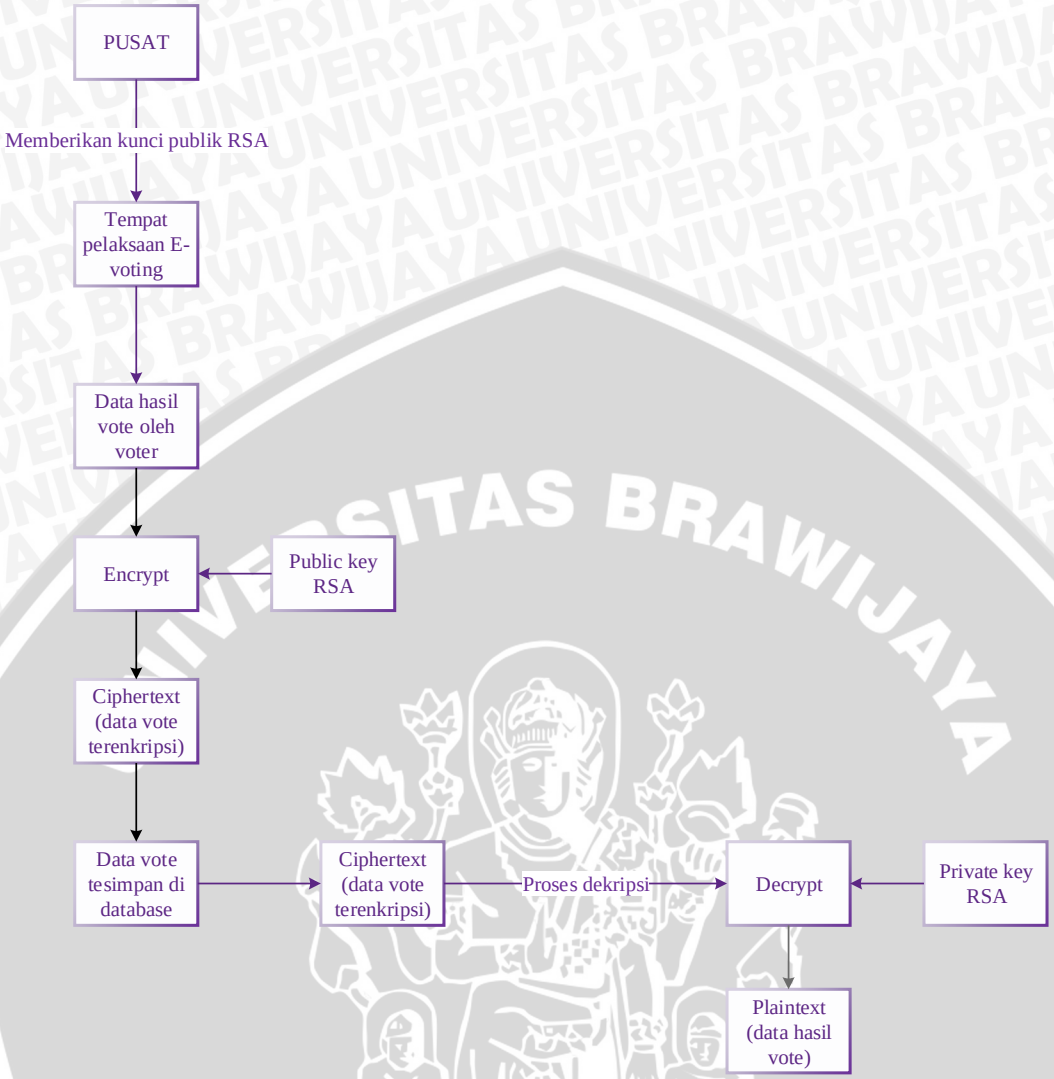
Bab ini membahas mengenai tahapan pengujian dan analisis sistem *e-voting* yang menggunakan kriptografi kunci-publik yaitu RSA untuk kevalidan data hasil *voting*.

5.1 Pengujian

Proses pengujian dilakukan dengan 3 tahapan yaitu pengujian sistem kriptografi, pengujian keamanan dan pengujian validasi. Pengujian validasi dilakukan untuk mengetahui apakah sistem yang dibangun sudah menyediakan fungsi-fungsi yang sesuai dengan yang dibutuhkan. Pengujian sistem kriptografi dilakukan untuk mengetahui apakah kriptografi kunci-publik yaitu RSA berhasil diterapkan dan telah berfungsi dengan baik sesuai dengan tujuan pada sistem. Pengujian keamanan dilakukan untuk mengetahui apakah data hasil *voting* dapat terjamin keamanannya.

5.1.1 Pengujian Sistem Kriptografi

Pengujian sistem kriptografi pada *e-voting* digunakan untuk mengetahui apakah pelaksanaan proses *confidentiality* data *e-voting* dengan kriptografi kunci-publik yaitu RSA telah berfungsi dengan baik dan sesuai dengan baik. Pada pengujian ini dapat diketahui apakah data *e-voting* yang di dekripsi sesuai dengan data asli yang awal dikirim dan bagaimana apabila terjadi perubahan data selama proses *voting*. Pada gambar 5.1 akan dijelaskan bagaimana proses validasi data *e-voting*.



Gambar 5.1 Diagram Proses Pengujian Sistem Kriptografi

Pada awal sebelum proses *voting*, dilakukan pengiriman kunci antara pusat dengan tempat pelaksanaan *voting*. Pusat akan memberikan kunci publik RSA yang akan digunakan untuk mengenkripsi data hasil *voting* pada masing-masing *voter*. Kemudian hasil data *voting* yang sudah di enkripsi oleh kunci publik RSA akan di simpan di *database*. Dilakukan pengujian validasi untuk data hasil *voting*.

- Hasil enkripsi data hasil *voting* dengan kandidat yang sudah dipilih oleh *voter* menggunakan kunci publik RSA pada gambar 5.2.

username	id_calon	date
alex	LssuJauB5wRhpicLoDB5qmu9Ib/npbj+TYELJGb7qYbij+W0Sg...	2014-02-05
anna	oxlFqTj1ficF3gMBtMEctA/9J4UdQAz/XVSYSgMQZcOg2thD0n...	2014-02-05
flint	0nZ7HtRhAqqCRtCLLEoOYM8A0M3saVRE2HwX0pY4v3MzLQIB2P...	2014-02-05
gru	WTrIlcYM9U3liUvD/ruyhljnqnnzlye6DroGrXGl4DYj8LN8aj...	2014-02-05
kevin	J55L6MwDhnMAc555rqUEDjiEt+WKXunQjYHaip8BABsFlfJpcl...	2014-02-05
kristof	IXGj+Y0esi4sUCIQ2e0qnwOBx22edkJUBFxf9RykDdZ3LpqwxN...	2014-02-05
lala	C20Stg0iFBZds5sExjGb2k0vFgOuq33JhI3PJVUr2GE/6uAZX0...	2014-02-05
lucy	Avm0JvG2fgTnf51RwAI3drHpaZe7PaR22OVZxex4iL5I9UTjSO...	2014-02-05
nadya	g34Myc+O+5PyK4q+VDu4CB3N5rEySCnvWDs0HpvkEV+74FbtoX...	2014-02-05
nyommy	WNQQTAbs31K9+RDx/TbNnoeEbrCiKMo2xpA9mi1K2ipqYTz738...	2014-02-05
olaf	K21na40aGZcf/HngBaJWydw/PRVAtQJaYz1d4oZsIhX0YyFADv...	2014-02-05
sven	uem02KXSWigL1/CDptRgDBx2AkRvJA2RrlHcuOfHVVvVHbzyrQ...	2014-02-05

Gambar 5.2 Enkripsi Data Hasil *Voting* dengan Kunci Publik RSA

- Hasil dekripsi data hasil *voting* menggunakan kunci privat RSA pada *log* pemilihan untuk mencocokan data hasil *voting* masing-masing *voter* pada gambar 5.3.

Vote Log

No.	Username	ID Calon	Tanggal
1	alex	cln03	2014-02-05
2	anna	cln02	2014-02-05
3	flint	cln03	2014-02-05
4	gru	cln03	2014-02-05
5	kevin	cln04	2014-02-05
6	kristof	cln03	2014-02-05
7	lala	cln02	2014-02-05
8	lucy	cln04	2014-02-05
9	nadya	cln02	2014-02-05
10	nyommy	cln01	2014-02-05
11	olaf	cln02	2014-02-05
12	sven	cln03	2014-02-05

Sorting

Gambar 5.3 Hasil Dekripsi Data Hasil *Voting* dengan Kunci Privat RSA

Tabel 5.1 Kasus Uji Sistem Kriptografi Kunci-Publik

Nama kasus uji	Validasi hasil enkripsi
Tujuan pengujian	Untuk menguji validitas enkripsi pada data hasil *voting*
Prosedur uji	Melakukan dekripsi pada data hasil *voting* dengan menggunakan kunci privat RSA
Hasil yang diharapkan	Sistem dapat melakukan dekripsi dengan *private key* pada data hasil *voting* yang sudah dienkripsi dengan *public key* kemudian menampilkannya
Hasil yang didapatkan	Sistem melakukan dekripsi dengan *private key* pada data hasil *voting* kemudian hasilnya dapat ditampilkan
Status Validalitas	Valid

5.1.2 Pengujian Keamanan

Pengujian keamanan pada *e-voting* dilakukan untuk mengetahui apakah data hasil *voting* sudah aman dengan adanya jaminan *confidentiality*. Pengujian keamanan dilakukan dengan 2 skenario manipulasi data. Manipulasi data yang dilakukan oleh *Administrator* adalah mengubah (*update*) isi hasil *voting* :

1. Administrator melakukan manipulasi data hasil *voting* dengan merubah isi pada tabel votecount:

- Hasil ada tabel votecount pada calon 4 yang awalnya berjumlah 4 dirubah menjadi 7 pada gambar 5.4:

id_calon	v_count
cln01	1
cln02	4
cln03	5
cln04	2

id_calon	v_count
cln01	1
cln02	7
cln03	5
cln04	2

Gambar 5.4 Manipulasi Data Hasil *Voting* Pada Votecount

- Hasil pada menu hasil pemilihan menunjukan data hasil *voting* pada calon 2 berubah menjadi 7 pada gambar 5.5:

Hasil Pemilihan Suara

	Calon 1	1 vote
	Calon 2	7 vote
	Calon 3	5 vote
E-VOTING	Calon 4	2 vote

Gambar 5.5 Hasil Pemilihan Suara

- Dilakukan pencocokan data hasil *voting* pada *log pemilihan* dengan meng-*upload private key*. Data hasil *voting* calon 2 tetap berjumlah 4 dan tidak mengalami perubahan pada gambar 5.6:

Vote Log

	cln03	
No.	Username	Tanggal
1	alex	2014-02-05
2	flint	2014-02-05
3	gru	2014-02-05
4	kristof	2014-02-05
5	sven	2014-02-05
	Jumlah	5

	cln02	
No.	Username	Tanggal
1	anna	2014-02-05
2	lala	2014-02-05
3	nadya	2014-02-05
4	olaf	2014-02-05
	Jumlah	4

Gambar 5.6 Data Hasil *Voting* pada *Log* Pemilihan

2. Administrator melakukan manipulasi data hasil *voting* dengan merubah isi pada kolom log vote :

- Data hasil *voting* pada tabel log vote dengan nama alex, id calon yang terenkripsi yang dipilih oleh alex dirubah admin dengan id calon yang terenkripsi yang dipilih oleh anna pada gambar 5.7.

username	id_calon	date
alex	YS1Wy8OroYwVddQabk2cn7cf+MI33Rh1Cq5zsH5QMfxPFIFKgy...	2014-02-13
anna	YkdkFiyDObohNJzjMfl0EXf9GBL3w8YgtjphnaPzXPS2snMqcU...	2014-02-13
daisy	vr2OBovWIb0rqjz8ta5XQyKg1zadl/BgXv4x/jzTAdGL3useYx...	2014-02-13
flint	prmoAGITVRhWrCCQZJWif0yuMz/Qf/29EEdfsxHmSdnkRyu4Mr...	2014-02-13
gru	yvGLhX6kXJQ0bNdC8r9GteG5aP4OyE2zp3sTY0aP3RiUkn1GoL...	2014-02-13
kevin	IQZY7eSDGAHcr3QAlmyuKwuc3HReJIJTaVoyk4Q5gDZNMzxmlH...	2014-02-13
kristof	bxSnirAAxGyVbGhKwlBz46+L+piRerpPvSttoeycHIN/VVgQRs...	2014-02-13
lala	iwlzND4AVtJaXr4pyzaS6g+blVW0CL7C+RfvHsmMzlycXZgvVL...	2014-02-13
lucy	yQy9r/LDMTbc/do6aqVJ2pKjX3raM6J9WOqBQH5zwnT0w2tRMr...	2014-02-13
nadya	GuMPAeQV+dAddy0h8ow1yEhQ2ZtVyPwqhcpE9RVEG75Cmc63K/...	2014-02-13
nyommy	SyE41Asdyk9DtAUC9zG1ETe/62IGBbD2J23shdfaFSdH77QAKX...	2014-02-13
olaf	BBX0hnP9kqIRO0CkQyKmEzXBjz7jJPDWX0pgjIO6CcK5SM23fu...	2014-02-13
sven	qvXubms/5hz4IRu5WFvPCXbCcsepEI8cYKMXAxMGOer0mIGyPU...	2014-02-13

username	id_calon	date
alex	YkdkFiyDObohNJzjMfl0EXf9GBL3w8YgtjphnaPzXPS2snMqcU...	2014-02-13
anna	YkdkFiyDObohNJzjMfl0EXf9GBL3w8YgtjphnaPzXPS2snMqcU...	2014-02-13
daisy	vr2OBovWIb0rqjz8ta5XQyKg1zadl/BgXv4x/jzTAdGL3useYx...	2014-02-13
flint	prmoAGITVRhWrCCQZJWif0yuMz/Qf/29EEdfsxHmSdnkRyu4Mr...	2014-02-13
gru	yvGLhX6kXJQ0bNdC8r9GteG5aP4OyE2zp3sTY0aP3RiUkn1GoL...	2014-02-13
kevin	IQZY7eSDGAHcr3QAlmyuKwuc3HReJIJTaVoyk4Q5gDZNMzxmlH...	2014-02-13
kristof	bxSnirAAxGyVbGhKwlBz46+L+piRerpPvSttoeycHIN/VVgQRs...	2014-02-13
lala	iwlzND4AVtJaXr4pyzaS6g+blVW0CL7C+RfvHsmMzlycXZgvVL...	2014-02-13
lucy	yQy9r/LDMTbc/do6aqVJ2pKjX3raM6J9WOqBQH5zwnT0w2tRMr...	2014-02-13
nadya	GuMPAeQV+dAddy0h8ow1yEhQ2ZtVyPwqhcpE9RVEG75Cmc63K/...	2014-02-13
nyommy	SyE41Asdyk9DtAUC9zG1ETe/62IGBbD2J23shdfaFSdH77QAKX...	2014-02-13
olaf	BBX0hnP9kqIRO0CkQyKmEzXBjz7jJPDWX0pgjIO6CcK5SM23fu...	2014-02-13
sven	qvXubms/5hz4IRu5WFvPCXbCcsepEI8cYKMXAxMGOer0mIGyPU...	2014-02-13

Gambar 5.7 Manipulasi Data Hasil *Voting* pada Log vote

- Pencocokan data hasil *voting* pada *log* pemilihan dengan meng-*upload private key*. Data log vote alex telah mengalami perubahan pada saat admin merubah data hasil *voting* tersebut dengan data hasil *voting* anna. Namun id calon yang terdapat pada alex, memiliki *username* anna dan data *vote* alex tidak valid pada gambar 5.8.

Vote Log

No.	Username	ID Calon	Tanggal	Keterangan
1	alex	anna-cln02	2014-02-13	Tidak Valid
2	anna	anna-cln02	2014-02-13	Valid
3	daisy	daisy-cln03	2014-02-13	Valid
4	flint	flint-cln04	2014-02-13	Valid
5	gru	gru-cln03	2014-02-13	Valid
6	kevin	kevin-cln03	2014-02-13	Valid
7	kristof	kristof-cln04	2014-02-13	Valid
8	lala	lala-cln01	2014-02-13	Valid
9	lucy	lucy-cln03	2014-02-13	Valid
10	nadya	nadya-cln04	2014-02-13	Valid
11	nyommy	nyommy-cln01	2014-02-13	Valid
12	olaf	olaf-cln03	2014-02-13	Valid
13	sven	sven-cln02	2014-02-13	Valid

Gambar 5.8 Daftar Data Hasil *Voting* oleh *Voter*

5.1.3 Pengujian Validasi

Pengujian validasi dilakukan untuk mengetahui apakah sistem yang dibangun sudah benar sesuai dengan daftar kebutuhkan yang ditentukan sebelumnya. Item-item yang telah dirumuskan dalam daftar kebutuhan dan merupakan hasil analisis kebutuhan akan menjadi acuan untuk melakukan pengujian validasi. Pengujian validasi menggunakan metode pengujian *Black-box*, karena tidak konsentrasi pada jalannya algoritma program tetapi lebih difokuskan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan. Pada skripsi ini dilakukan pengujian validasi terhadap sistem *e-voting*.

5.1.3.1 Kasus Uji Validasi *Voting*

Kasus uji validasi *voting* pada tabel 5.2 :

Tabel 5.2 Kasus Uji Validasi *Voting*

Nama kasus uji	*Voting*
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas *voting*
Prosedur uji	1. *User* melakukan *login* 2. *User* masuk ke halaman *vote* sekarang 3. *User* memilih kandidat 4. *User* menekan tombol *vote*
Hasil yang diharapkan	Sistem dapat melakukan seleksi *user* pada saat proses *vote*. Sistem menampilkan halaman *vote* sekarang hanya untuk *user* yang belum melakukan *vote*, untuk *user* yang sudah *vote* tidak dapat mengakses halaman *vote* sekarang.
Hasil yang didapatkan	Sistem melakukan seleksi *user* yang sudah dan yang belum melakukan *vote*. Sistem menampilkan halaman *vote* sekarang untuk *user* yang belum melakukan *vote*
Status Validalitas	Valid

5.1.3.2 Kasus Uji Validasi Pendaftaran *Voter* Baru

Kasus uji validasi pendaftaran *voter* baru pada tabel 5.3 :

Tabel 5.3 Kasus Uji Validasi Pendaftaran Pemilih (*Voter*) Baru

Nama kasus uji	Pendaftaran pemilih
Tujuan pengujian	Untuk menguji validitas kinerja sistem dalam menyediakan fasilitas pendaftaran pemilih baru
Prosedur uji	1. Pemilih masuk ke halaman *contact* atau menekan tautan daftar 2. Pemilih mengisikan data yang diperlukan

	3. Pemilih baru menekan tombol *send*
Hasil yang diharapkan	Sistem dapat melakukan penyimpanan data pemilih baru. Data tersebut dapat dilihat oleh Administrator
Hasil yang didapatkan	Sistem menyimpan data pemilih baru. Administrator dapat melihat data baru tersebut
Status Validalitas	Valid

5.1.3.3 Kasus Uji Validasi Olah Data *Voter*

Kasus uji validasi olah data *voter* oleh Administrator sebagai berikut pada tabel 5.4, tabel 5.5 dan tabel 5.6 :

Tabel 5.4 Kasus Uji Validasi Lihat Data *Voter*

Nama kasus uji	Lihat data *voter*
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas lihat data *voter* oleh Administrator
Prosedur uji	1. Administrator melakukan login 2. Masuk ke halaman menu 3. Masuk ke halaman pemilih
Hasil yang diharapkan	Sistem dapat menampilkan data *voter* sesuai dengan data *voter* yang terdaftar.
Hasil yang didapatkan	Sistem menampilkan data *voter* sesuai dengan data *voter* yang terdaftar.
Status Validalitas	Valid

Tabel 5.5 Kasus Uji Validasi Penambahan Data *Voter*

Nama kasus uji	Penambahan data *voter*
Tujuan pengujian	Untuk menguji validitas kinerja sistem dalam menyediakan fasilitas penambahan *voter* oleh Administrator
Prosedur uji	1. Administrator melakukan login 2. Administrator masuk ke halaman menu

	3. Administrator masuk ke halaman pemilih 4. Administrator menekan tombol *add voter* 5. Administrator mengisi data pemilih baru 6. Administrator menekan tombol send 7. Administrator melihat data pemilih baru yang baru dimasukkan ke dalam daftar pemilih
Hasil yang diharapkan	Sistem dapat melakukan penyimpanan data baru pemilih (*voter*). Lalu dapat melihat data pemilih baru yang sudah tersimpan di daftar pemilih pada halaman pemilih.
Hasil yang didapatkan	Sistem menyimpan data baru pemilih. Dapat melihat data pemilih baru di daftar pemilih pada halaman pemilih.
Status Validalitas	Valid

Tabel 5.6 Kasus Uji Validasi Hapus Data *Voter*

Nama kasus uji	Hapus *voter*
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas untuk menghapus pemilih (*voter*) yang belum melakukan *voting* oleh *Administrator*
Prosedur uji	1. Administrator melakukan *login* 2. Administrator masuk ke halaman menu 3. Administrator masuk ke halaman pemilih 4. Administrator menekan tautan *delete* pada tabel hapus
Hasil yang diharapkan	Sistem dapat melakukan seleksi *voter* yang sudah dan yang belum melakukan *voting*. Menampilkan tautan *delete* pada *voter* yang belum melakukan *voting* dan melakukan penghapusan *voter* yang telah dipilih.
Hasil yang didapatkan	Sistem melakukan seleksi *voter* yang sudah dan yang belum melakukan *voting*. Menampilkan tautan *delete*

	pada *voter* yang belum melakukan *voting* dan melakukan penghapusan *voter* yang telah dipilih.
Status Validalitas	Valid

5.1.3.4 Kasus Uji Validasi Olah Data Kandidat

Kasus uji validasi olah data kandidat oleh Administrator sebagai berikut pada tabel 5.7, tabel 5.8 dan tabel 5.9 :

Tabel 5.7 Kasus Uji Validasi Lihat Data Kandidat

Nama kasus uji	Lihat data kandidat
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas lihat data kandidat oleh Administrator
Prosedur uji	1. Administrator melakukan *login* 2. Administrator masuk ke halaman menu 3. Administartor masuk ke halaman kandidat
Hasil yang diharapkan	Sistem dapat menampilkan data kandidat sesuai dengan data kandidat yang terdaftar
Hasil yang didapatkan	Sistem menampilkan data kandidat sesuai dengan data kandidat yang terdaftar
Status Validalitas	Valid

Tabel 5.8 Kasus Uji Validasi Penambahan Data Kandidat

Nama kasus uji	Penambahan data kandidat
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas penambahan data kandidat oleh Administrator
Prosedur uji	1. Administrator melakukan *login* 2. Administrator masuk ke halaman menu 3. Administrator masuk ke halaman kandidat 4. Administrator menekan tombol tambah kandidat 5. Administrator menekan tombol submit

	6. Administrator melihat data kandidat yang baru saja di masukkan ke dalam daftar kandidat
Hasil yang diharapkan	Sistem dapat menyimpan data kandidat baru. Masuk ke halaman daftar kandidat, melihat data baru kandidat yang terisi
Hasil yang didapatkan	Sistem menyimpan data kandidat baru. Masuk ke halaman daftar kandidat, melihat data baru kandidat yang terisi
Status Validalitas	Valid

Tabel 5.9 Kasus Uji Validasi Hapus Data Kandidat

Nama kasus uji	Hapus data kandidat
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas hapus data kandidat oleh Administrator
Prosedur uji	1. Administrator melakukan *login* 2. Administrator masuk ke halaman menu 3. Administrator masuk ke halaman kandidat 4. Administrator menekan tautan *delete* pada tabel hapus
Hasil yang diharapkan	Sistem dapat melakukan penghapusan data kandidat yang dipilih
Hasil yang didapatkan	Sistem melakukan penghapusan data kandidat yang dipilih
Status Validalitas	Valid

5.1.3.5 Kasus Uji Validasi Olah Data *Result*

Kasus uji validasi olah data *result* oleh Administrator sebagai berikut pada tabel 5.10 :

Tabel 5.10 Kasus Uji Validasi Olah Data Hasil Pemilihan

Nama kasus uji	Lihat data *result*
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas lihat data hasil pemilihan oleh Administrator
Prosedur uji	1. Administrator melakukan *login* 2. Administrator masuk ke halaman menu 3. Administrator masuk ke halaman hasil pemilihan
Hasil yang diharapkan	Sistem dapat menampilkan data hasil *voting*
Hasil yang didapatkan	Sistem menampilkan data hasil *voting*
Status Validalitas	Valid

5.1.3.6 Kasus Uji Validasi Super Administrator

Kasus uji validasi Super Administrator untuk *generate* kunci RSA pada tabel 5.11 dan *log* pemilihan pada tabel 5.12 :

Tabel 5.11 Kasus Uji Validasi *Generate* Kunci RSA

Nama kasus uji	*Generate* kunci
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas *generate* kunci RSA
Prosedur uji	1. Super Administrator melakukan *login* 2. Super Administrator masuk ke halaman menu 3. Super Administrator masuk ke halaman *generate* kunci 4. Super Administrator meng-*generate* kunci RSA
Hasil yang diharapkan	Sistem dapat melakukan *generate* dan menghasilkan kunci RSA yaitu kunci publik dan kunci privat
Hasil yang didapatkan	Sistem melakukan *generate* dan menghasilkan kunci RSA (kunci publik dan kunci privat)
Status Validalitas	Valid

Tabel 5.12 Kasus Uji Validasi Lihat Data *Log* Pemilihan

Nama kasus uji	Lihat *log* pemilihan
Tujuan pengujian	Untuk menguji validitas kinerja dari sistem dalam menyediakan fasilitas lihat data *log* pemilihan
Prosedur uji	1. Super Administrator melakukan *login* 2. Super Administrator masuk ke halaman menu 3. Super Administrator masuk ke halaman *log* pemilihan 4. Super Administrator memiliki dan meng-upload *private key* 5. Super Administrator melihat data *log* pemilihan 6. Super Administrator men-*sorting* data *log* pemilihan
Hasil yang diharapkan	Sistem dapat melakukan dekripsi dengan *private key* pada data hasil *voting* kemudian hasilnya ditampilkan pada halaman *log* pemilihan
Hasil yang didapatkan	Sistem melakukan dekripsi dengan *private key* pada data hasil *voting* kemudian hasilnya ditampilkan pada halaman *log* pemilihan
Status Validalitas	Valid

5.2 Analisis

Proses analisi bertujuan untuk mendapatkan kesimpulan dari hasil pengujian sistem *e-voting* dengan jaminan *confidentiality* data pada *e-voting* menggunakan kriptografi kunci publik yaitu RSA yang telah dilakukan. Proses analisis mengacu pada dasar teori sesuai dengan hasil pengujian yang didapatkan. Analisis dilakukan terhadap hasil pengujian disetiap tahap pengujian. Proses analisis yang dilakukan meliputi analisis hasil pengujian sistem kriptografi, pengujian keamanan dan pengujian validasi.

5.2.1 Analisis Pengujian Sistem Kriptografi

Proses analisis terhadap pengujian sistem kriptografi dilakukan dengan melihat kesesuaian antara kinerja proses enkripsi dengan kebutuhan. Hasil pengujian sistem kriptografi pada data hasil *voting* telah berjalan sesuai dengan

analisa kebutuhan sistem. Dan dapat disimpulkan bahwa implementasi proses enkrispi menggunakan kriptografi kunci-publik yaitu RSA sudah berjalan dengan baik dan sesuai dengan kebutuhan yang ditentukan sebelumnya.

5.2.3 Analisis Pengujian Keamanan

Proses analisis terhadap hasil pengujian keamanan dilakukan dengan melihat kesesuaian antara kinerja mengamankan data hasil *voting* dengan kebutuhan. Hasil pengujian kemanan pada sistem *e-voting* berjalan sesuai dengan yang diinginkan. Manipulasi data yaitu perubahan data (*update*) hasil *voting* yang dilakukan oleh admin terdeteksi pada saat pencocokan data hasil *voting*.

5.2.3 Analisis Pengujian Validasi

Proses analisis terhadap hasil pengujian validasi dilakukan dengan melihat kesesuaian antara hasil kinerja sistem dengan daftar kebutuhan yang sudah ditentukan sebelumnya. Hasil pengujian validasi dengan metode *Black-box testing* telah berjalan sesuai dengan analisa kebutuhan sistem. Dan dapat disimpulkan bahwa fungsionalitas dalam sistem *e-voting* sudah berjalan dengan baik dan sesuai dengan kebutuhan yang di tentukan dalam analisa kebutuhan.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang dilakukan, maka diambil kesimpulan sebagai berikut :

1. Berdasarkan dari hasil pengujian validasi dengan metode *Black-box* testing menunjukan bahwa aplikasi telah berjalan sesuai rancangan yang telah dibangun.
2. Proses jaminan *confidentiality* data pada sistem *e-voting* telah berjalan dengan baik dan telah valid karena tidak ada perubahan data sebelum proses dan sesudah proses enkripsi dan dekripsi pada data hasil *voting* oleh *voter*.
3. Sistem *e-voting* ini telah di implementasikan sesuai dengan perancangan dan dapat digunakan untuk melakukan proses *voting* dan dapat menjamin *confidentiality* data hasil *voting.*

6.2 Saran

Saran yang diberikan untuk pengembangan penelitian selanjutnya, yaitu penelitian ini fokus pada unsur keamanan *confidentiality* yaitu jaminan kerahasiaan suatu data, maka untuk pengembangan selanjutnya dapat ditambahkan unsur keamanan data lainnya seperti *authentication* dan *non-repudiation.*

Daftar Pustaka

[ACO-96] Menezes, Alfred J., Paul C. van Oorschot, Scott A. Vanstone. 1996. *Applied Cryptography*. Diperoleh dari http://citeseerx.ist.psu.edu

[AJM-04] Kler Jeffrey dan Aleksandar Milojkovic. 2004. *An Analysis and Recommendations for an E-Voting System*. Diperoleh dari https://courses.ece.ubc.ca

[ALP-11] Rokhman, Ali. 2011. Prospek Dan Tantangan Penerapan E-*Voting* Di Indonesia. Diperoleh dari http://arokhman.blog.unsoed.ac.id

[CAN-11] Utama, Candra. 2011. *CodeIgniter Framework*. Diperoleh dari http://files.candrautama.com

[DWV-13] Chakole, J. B., P. R. Pardhi. 2013. *The Design of Web Based Secure Internet Voting System for Corporate Election*. Diperoleh dari http://www.ijsr.net

[EMR-09] Milanov, Evgeny. 2009. *The RSA Algorithm*. Diperoleh dari https://www.math.washington.edu

[EPC-11] Al-Anie, Hayam K., Mohammad A. Alia, Adnan A. Hnaif. 2011 *E-voting Protocol Based On Public-Key Cryptography*. Diperoleh dari http://airccse.org/journal/nsa/0711ijnsa08.pdf

[IDI-11] Daqiqil, Ibnu Id. 2011. *Framework CodeIgniter*. Diperoleh dari http://elib.unikom.ac.id

[ISE-11] Sommerville, Ian. 2011. *Software Engineering*. 9th Edition, Addison-Wesley, New York.

[KTE-10] Fahmi Husni dan Dwi Handoko. 2010. Kajian Teknis tentang Pemungutan Suara secara Elektronik (*Electronic Voting*). Diperoleh dari http://husnifahmi.com

[NSO-02] Chandra, Pravir; Matt Messier, John Viega. 2002. *Network Security with OpenSSL*. Diperoleh dari http://en.bookfi.org/book/649825

[MSE-10] Stenbro, Martine. 2010. *A Survey of Modern Electronic Voting Technologies*. Diperoleh dari http://www.diva-portal.org

[REV-20] Rivest, Ronald L. 2000. *Electronic Voting*. Diperoleh dari http://people.csail.mit.edu

[RPE-01] Pressman, Roger S. 2001. *Software Engineering A Practitioner's Approach*. 5th Edition. Boston: McGraw-Hill.

[RSK-04] Munir, Rinaldi. 2004. Sistem Kriptografi Kunci-Publik. Diperoleh dari http://informatika.stei.itb.ac.id

[SDI-03] Dharwiyanti, Sri dan Romi Satria Wahono. 2003. Pengantar Unified Modeling Language (UML). Diperoleh dari http://ikc.dinus.ac.id

[SKT-09] Kromodimoeljo, Sentot.2009. Teori dan Aplikasi Kriptografi. Diperoleh dari http://aqwamrosadi.staff.gunadarma.ac.id

[STE-11] SOLGM Electoral Working Party. 2011. *The Way Forward for E-voting in Local Government Elections and Polls*. Diperoleh dari http://www.solgm.org.nz