IMPLEMENTASI ALGORITMA *RIVEST SHAMIR ADLEMAN* DAN *INTERLOCK PROTOCOL* UNTUK SISTEM *E-VOTING*

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan pendidikan sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Komputer

oleh:

Aly Wafi

(152410101163)

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2022

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Suharis dan Ibunda Kustini;
2. Istriku Maulita Siswinarti;
3. Anakku Aulian Rafasya;
4. Sahabat-sahabatku bersama dukungan dan doanya;
5. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi;

MOTTO

“ “Menyerah hanyalah untuk orang yang kalah”

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Aly Wafi

NIM : 152410101163

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "IMPLEMENTASI ALGORITMA RIVEST SHAMIR ADLEMAN DAN INTERLOCK PROTOCOL UNTUK SISTEM E-VOTING"adalah benar-benar hasil karya saya sendiri, kecuali jika ada pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada instansi manapun, dan bukti karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika dikemudian hari pernyataan ini tidak benar.

Jember, 17 Juni 2022

Yang menyatakan,

Aly Wafi

NIM 152410101163

SKRIPSI

IMPLEMENTASI ALGORITMA RIVEST SHAMIR ADLEMAN DAN *INTERLOCK PROTOCOL* UNTUK SISTEM *E-VOTING*

oleh

Aly Wafi

NIM 152410101163

Pembimbing

Dosen Pembimbing Utama : Drs. Antonius Cahya P, M.App., Sc., Ph.D.

Dosen Pembimbing Anggota : Diksy Media Firmansyah S.Kom, M.Kom.

PENGESAHAN PEMBIMBING

Skripsi berjudul "IMPLEMENTASI ALGORITMA RIVEST SHAMIR ADLEMAN DAN INTERLOCK PROTOCOL UNTUK SISTEM E-VOTING", telah diuji dan disahkan pada:

Hari tanggal : Jumat, 27 Mei 2022

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,	Pembimbing II,
Drs. Antonius Cahya P, M.App., Sc., Ph.D	Diksy Media Firmansyah S.Kom, M.Kom.
NIP. 196909281993021001	NIP/NRP. 760016853

PENGESAHAN PENGUJI

Skripsi berjudul "IMPLEMENTASI ALGORITMA RIVEST SHAMIR ADLEMAN DAN INTERLOCK PROTOCOL UNTUK SISTEM E-VOTING", telah diuji dan disahkan pada:

Hari tanggal : Jumat, 17 Juni 2022

Tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Penguji I,	Penguji II,
Yanuar Nurdiansyah ST,.M.Cs.	Qurrota A'yuni Ar Ruhimat S.Pd., M.Sc.
NIP. 198201012010121004	NIP/NRP. 760018029

Mengesahkan
Dekan Fakultas Ilmu Komputer,

Prof. Saiful Bukhori,ST., M.Kom
NIP. 196811131994121001

RINGKASAN

E-voting merupakan suatu metode pemungutan suara dan penghitungan suara dalam suatu pemilihan dengan menggunakan perangkat elektronik. Untuk mengamankan data ketika proses e-voting berlangsung diperlukan sebuah kriptosistem. Kombinasi algoritma RSA dan interlock protocol akan digunakan dalam pengamanan proses e-voting. Hasil pilihan dari peserta voting akan dienkripsi menggunakan algoritma RSA, kemudian *ciphertext* akan dipecah menggunakan aturan interlock protocol sebelum ahirnya dikirim ke server untuk didekripsi dan dihitung hasil perolehan suaranya.

Untuk keamanan pada system E-voting diuji menggunakan simulasi serangan bruteforce dan man in the middle. Teknik bruteforce menghasilkan 153 percobaan dari 22200 kemungkinan yang ada dalam waktu 2.5 menit. Artinya hanya ada kemungkinan 0.64% untuk menebak *ciphertext* yang ada. Pemecahan *ciphertext* menjadi 2 bagian saat uji *man in the middle* mengharuskan peretas untuk menyusun *ciphertext* tersebut sebelum mendekripsinya, Dari hasil kedua percobaan tersebut system yang telah diimplementasi menggunakan algoritma RSA dan Interlock protocol dapat dikatakan aman.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul "IMPLEMENTASI ALGORITMA RIVEST SHAMIR ADLEMAN DAN *INTERLOCK PROTOCOL* UNTUK SISTEM *E-VOTING*". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Ayahanda Suharis, dan Ibunda Kustini yang selalu mendukung dan mendoakan proses pengerjaan skripsi;
2. Prof. Dr. Saiful Bukhori, ST.,M.Kom., selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
3. Drs. Antonius Cahya P, M.App., Sc., Ph.D selaku Dosen Pembimbing Utama dan Diksy Media Firmansyah S.Kom, M.Kom., selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, pikiran, dan perhatian dalam membantu penulisan skripsi;
4. Diah Ayu Retnani Wulandari S.T., M.Eng selaku Dosen Pembimbing Akademik (DPA), yang telah mendampingi penulisan skripsi;
5. Seluruh Bapak dan Ibu dosen di Program Studi Sistem Informasi Universitas Jember yang telah memberikan banyak ilmu;
6. Istriku Maulita Siswinarti yang telah membantu dan mendukung penulisan skripsi
7. Sahabat- sahabat saya di sekolah maupun di kampus yang telah mendukung dan tak henti-hentinya memberikan semangat ;

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB 1. PENDAHULUAN

Bab ini menjelaskan hal-hal yang berkaitan dengan pendahuluan penelitian. Adapun pembahasan pada bab ini meliputi latar belakang, rumusan masalah, tujuan dan manfaat, serta batasan masalah.

1.1 Latar belakang

Perkembangan teknologi yang begitu pesat merupakan suatu hal yang tidak dapat kita hindari lagi, beragam teknologi serta ilmu baru terus ditemukan untuk mempermudah pekerjaan. Era globalisasi telah mengantar perkembangan di berbagai aspek kehidupan manusia (Ahmad, 2012). Pada bidang musyawarah juga tak luput dari dampak perkembangan teknologi, seperti contoh kecil dalam voting. Voting merupakan pengambilan keputusan berdasarkan pemilih terbanyak. Umumnya voting dilakukan secara konvensional seperti menggunakan kertas dan perhitungan manual oleh panitia penyelenggara (Nabilah & Amrozi, 2019).

Voting inilah yang merupakan cikal bakal dari *E-voting*, *E-voting* masih mengandung konsep dan metode voting manual sebelumnya, perbedaannya hanya pada sistem *online* yang digunakan, sehingga dapat menghemat waktu dan biaya jika penggunaannya dalam skala besar. Pada Penelitian yang dilakukan oleh Andi Rosano pada tahun 2019 yang berjudul Penggunaan Aplikasi *e-voting* Berbasis *Decision Support Systems* pada Pilkades (Studi Kasus : Desa Kedungbanjar, Taman, Pemalang) mengungkapkan bahwa sebanyak 68% masyarakat mendukung dengan diadakannya sistem *E-voting* dalam pemilihan kepala desa . padahal hanya 7% masyarakat yang pernah mengenal komputer. Dengan ini kita mengetahui bahwa masyarakat sangat berantusias dalam penerapan voting secara online. Selain itu tidak adanya kriptografi pada *e-voting* tersebut menjadi sebuah kelemahan terbesar, mengingat banyaknya ancaman yang sering terjadi pada dunia maya. Kelemahan *E-voting* berasal dari teknologi yang mendasarinya sehingga sistem tersebut rentan terhadap serangan keamanan (Wijaya et al., 2019). Tentunya ancaman tersebut adalah seorang *hacker*, sehingga menuntut pemberlakuan keamanan yang ketat pada sistem *E-voting*.

Perubahan atau penyadapan data dapat terjadi apabila penyerang dapat bertindak sebagai *man in the middle* saat terjadi proses pertukaran data pada mesin voting (Satrya et al., 2015). Seperti contoh kasus *E-voting* tidak valid yang terjadi pada pemilu Riigikogu pada 26 Februari 2011. Salah satu analisis tentang kegagalan *E-voting* pada pemilu disebabkan adanya serangan *man in the middle* yang memodifikasi daftar kandidat asli yang dikirim ke IVCA (*i-voting client application*), sehingga berisi nomor calon yang tidak valid, hal itu terjadi karena VFS (*vote forwarding system*) tidak menandatangani secara digital daftar tersebut, sehingga hanya mengandalkan keamanan saluran HTTPS untuk integritas pesan, (Hutchison & Mitchell, 2011). Dari contoh kasus ini maka diperlukan sebuah algoritma kriptografi yang dapat menjamin keamanan pesan ketika dikirimkan pada server. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*) (Schneier dalam Karandikar, 2007).

Berdasarkan kuncinya algoritma kriptografi dibagi menjadi dua yaitu kriptografi simetris yang merupakan kriptografi dengan kunci enkripsi dan dekripsi sama, dan kriptografi asimetris yang memiliki kunci berbeda untuk enkripsi dan dekripsi. Menurut Basri (2016), pada algoritma asimetris Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki kunci *private* yang dapat melakukan pembongkaran terhadap sandi yang dikirim untuknya.

Salah satu dari algoritma kriptografi asimetris adalah *Rivest Shamir Adleman* (RSA), Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya. Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik) (Ginting et al., 2015). Akan tetapi menurut Rizal (2012), pelaku *man in the middle* dapat menyadap data dengan cara mengganti kunci publik yang telah dibangkitkan dengan kunci publik milik pelaku, sehingga ketika data dikirimkan kembali, pelaku dapat mendekripsi data tersebut dengan

kunci privat miliknya. dan mengenkripsi kembali dengan kunci yang telah didapatkan sebelumnya, sehingga sistem akan berjalan normal dan seolah tidak ada yang terjadi. Problem ini dapat dicegah dengan menerapkan *Interlock Protocol.* Algoritma dasar dari *interlock protocol* adalah bahwa ia mengirimkan dua bagian dari pesan terenkripsi. Bagian pertama dapat merupakan hasil dari *hash* pesan satu arah, dan bagian kedua adalah pesan terenkripsi itu sendiri. Ini mencegah penyadap mendekripsi pesan dengan kunci pribadi yang dimiliki (Rizal, 2012). Penelitian ini dilaksanakan untuk menginvestigasi bagaimana implementasi dari algoritma RSA dan *Interlock Protocol* pada sebuah aplikasi *E-voting* serta tingkat keamanannya dalam menjaga kerahasiaan pesan yang dikirim oleh *client* kepada server ketika proses voting berlangsung.

1.2 Rumusan masalah

Dalam penelitian ini dapat dirumuskan permasalahan - permasalahan yang terjadi sebagai berikut:

1. Bagaimana implementasi Algoritma *Rivest Shamir Adleman* dan *Interlock Protocol* pada sistem *E-voting*?
2. Bagaimana tingkat keamanan dari penggunaan Algoritma *Rivest Shamir Adleman* dan *Interlock Protocol* pada sistem *E-voting*?

1.3 Batasan masalah

Adapun Batasan dalam penelitian ini adalah:

1. Aplikasi *Client* Server berbasis web.
2. Penggunaan Algoritma RSA dan *Interlock Protocol* hanya dikhususkan untuk mengatasi serangan *man in the middle attack* pada proses pengiriman data antara *client* dan server saat *e-voting* berlangsung.
3. *E-Voting* yang dicontohkan merupakan E-Voting pemilihan Kepala Desa Sumberwaru, Banyuputih, Situbondo.
4. Diasumsikan akun untuk peserta e-voting (*username* dan *password*) hanya dimiliki oleh peserta.

1.4 Tujuan penelitian

Tujuan yang diperoleh dari penelitian ini adalah :

1. Mengetahui implementasi dari Algoritma *Rivest Shamir Adleman* dan *Interlock Protocol* pada sistem *E-voting*.
2. Mengetahui tingkat keamanan pada pesan yang dikirim oleh *client* kepada server ketika proses voting berlangsung setelah terimplementasi Algoritma *Rives Shamir Adleman* dan *Interlock Protocol*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Bagi Akademis

   Hasil penelitian yang dilakukan dapat membuktikan adanya peningkatan keamanan pada proses pengiriman data yang menerapkan *Interlock Protocol* dan algoritma RSA.

2. Bagi Peneliti

   Dapat mengetahui dan menerapkan *Interlock Protocol* dan algoritma RSA untuk meningkatkan keamanan pada sebuah sistem.

BAB 2. TINJAUAN PUSTAKA

Bagian ini memaparkan tinjauan yang berkaitan dengan masalah yang dibahas, serta kajian teori yang dikaitkan dengan permasalahan yang dihadapi. Teori yang diadopsi nantinya akan digunakan sebagai landasan dalam penulisan untuk menyelesaikan penelitian ini.

2.1 Penelitian Terdahulu

Aplikasi Pemungutan Suara Elektronik/*E-voting* pernah diteliti oleh Risnanto (2018) menggunakan teknologi *Short Message Service* dan *At Command*, Penelitian tersebut membahas tentang penggunaan *SMS* yang dikendalikan oleh *AT Command* dalam pengumpulan suara yang dikirim oleh peserta *E-voting*. Hasil dari penelitian tersebut adalah terselenggaranya *E-voting* secara online yang tentunya dengan efisiensi yang lebih dari voting yang diselenggarakan secara manual, akan tetapi pada penelitian tersebut tidak disebutkan tentang keamanan yang digunakan pada sistem tersebut. Dari hasil penelitian di atas akan diadopsi sistem *E-voting* yang sama namun dengan platform yang berbeda kemudian pada penelitian ini juga akan ditambahkan Algoritma RSA dan *Interlock protocol* untuk menambah keamanan pada sistem.

Ridwan dkk (2016) menggunakan Algoritma *RSA* dalam rancang bangun sistem *E-voting* yang berbasis WEB dengan hasil penelitian 100 % data tidak mengalami perubahan saat pengiriman. Hal yang diadopsi dalam penelitian ini adalah Algoritma RSA yang kemudian akan dikombinasikan dengan *Interlock Protocol* untuk menghindari serangan *man in the middle*.

Penelitian yang dikembangkan oleh Yusmiarti (2020) dengan judul *E-voting* pemilihan kepala desa berbasis android, penelitian tersebut menggunakan server dengan platform web dan *client* menggunakan android. Platform server diadopsi pada penelitian ini akan tetapi *client* juga akan menggunakan platform web pada penelitian ini agar *client* lebih mudah untuk mengakses situs *E-voting*. Perbedaan yang sangat mencolok dengan penelitian ini terletak pada sisi keamanan di mana penelitian tersebut tidak menggunakan algoritma maupun metode untuk mengamankan transaksi data antara *client* dan server, sedangkan pada penelitian ini menggunakan Algoritma *RSA* dan *Interlock Protocol*.

Penelitian terakhir yang dirujuk dilaksanakan oleh Rizal (2012) dengan judul Perancangan Simulasi *Man In The Middle Attack* Pada Algoritma Kriptografi RSA Dan Pencegahannya Dengan *Interlock Protocol*. Dalam simulasi penelitian tersebut diperoleh bahwa penyadap tidak dapat mendapatkan pesan yang diinginkan walaupun telah mengganti kunci publik yang digunakan *client*. Selain itu keberadaan penyadap dapat diketahui dengan peringatan perbedaan *hash* yang dikirim penyadap pada server. Persamaan penelitian tersebut dengan penelitian ini adalah dalam metode keamanan yang digunakan, perbedaannya adalah penelitian tersebut menggunakan aplikasi *dekstop* untuk menyimulasikan penyerangan *man in the middle* sedangkan pada penelitian ini menggunakan web untuk Server dan *client*. Dengan hasil penelitian sebelumnya yang sukses diharapkan dengan metode yang sama dapat dihasilkan keamanan yang serupa walaupun dengan platform yang berbeda.

2.2 Electronic voting

Electronic voting (E-voting) adalah suatu metode pemungutan suara dan penghitungan suara dalam suatu pemilihan dengan menggunakan perangkat elektronik. Tujuan dari E-voting adalah menyelenggarakan pemungutan suara dengan biaya hemat dan penghitungan suara yang cepat dengan menggunakan sistem yang aman dan mudah untuk dilakukan audit. Dengan E-voting perhitungan suara akan lebih cepat, bisa menghemat biaya pencetakan surat suara, pemungutan suara lebih sederhana, dan peralatan dapat digunakan berulang kali untuk Pemilu dan Pilkada.

Penerapan E-voting juga harus menerapkan asas-asas pemilu yang telah diatur dalam UU Nomor 7 Tahun 2017. Berdasarkan UU Nomor 7 Tahun 2017 tentang pemilihan umum, asas-asas tersebut terdiri dari 6 hal :

1. Langsung

   Asas langsung mengandung makna bahwa pemilih memiliki hak secara langsung tanpa melalui perantara, dalam penerapannya pada sistem E-voting pengguna dapat mengakses langsung menggunakan *username* dan *password* yang telah disediakan.
2. Umum

Asas umum yakni memberikan kesempatan bagi semua warga negara untuk berpartisipasi dalam pemilihan tanpa adanya diskriminasi, dalam penerapannya pada sistem E-voting pengguna akan memiliki akun dengan level dan hak guna yang sama antara satu dengan lainnya.

3. Bebas

   Asas bebas adalah pemilih bebas menentukan pilihannya tanpa adanya paksaan dari siapa pun, dukungan dari sistem E-voting untuk asas ini adalah dengan menampilkan deksripsi dari setiap Topik (Calon) dengan tampilan dan format yang sama, sehingga tidak ada kesan memberatkan terhadap satu Topik atau Calon.

4. Rahasia

   Asas rahasia berupa jaminan untuk merahasiakan pilihan suara, dalam E-voting Asas rahasia merupakan asas yang sangat diunggulkan karena asas tersebut ditopang dengan implementasi Algoritma untuk mengamankan data hasil pilihan, selain itu *super user* (Admin), juga tidak dapat mengetahui pilihan dari pemilih.

5. Jujur

   Asas jujur berlaku untuk panitia, pengawas maupun pemilih yang harus bersikap jujur, dalam kasus E-voting pemberlakuan *multi user* (Admin dan petugas) pada sistem digunakan untuk *monitoring* satu dan lainnya sehingga meminimalisir kecurangan yang berlaku.

6. Adil

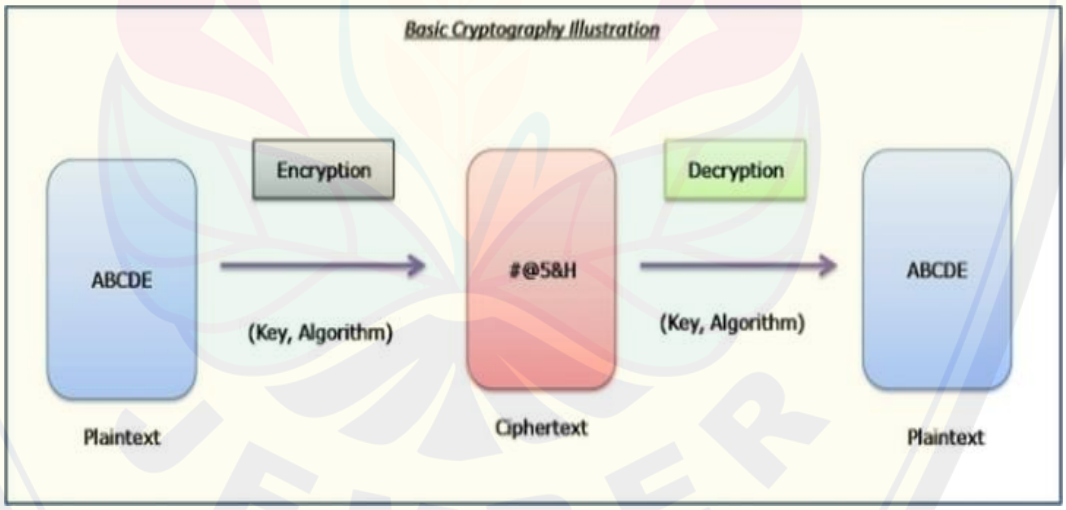
   Asas adil berarti setiap pemilih berhak mendapatkan perilaku yang sama, dalam E-voting semua pemilih akan mendapatkan akun dengan level dan hak yang sama.

Memang E-voting bukanlah sebuah hal yang baru, India dan Amerika Serikat telah melakukan Pemilu Parlemen dengan E-voting. Tidak hanya di luar negeri, di Bali, tepatnya di Kabupaten Jembrana, telah dilakukan puluhan kali E-voting untuk Pemilihan Kepala Dusun (Iskandar, 2010). Kedua e-voting tersebut menggunakan platform web, namun untuk Algoritma keamanan yang digunakan tidak disebutkan.

2.3 Kriptografi

Kriptografi berasal dari bahasa Yunani, yang terdiri dari dua suku kata yaitu *kripto* dan *graphia*. *Kripto* artinya menyembunyikan, dan *graphia* artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik matematika yang berhubungan dengan keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta otentikasi data. Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak yang tidak diinginkan. Untuk menjaga pesan, maka pesan tersebut dapat diubah menjadi sebuah kode yang tidak dapat dimengerti pihak lain .

Algoritma utama dalam kriptografi adalah enkripsi dan dekripsi, enkripsi adalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang bisa dimengerti (*plaintext*) menjadi sebuah kode yang tidak bisa dimengerti (*ciphertext*). Sedangkan dekripsi adalah proses untuk mengubah *cipertext* menjadi *plaintext*. Proses enkripsi dan deskripsi memerlukan suatu mekanisme dan kunci tertentu (Kurniawan et al., 2018).



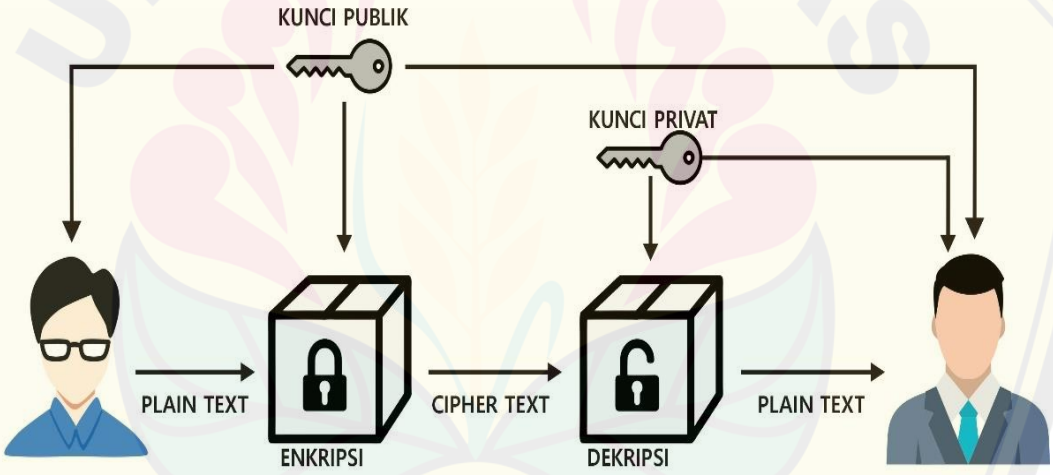
Gambar 2.1 Ilustrasi Dasar Kriptografi

Untuk menghasilkan sebuah kunci dalam kriptografi dibutuhkan sebuah algoritma khusus. Berdasarkan kunci enkripsi dan dekripsinya algoritma kriptografi dikelompokkan menjadi 2 jenis, yaitu kriptografi simetris dan kriptografi asimetris (Basri, 2016). Kriptografi simetris merupakan Algoritma kriptografi yang memiliki

kunci enkripsi dan dekripsi sama, sedangkan kriptografi asimetris adalah Algoritma kriptografi yang memiliki kunci enkripsi dan dekripsi yang berbeda. Contoh kriptografi kunci publik antara lain RSA, El Gamal dan rabin.

2.4 Algoritma Rivest Shamir Adleman (*RSA*)

RSA merupakan algoritma kriptografi asimetris, dimana kunci yang digunakan untuk mengenkripsi berbeda dengan yang digunakan untuk mendekripsi. Kunci yang digunakan untuk mengenkripsi disebut dengan kunci publik, dan yang digunakan untuk mendekripsi disebut dengan kunci privat. RSA membutuhkan tiga langkah dalam prosesnya, yaitu pembangkitan kunci, enkripsi, dan dekripsi. Proses enkripsi dan dekripsi merupakan proses yang hampir sama. Jika bilangan acak yang dibangkitkan kuat, maka akan lebih sulit untuk melakukan *cracking* terhadap pesan. Parameter kuat tidaknya suatu kunci terdapat pada besarnya bilangan acak yang digunakan. (Zwaini, 2013).



Gambar 2.2 Ilustrasi Alur Penerapan Algoritma RSA

Ilustrasi pada Gambar 2.2 menggambarkan bahwa setelah proses pembangkitan kunci maka kunci publik dapat dimiliki oleh semua orang yang akan digunakan untuk proses enkripsi *ciphertext*. Kunci privat hanya akan dimiliki oleh orang tertentu saja (dalam penelitian ini adalah server), sehingga hanya serverlah yang dapat mendekripsi *ciphertext*.

Proses Pembangkitan Kunci :

1) Menentukan p dan q (harus prima keduanya).
2) Menghitung nilai $n. n = p. q$

3) Menghitung nilai $\emptyset(n)$. $\emptyset(n) = (p-1)(q-1)$
4) Menentukan *e*. Nilai *e* relatif prima terhadap $\emptyset(n), fpb\ (e, \emptyset(n)) \equiv 1$
5) Menghitung nilai $d. d = e^{-1} mod\ \emptyset(n)$

Perhitungan di atas akan menghasilkan pasangan kunci publik e dan n, dan pasangan kunci *private* d dan n.

Proses Enkripsi

Proses enkripsi dilakukan menggunakan rumus $c = m^e mod\ n$.

Proses Dekripsi

Setiap blok *ciphertext* didekripsi kembali menjadi blok *plaintext* dengan rumus $m = c^d mod\ n$.

Keterangan :

$c =$ *Ciphertext.*

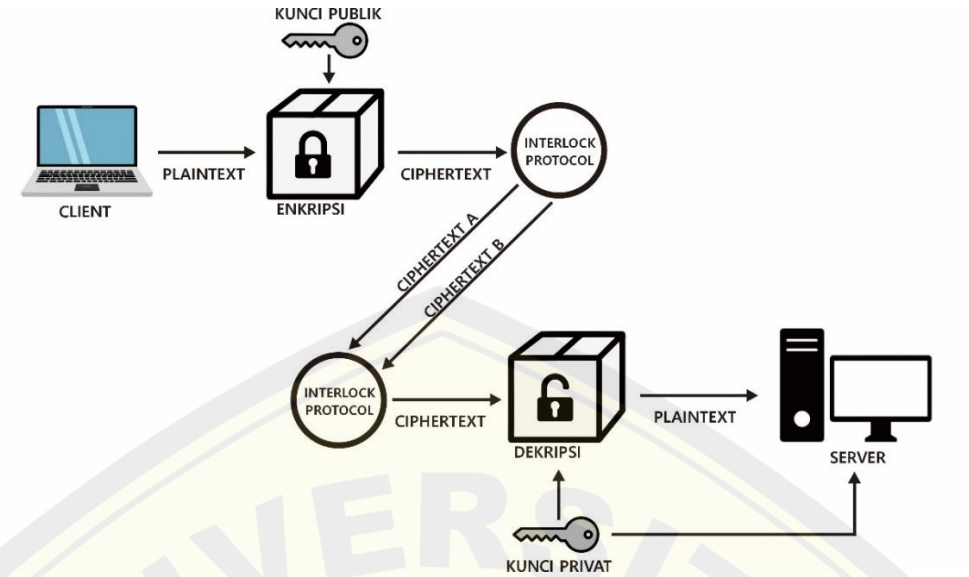
$m =$ *Plaintext* (pesan).

$e, n =$ Pasangan kunci publik.

$d, n =$ Pasangan kunci publik.

2.5 Interlock Protocol

Interlock Protokol dibuat oleh Ron Rivest dan Adi Shamir. Algoritma dasar dari protokol ini adalah untuk mengirim dua bagian dari pesan terenkripsi. Bagian pertama dapat merupakan hasil dari hash pesan satu arah, dan bagian kedua adalah pesan terenkripsi itu sendiri. Ini mencegah mata-mata dari mendekripsi pesan pertama dengan kunci pribadinya. Dia hanya bisa menulis pesan baru dan mengirimkan kepada penerima. (Ritonga, 2017). Secara singkat, cara kerja *Interlock Protocol* yang dikombinasikan dengan RSA dapat dilihat pada Gambar 2.3:



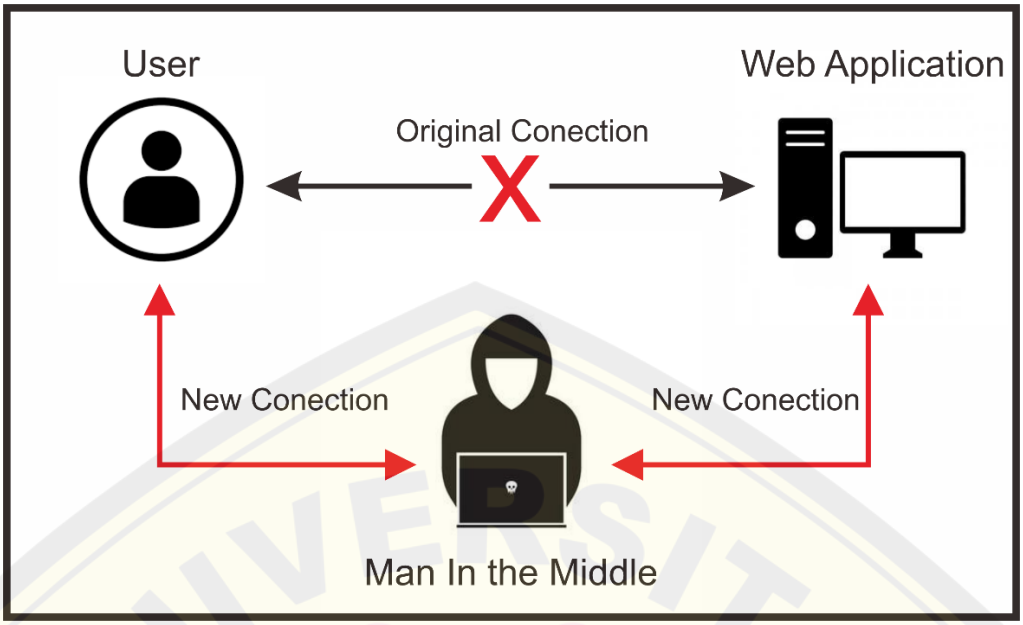
Gambar 2.3 Alur RSA yang kombinasikan dengan Interlock Protocol

Client akan mengenkripsi *plaintext* menjadi *ciphertext*, setelah itu ciphertext akan dipecah menjadi dua bagian menggunakan kaidah interlock *protocol*, sehingga data yang dikirimkan pada server berupa dua *ciphertext*. Kedua *ciphertext* tersebut akan digabungkan ulang sesuai aturan *interlock protocol* yang berlaku untuk kemudian dilakukan proses dekripsi sehingga didapatkan data yang bermakna.

2.6 Ancaman Keamanan pada Sistem E-voting

Man in the midle (MITM) merupakan jenis penyerangan di mana penyerang melakukan pembacaan data atau pengiriman data antara dua objek yang saling berkomunikasi satu sama lain *Man in the middle* sendiri jika diartikan ke bahasa Indonesia berarti seseorang yang berada di tengah yang berarti posisi penyerang berada di antara target dan sumber data. Tujuan dari teknik MITM adalah mencuri data informasi personal, data log in, data akun dan kartu kredit.

Teknik MITM bekerja dengan cara mengelabui arus data antara client dan server. Jika dalam keadaan normal *user* langsung berkomunikasi dengan server tetapi jika terdapat penyerangan MITM penyerang akan melihat arus data yang dikirim oleh *client* ke server dan sebaliknya. skema penyerangan dapat dilihat seperti pada Gambar 2.4.



Gambar 2.4. alur kerja *man in the middle attack*

Penggunaan *man in the middle* berdampak kepada kerahasiaan data yang dikirim dalam satu jaringan. Data yang dikirim atau diterima oleh pengguna dibaca oleh penyerang. Penggunaan enkripsi sangat diperlukan untuk menangkal penyerangan jenis ini. Keamanan dalam pengiriman data dan informasi tidak hanya bergantung pada faktor kuatnya algoritma kriptografi yang digunakan pada pesan, tapi juga pada jalur informasi yang dilewati. Bila jalur informasi tersebut mampu disadap, penyerangan lebih lanjut dapat dilakukan pada pesan yang telah terenkripsi baik dalam hal keutuhan pesan maupun kebenaran pesan.

BAB 3. METODOLOGI PENELITIAN

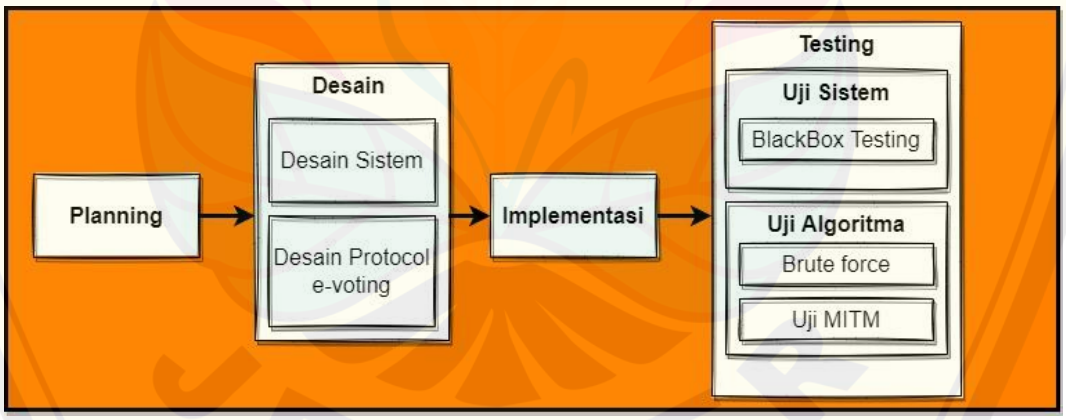
Pada bab ini akan membahas jenis penelitian, tahapan penelitian yang digunakan dalam pembuatan implementasi Algoritma RSA dan *Interlock protocol* dalam sistem *E-voting*.

3.1 Jenis Penelitian

Penelitian ini tergolong pada jenis penelitian pengembangan. Penelitian pengembangan adalah aktivitas riset dasar yang bertujuan mendapatkan informasi untuk dikembangkan. Hasil dari penelitian pengembangan adalah sebuah produk yang telah diuji efektivitasnya. Dalam penelitian akan mengimplementasikan sekaligus menguji tingkat keamanan dari Algoritma RSA dan *Interlock Protocol* pada aplikasi *E-voting*.

3.2 Tahapan Penelitian

Tahapan Penelitian yang akan dilakukan dapat dilihat dalam diagram alur di bawah ini. Penelitian ini terdiri dari 4 tahap mulai dari planing, desain, implementasi sampai tahap testing. Tahap-tahap ini harus di lakukan secara urut karena tahap sebelumnya berpengaruh ke tahap selanjutnya.



Gambar 3.1 Tahapan Penelitian

3.2.1 Planning (Perencanaan)

Kegiatan Pembuatan sistem dilakukan untuk membuat aplikasi *E-voting* sederhana. Pembuatan aplikasi ini menggunakan SDLC *Extreme Programming* (XP). *Extreme Programming (XP)* merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari

metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan *requirement* yang tidak jelas maupun terjadi perubahan–perubahan *requirement* yang sangat cepat (Carolina & Supriyatna, 2019). Tahapan ini merupakan langkah awal dalam pembangunan sistem di mana dalam tahapan ini akan disusun kebutuhan fungsional dan kebutuhan non-fungsional.

3.2.2 Desain

Tahapan berikutnya adalah desain (perancangan) di mana pada tahapan ini dilakukan kegiatan desain sistem dan desain protokol e-voting. desain protokol e-voting akan mengimplementasikan Algoritma RSA dan *Interlock Protocol* pada alur voting.

3.2.2.1 Desain Sistem

Desain Sistem akan mengidentifikasi komponen-komponen yang dibutuhkan sistem. Komponen tersebut berupa :

a) *Use Case* Diagram

Use Case diagram dibuat untuk menggambarkan hubungan antara pengguna dengan sistem yang dibuat.

b) *Activity* Diagram

Activity diagram digunakan untuk mengetahui runtutan dari proses pada sistem.

c) *Deployment* Diagram

Deployment diagram digunakan untuk memvisualisasikan hubungan antara software dan hardware yang digunakan dalam sistem.

d) *Logical Record Structure*.

Logical record structure digunakan dalam perancangan database untuk menunjukkan struktur *record* pada tabel.

3.2.2.2 Desain Protocol E-voting

Algoritma *RSA dan interlock protocol* akan digunakan untuk mengamankan data yang calon yang telah pengguna pilih. Dengan adanya kombinasi 2 algoritma ini diharapkan data yang dikirim tidak mudah terbaca oleh peretas. Pembangkitan kunci publik RSA dilakukan sejak data pengguna ditambahkan oleh admin, kemudian kunci publik tersebut akan digunakan untuk mengenkripsi pilihan

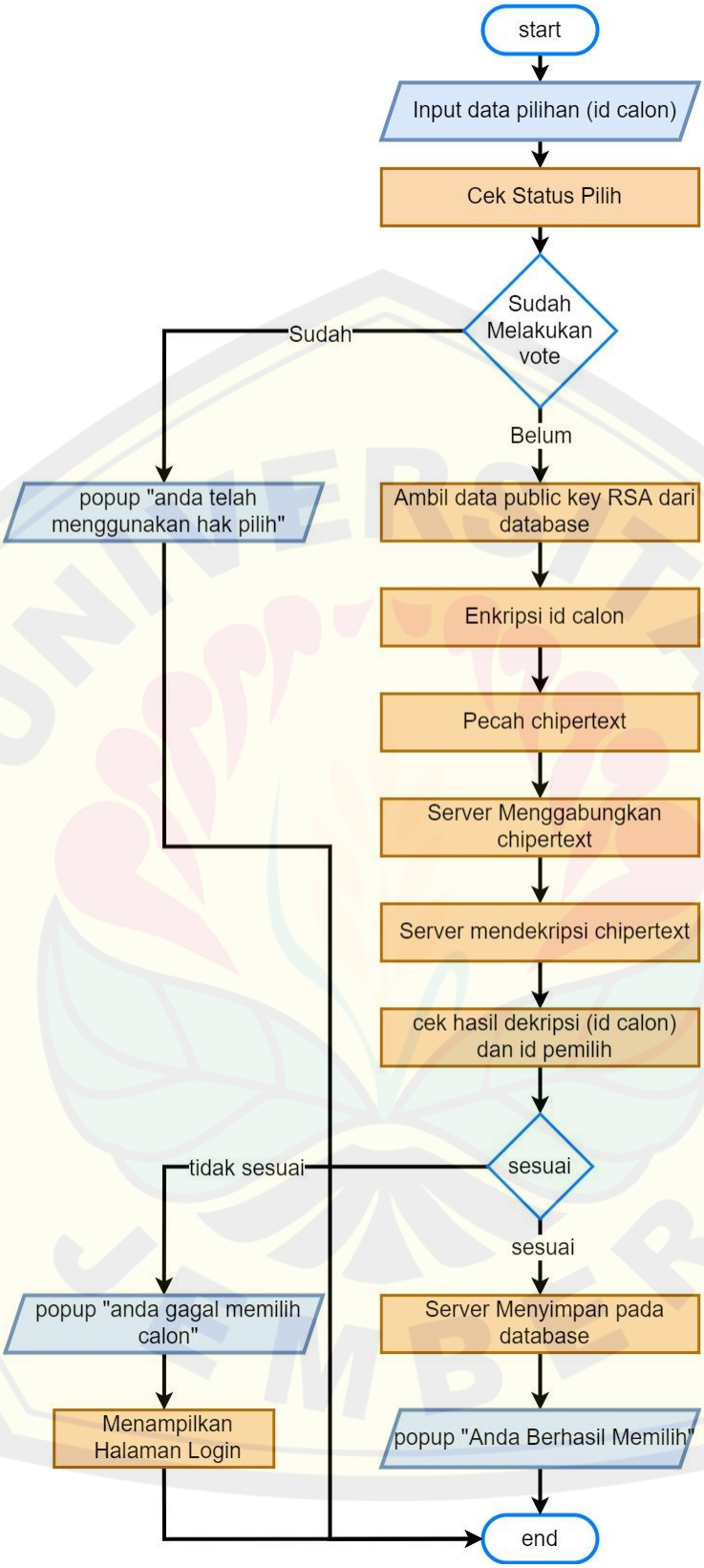
pengguna saat melakukan voting sehingga data tersebut akan berubah menjadi *ciphertext*, kemudian *ciphertext* akan dipecah menjadi 2 bagian sesuai kaidah interlock protocol , *ciphertext* yang telah terpecah menjadi 2 akan dikirimkan ke server, sehingga Ketika terjadi ancaman dari pihak luar, peretas masih harus menyusun *ciphertext* sebelum mendekripsinya. Server akan menerima kedua data tersebut dan kemudian Menyusun *ciphertext* tersebut lalu akan mendekripnya menggunakan kunci privat yang server miliki (lihat Gambar 3.2).

3.2.3 Implementasi

Tahapan ini merupakan kegiatan penerapan pemodelan yang sudah dibuat ke dalam bentuk *user inteface* dengan menggunakan bahasa pemrograman. Adapun bahasa pemrograman yang digunakan adalah *php*, *java script* dan *html*. Untuk sistem manajemen basis data menggunakan piranti lunak MySQL.

3.2.4 Testing

Setelah tahapan implementasi selesai, kemudian dilakukan tahapan pengujian untuk mengetahui kesalahan apa saja yang timbul saat aplikasi sedang berjalan serta mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Uji yang pertama adalah uji sistem yang mengetahui kesesuaian fungsional sistem, kemudian adalah uji algoritma untuk mengukur tingkat keamanan dari Algoritma RSA dan *Interlock Protocol*.



Gambar 3.2 Flowchart Penggunaan RSA dan *Interlock Protocol*

3.2.4.1 Uji Sistem

Pengujian sistem akan dilakukan menggunakan metode *Black Box*. *Black box* testing adalah pengujian yang dilakukan terhadap seluruh fungsional fitur apakah berjalan sesuai dengan fungsi masing-masing.

3.2.4.2 Uji Algoritma

Uji Algoritma dilakukan menggunakan simulasi serangan *brute force* dan *man in the middle*. Simulasi serangan *bruteforce* bertujuan untuk mengetahui ketahanan algoritma dalam menerima serangan secara paksa dengan cara menebak kombinasi angka, kemudian simulasi serangan *man in the middle* ditujukan untuk mengantisipasi penyadapan saat pengiriman data.

a. Uji *Brute Force*

Brute force merupakan metode serangan lama dan sederhana yang memiliki persentase sukses yang sangat tinggi dan dinilai sangat efektif. Cara kerja *brute force* adalah memaksa mengakses jaringan dengan menebak data yang diinginkan (dalam penelitian ini adalah *ciphertext*). Pada praktiknya *Brute force* memiliki beberapa metode yang sering digunakan seperti metode sederhana, metode kamus (*dictionary attack*) dan *hybrid Brute force*. Metode sederhana atau *Simple Brute force attack* merupakan metode yang akan digunakan dalam penelitian ini karena seluruh kemungkinan berupa urutan angka.

Uji *brute force* dalam penelitian ini akan dilakukan menggunakan *software* Burpsuite. Burpsuite akan memotong jalur komunikasi dan dapat melihat data yang yang dikirim oleh pemilih berupa *ciphertext* yang telah dipecah menjadi 2 bagian, kemudian Burpsuite akan menebak seluruh kombinasi angka agar sesuai ketika didekripsi oleh server.

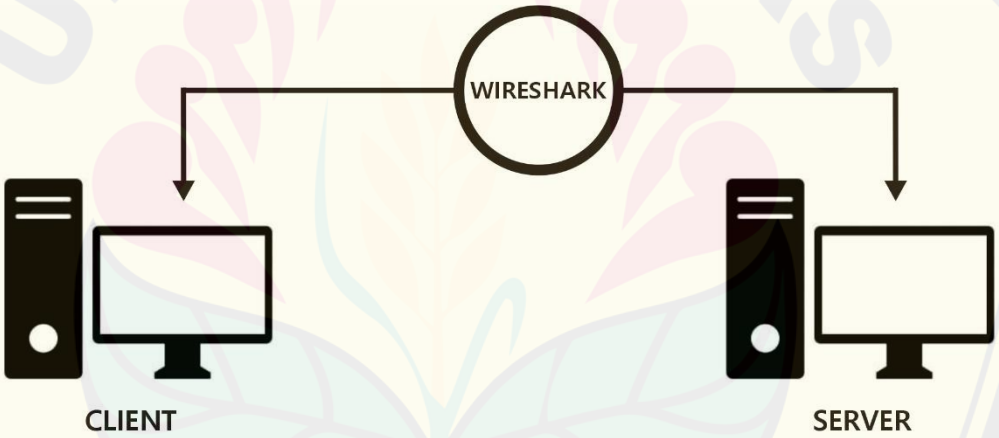
Hasil pilihan *e-voting* akan dienkripsi menjadi *ciphertext* yang terdiri dari 6 digit bilangan bulat yang dipecah menjadi 2 bagian masing-masing 2 dan 4 digit, sehingga Burpsuite akan menebak seluruh kemungkinan kombinasi angka dari kedua blok angka tersebut. apabila Burpsuite mengirimkan kode yang berhasil maka server akan menampilkan *pop-up* “Anda berhasil memilih” karena *ciphertext* terdekripsi dengan benar dan sesuai dengan data Id_Calon pada database, akan tetapi jika salah, sistem akan menampilkan *pop-up* “Anda gagal memilih”.

Pengujian *bruteforce* dilakukan dengan spesifikasi komputer :

1. Prosesor : Intel Core i7-6500U (4 CPus) 2.6 GHz.
2. Sistem Operasi : Windows 7 Professional.
3. Ram : 8 GB DDR4 .
4. Grafis : NVIDIA GeForce GT 940MX 2 GB.
5. Penyimpanan : 256Gb SSD.

b. Uji MITM

Cara kerja *Man in the Middle* (MITM) adalah merekam lalu lintas data yang pada situs web. Uji MITM menggunakan *software* Wireshark yang berfungsi membaca lalu lintas data antara *client* dan server, sehingga dalam penelitian ini ketika pemilih melakukan voting maka data pilihan akan terekam oleh MITM, MITM akan membaca data hasil voting yang pengguna kirimkan kepada server yang pasti berupa *ciphertext*. Skema pengujian dapat dilihat pada Gambar 3.3.



Gambar 3.3 Skema pengujian MITM

BAB 4. HASIL DAN PEMBAHASAN

4.1 Planning

Tahap *planning* menjelaskan tentang kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan fungsional dan non-fungsional nantinya akan dijadikan landasan untuk pembuatan desain sistem.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional meliputi proses yang dilakukan oleh sistem. kebutuhan fungsional juga mencakup informasi yang perlu ada dan dihasilkan oleh sistem .

1. Sistem menyajikan fitur kelola data pengguna (admin, pemilih dan petugas), sehingga admin dan petugas dapat menambah, mengubah dan menghapus data pengguna.
2. Sistem menyajikan data kandidat, sehingga admin dan petugas dapat menambah, mengubah dan menghapus data kandidat.
3. Sistem dapat mengelola data pemilih, sehingga admin dan petugas dapat menentukan siapa saja yang akan menjadi peserta e-voting.
4. Sistem dapat menyajikan data e-voting (data kandidat dan hasil e-voting) secara cepat dan akurat.
5. Sistem dapat menentukan waktu untuk menampilkan hasil e-voting.

Dari kebutuhan fungsional tersebut dapat disimpulkan fitur yang didapatkan oleh setiap pengguna, dapat dilihat pada Tabel 4.1.

Tabel 4.1 User dan fitur

User	Keterangan
Admin	Admin adalah orang yang dapat mengelola data kandidat, mengelola data pemilih dan mengelola *user* yang ada pada sistem, mengatur waktu untuk menampilkan hasil *e-voting.*
Petugas	Petugas adalah orang yang dapat mengelola data kandidat, mengelola data pemilih, mengatur waktu untuk menampilkan hasil *e-voting*..
Pemilih	Pemilih adalah orang yang hanya dapat log in dan melakukan *e-voting* pada pilihan yang dikehendaki.

4.1.2 Kebutuhan non-fungsional

Kebutuhan non-fungsional termasuk masalah yang terkait dengan pembatasan pengguna. Batasan tersebut berupa batasan waktu maupun perangkat lunak yang dapat digunakan.

1. Sistem hanya dapat dijalankan pada web browser, baik melalui komputer ataupun *smartphone*.
2. Seluruh pengguna aplikasi dapat melihat hasil e-voting tanpa harus log in.
3. Sistem harus dapat menjamin keamanan aliran data saat e-voting berlangsung.
4. Sistem harus mudah dipahami oleh pemilih e-voting.
5. Hasil e-voting hanya ditampilkan sesuai dengan waktu yang telah ditentukan.

4.2 Desain

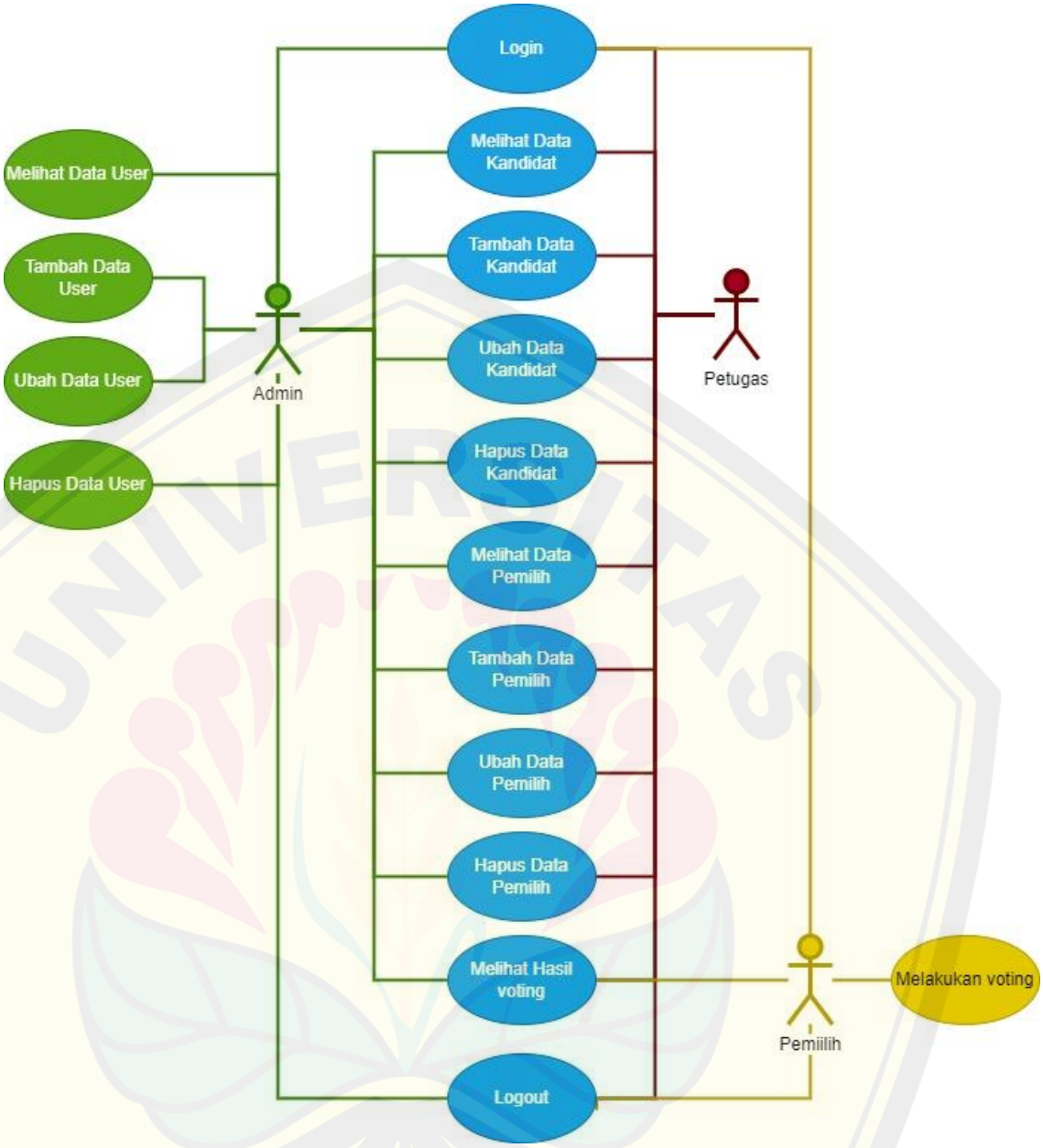
Tahap ini berisi pembuatan komponen-komponen yang dibutuhkan dalam pembuatan sistem termasuk pembuatan desain protokol e-voting. pembuatan desain ditujukan untuk memberikan gambaran saat proses *coding* (pengkodean) dilakukan.

4.2.1 Desain Sistem

Desain sistem meliputi pembuatan *usecase, activity diagram, deployment diagram,* dan *logical record structure.* Desain ini ditujukan untuk menggambarkan sistem yang akan dibuat serta untuk lebih memahami alur dari sistem yang akan dibuat.

4.2.1.1 Use Case Diagram

Pembuatan *use case diagram* dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Hasil representasi dari skema tersebut dibuat secara sederhana dan bertujuan untuk memudahkan pengguna dalam membaca informasi yang diberikan. *Usecase* diagram dapat dilihat pada Gambar 4.1.

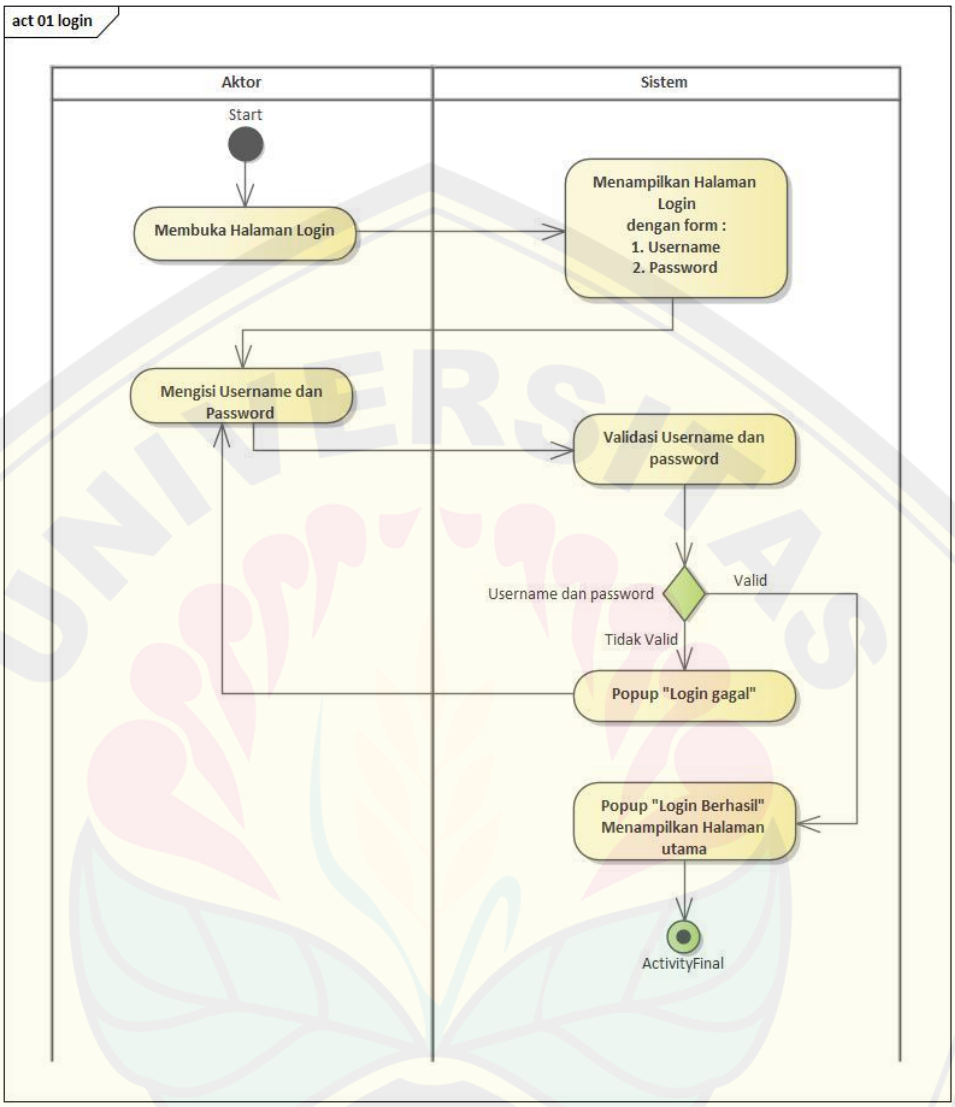


Gambar 4.1 *Use Case Diagram*

Admin adalah aktor utama dalam sistem ini yang dapat mengakses seluruh fitur dan mengelola seruh aktor. Aktor kedua adalah kandidat yang memiliki kewenangan dalam mengatur seluruh proses berjalannya e-voting. Kemudian pemilih merupakan aktor yang hanya dapat melakukan e-voting dan melihat hasilnya.

4.2.1.2 Activity Diagram

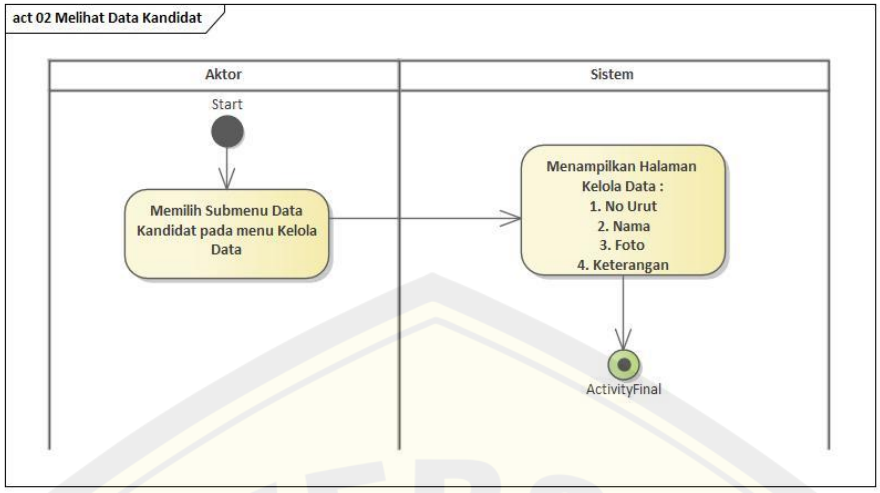
1. Log in



Gambar 4.2 *Activity Diagram* Log in

Aktor akan membuka halaman log in lalu mengisi *username* dan *password*, kemudian sistem akan memvalidasi *username* dan *password*. Apabila *username* dan *password* benar maka aktor akan masuk ke halaman utama, dan apabila salah maka aktor akan tetap berada pada halaman log in.

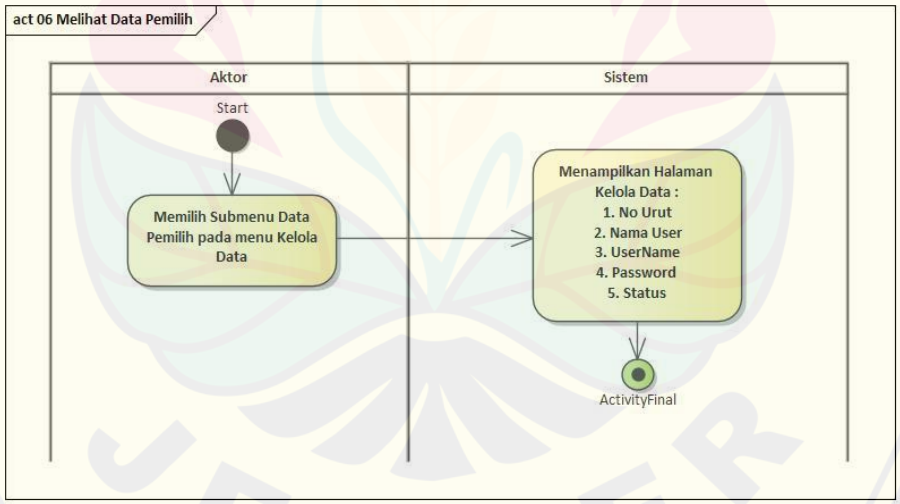
2. Melihat data kandidat



Gambar 4.3 *Activity Diagram* Melihat Data Kandidat

Aktor yang berada pada halaman utama akan memilih submenu data kandidat, kemudian sistem akan menampilkan data kandidat/calon. Data yang ditampilkan akan disajikan dalam bentuk tabel dengan beberapa tombol kelola di samping data.

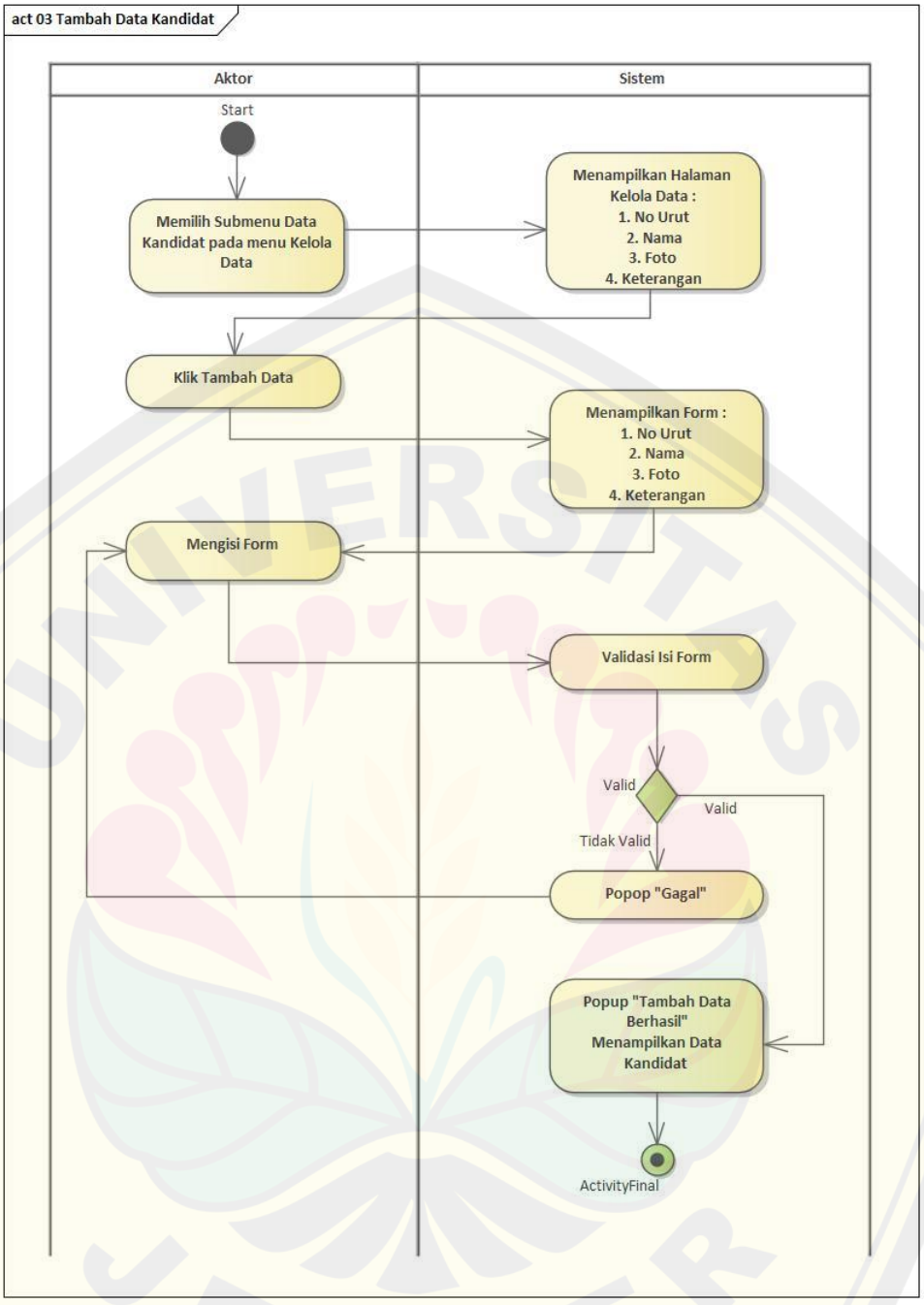
3. Melihat data pemilih



Gambar 4.4 *Activity Diagram* Melihat Data Pemilih

Aktor yang berada pada halaman utama akan memilih submenu data pemilih, kemudian sistem akan menampilkan data pemilih. Data yang ditampilkan akan disajikan dalam bentuk tabel dengan beberapa tombol kelola di samping data.

4. Tambah Data Kandidat

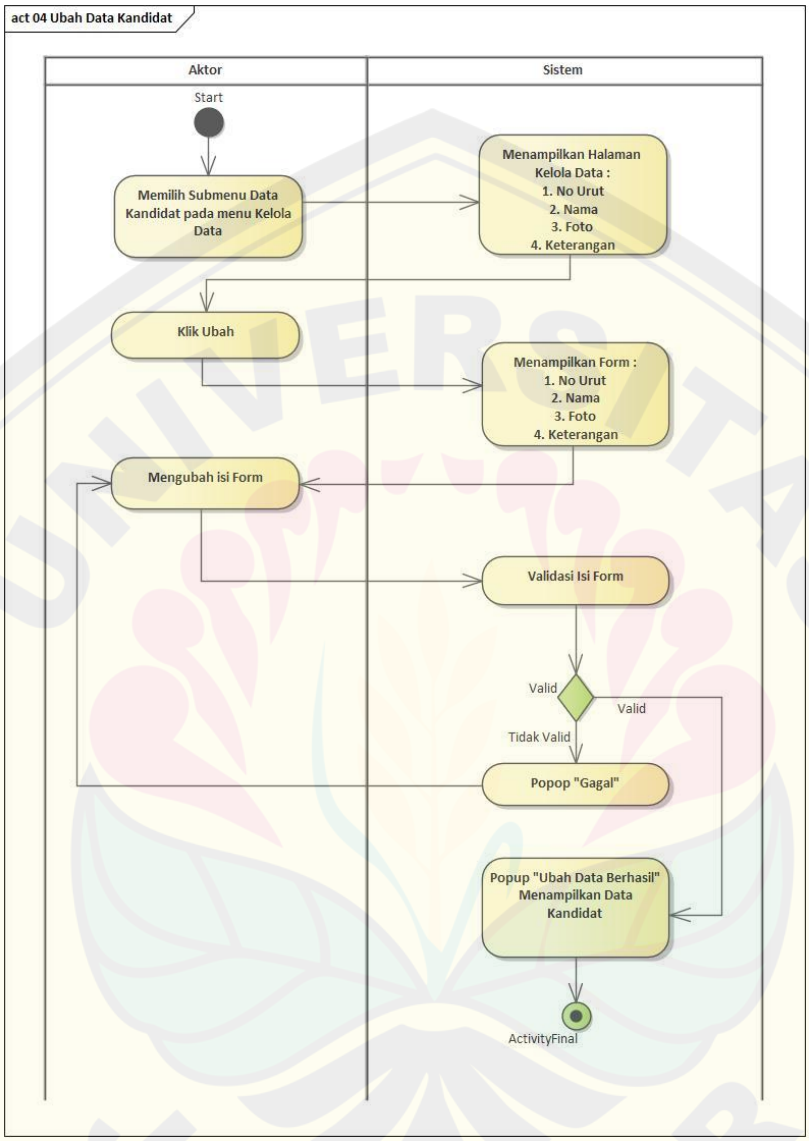


Gambar 4.5 *Activity Diagram* Tambah Data Kandidat

Aktor yang berada pada halaman utama akan memilih submenu data kandidat, kemudian sistem akan menampilkan data kandidat/calon .Aktor akan memilih tambah data yang kemudian dilanjutkan mengisi *form* yang disediakan oleh sistem. Sistem akan memvalidasi isian *form* tersebut termasuk kesesuaian tipe data ataupun data yang kosong, kemudian jika *form* valid maka data akan disimpan

dalam database, dan jika tidak valid maka akan muncul *pop-up* “gagal” dan aktor harus memperbaiki isi dari *form* tersebut.

5. Ubah Data Kandidat

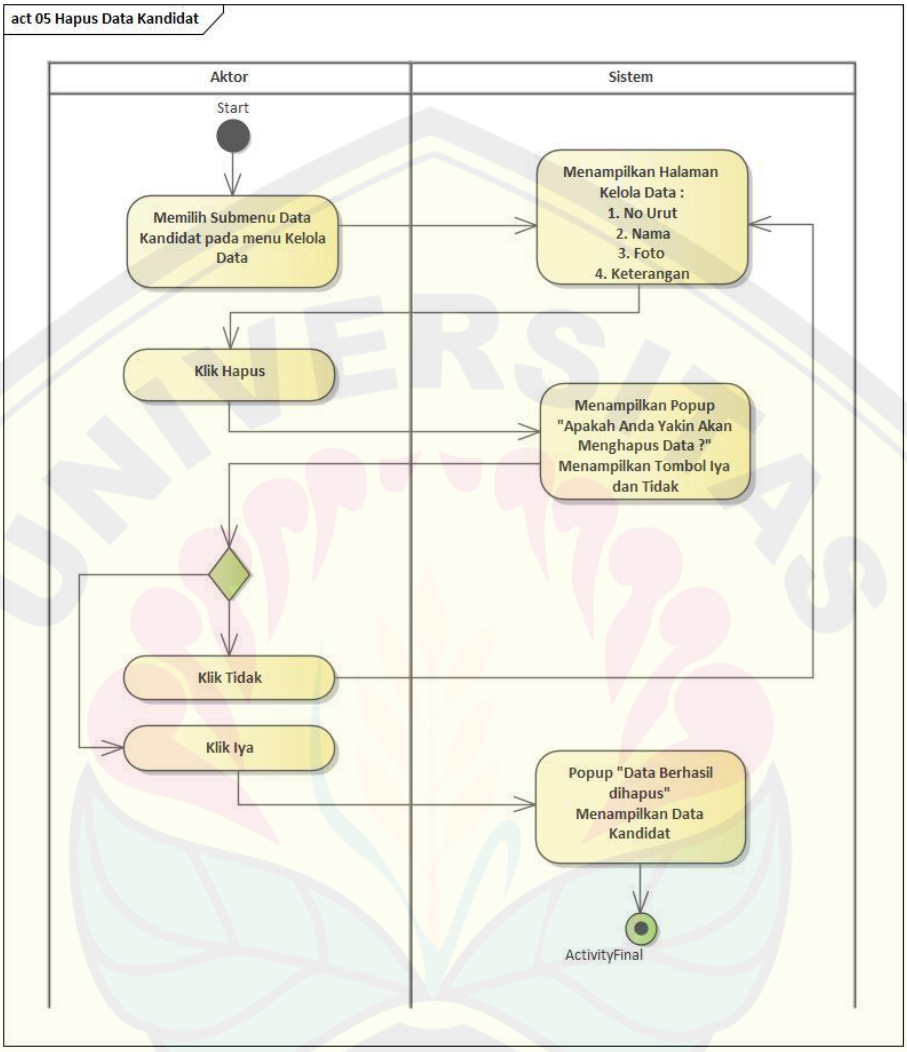


Gambar 4.6 *Activity Diagram* Ubah Data Kandidat

Aktor yang berada pada halaman utama akan memilih submenu data kandidat, kemudian sistem akan menampilkan data kandidat/calon. Aktor akan memilih data kandidat yang akan diubah, selanjutnya Aktor akan memilih tombol ubah, kemudian aktor akan mengubah data yang telah tersedia pada *form* yang ditampilkan. Sistem akan memvalidasi isian *form* tersebut termasuk kesesuaian tipe data ataupun data yang kosong, kemudian jika *form* valid maka data akan disimpan

dalam database, dan jika tidak valid maka akan muncul *popup* “gagal” dan aktor harus memperbaiki isi dari *form* tersebut.

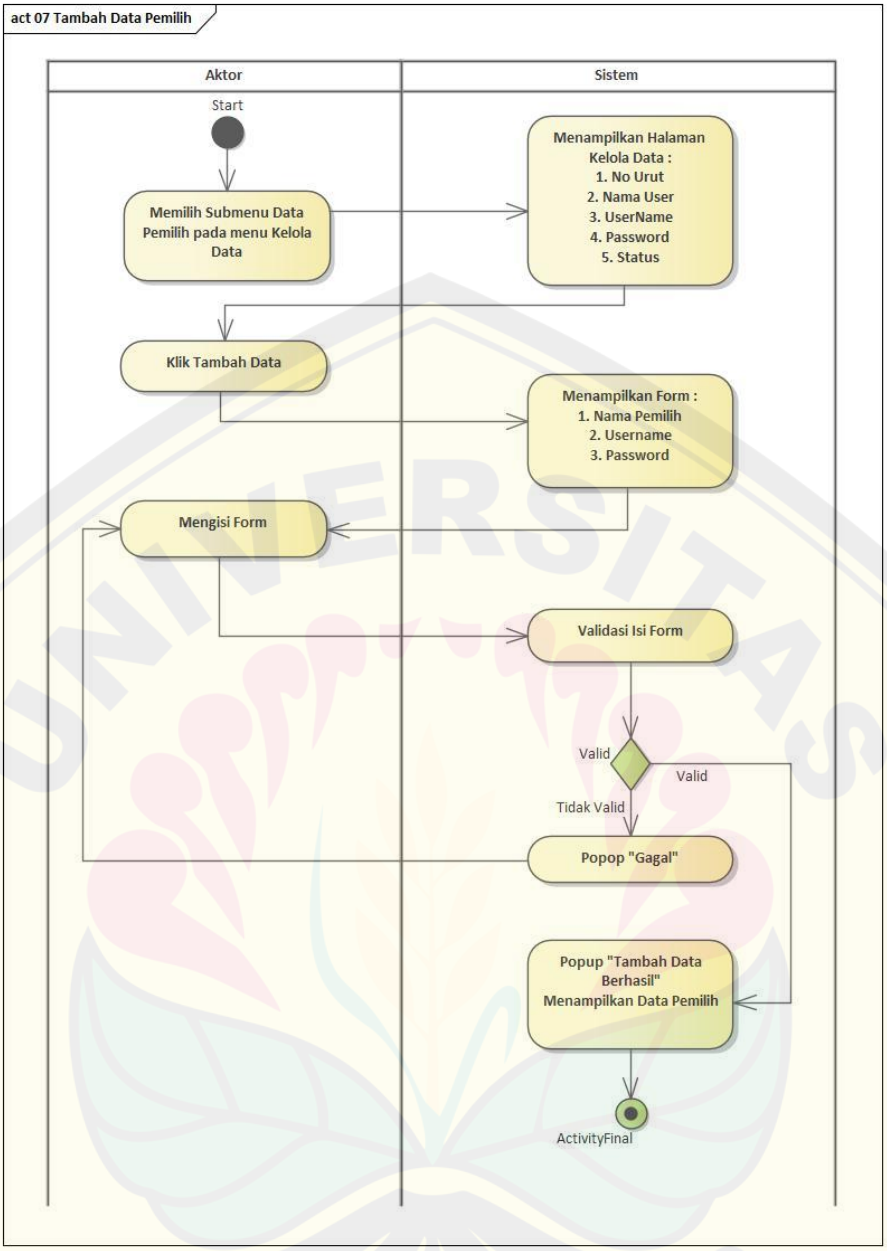
6. Hapus Data Kandidat



Gambar 4.7 *Activity Diagram* Hapus Data Kandidat

Aktor yang berada pada halaman utama akan memilih submenu data kandidat, kemudian sistem akan menampilkan data kandidat/calon. Aktor akan memilih data kandidat yang akan dihapus, kemudian aktor memilih tombol hapus. Sistem akan menghapus data yang dipilih oleh aktor pada database, data yang telah dihapus bersifat permanen dan tidak dapat dikembalikan.

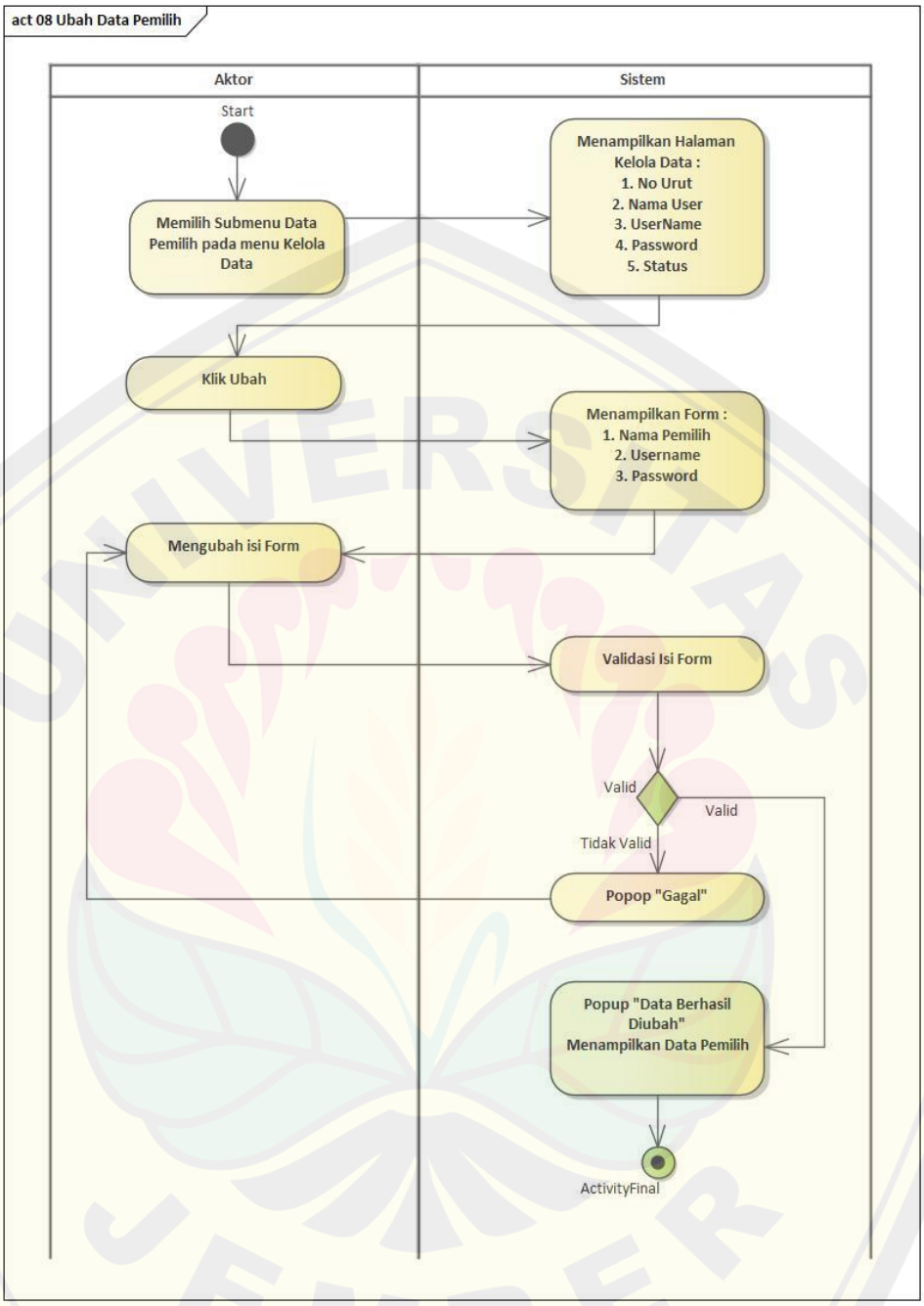
7. Tambah Data Pemilih



Gambar 4.8 *Activity Diagram* Tambah Data Pemilih

Aktor yang berada pada halaman utama akan memilih submenu data pemilih, kemudian sistem akan menampilkan data pemilih, Aktor akan memilih tambah data yang kemudian dilanjutkan mengisi *form* yang disediakan oleh sistem. Sistem akan memvalidasi isian *form* tersebut termasuk kesesuaian tipe data ataupun data yang kosong, kemudian jika *form* valid maka data akan disimpan dalam database, dan jika tidak valid maka akan muncul *pop-up* "gagal" dan aktor harus memperbaiki isi dari *form* tersebut.

8. Ubah Data Pemilih

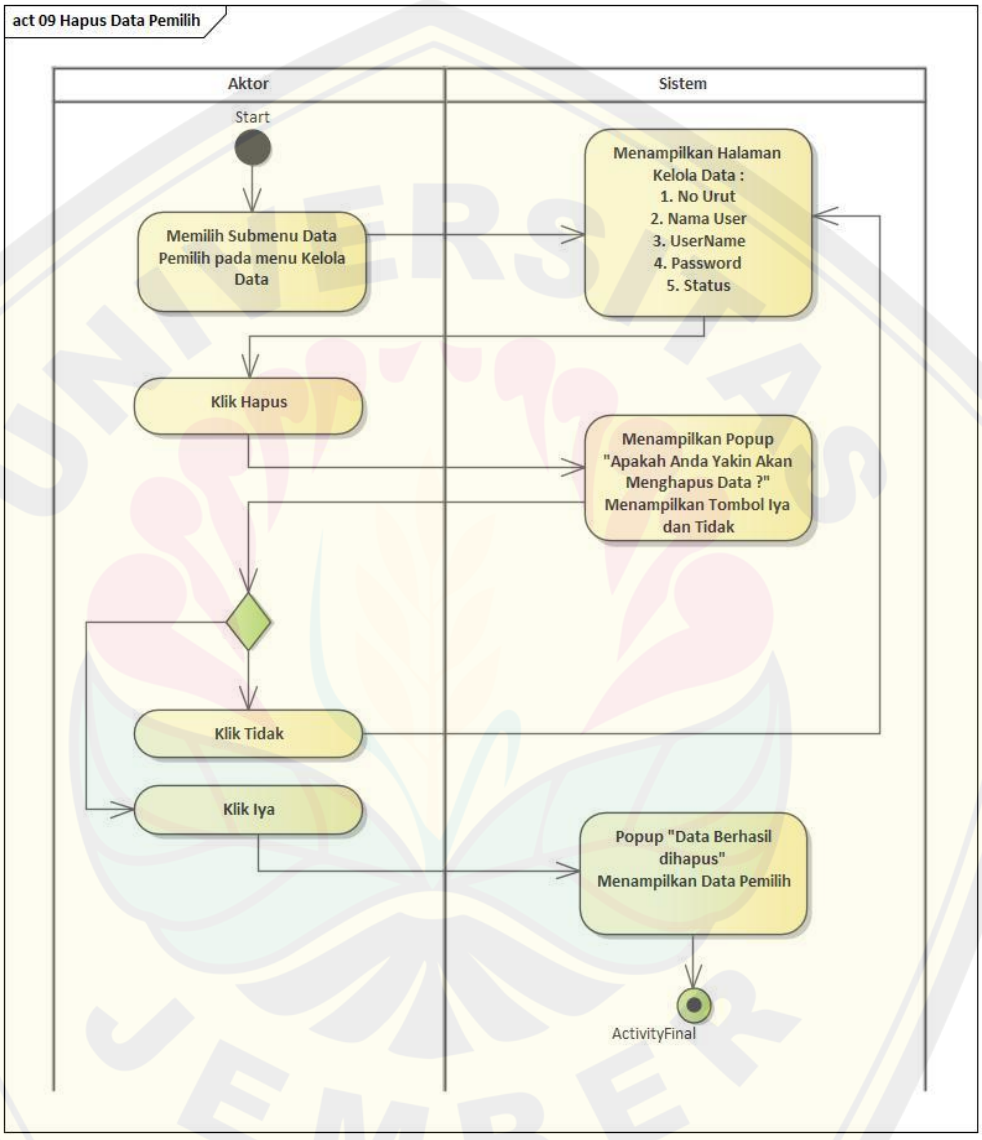


Gambar 4.9 *Activity Diagram* Ubah Data Pemilih

Aktor yang berada pada halaman utama akan memilih submenu data pemilih, kemudian sistem akan menampilkan data pemilih, aktor akan memilih satu data yang akan diubah. Selanjutnya aktor akan memilih tombol ubah, kemudian aktor akan mengubah data yang telah tersedia pada *form* yang ditampilkan. Sistem

akan memvalidasi isian *form* tersebut termasuk kesesuaian tipe data ataupun data yang kosong, kemudian jika *form* valid maka data akan disimpan dalam database, dan jika tidak valid maka akan muncul *pop-up* "gagal" dan aktor harus memperbaiki isi dari *form* tersebut.

9. Hapus Data Pemilih

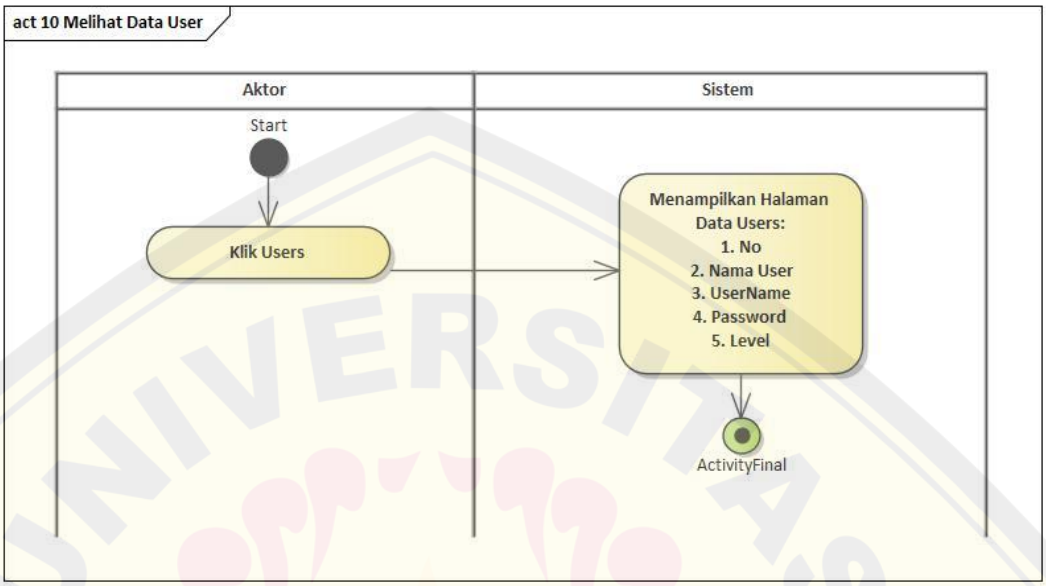


Gambar 4.10 *Activity Diagram* Hapus Data Pemilih

Aktor yang berada pada halaman utama akan memilih submenu data pemilih, kemudian sistem akan menampilkan data pemilih, aktor akan memilih satu data yang akan dihapus, kemudian aktor memilih tombol hapus. Sistem akan

menghapus data yang dipilih oleh aktor pada database, data yang telah dihapus bersifat permanen dan tidak dapat dikembalikan.

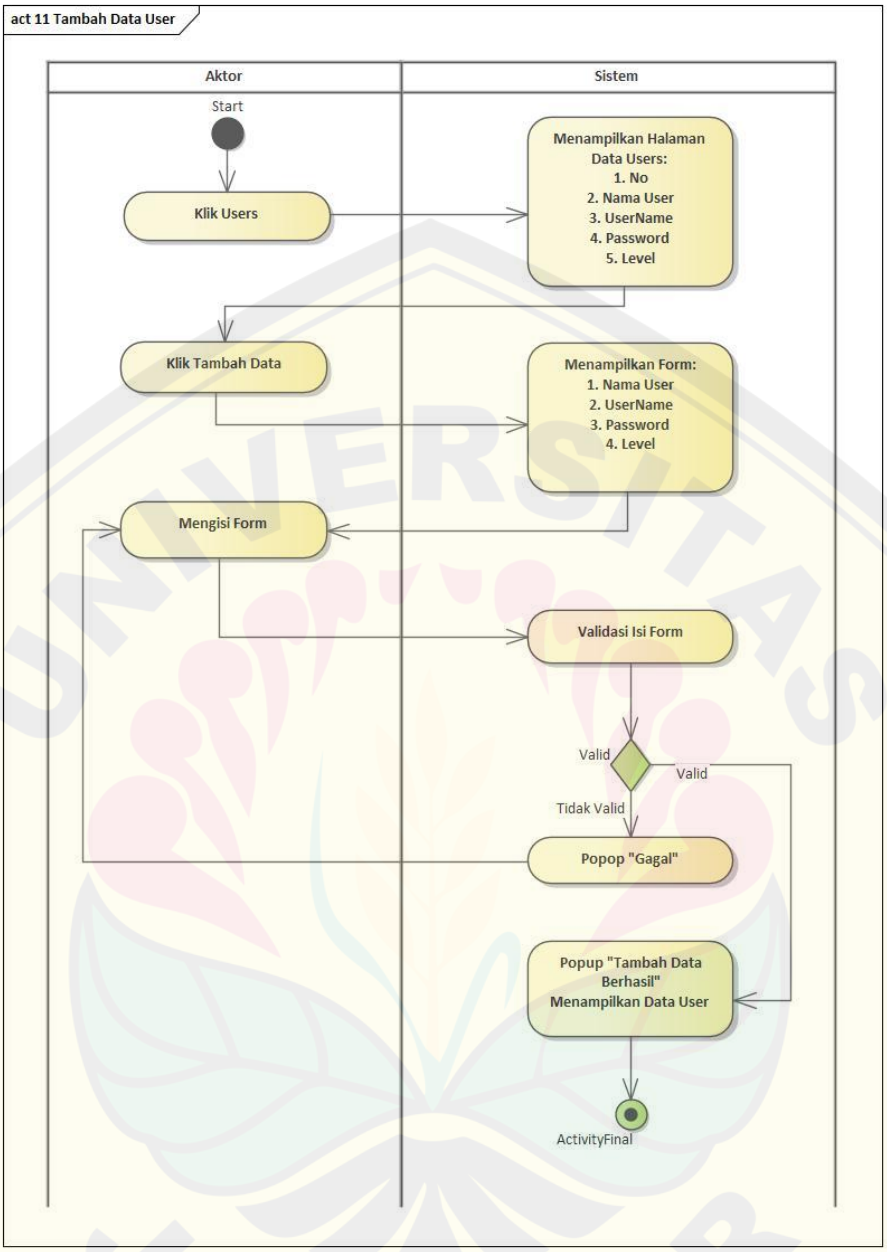
10. Melihat Data User



Gambar 4.11 *Activity Diagram* Melihat Data user

Aktor yang berada pada halaman utama akan memilih menu data *user*, kemudian sistem akan menampilkan data *User*. Data yang ditampilkan akan disajikan dalam bentuk tabel dengan beberapa tombol kelola di samping data.

11. Tambah Data User

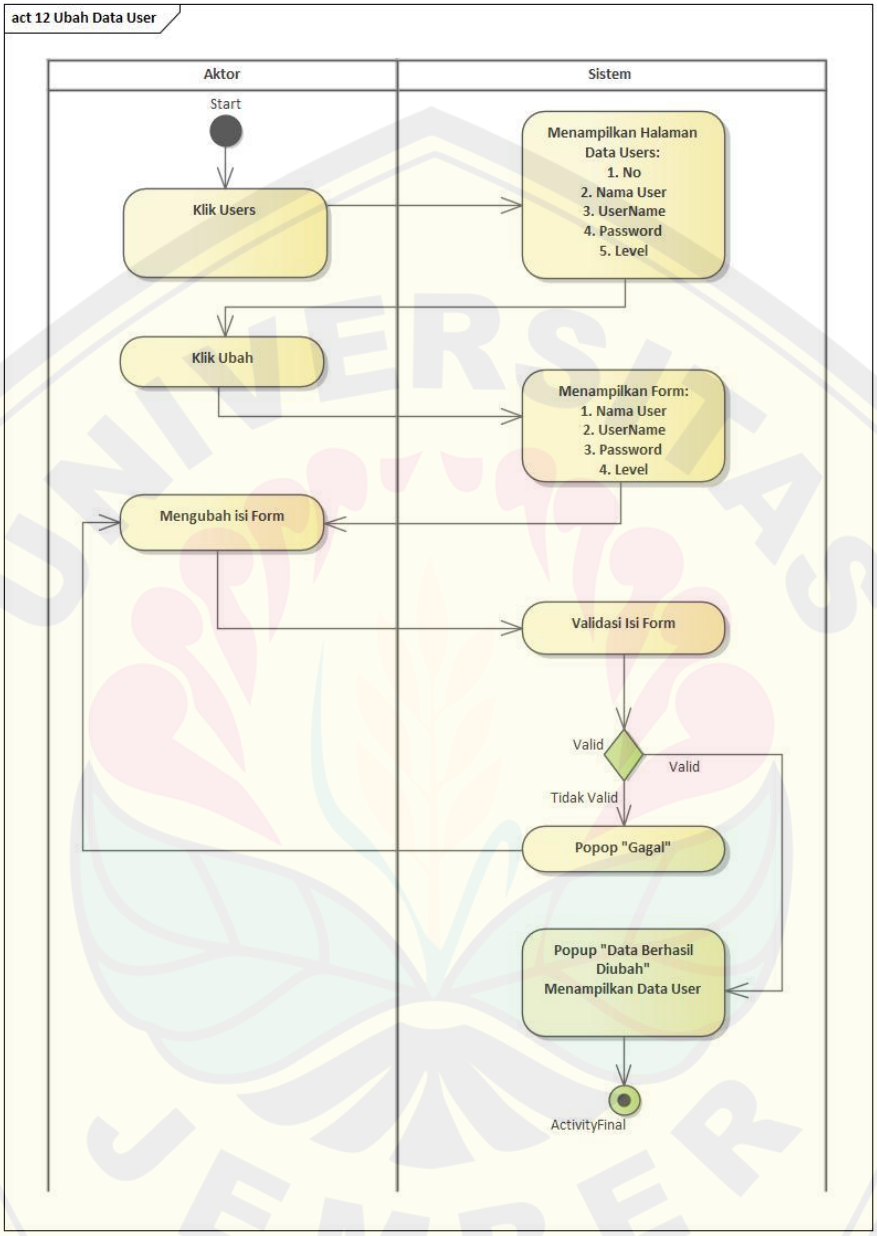


Gambar 4.12 *Activity Diagram* Tambah Data User

Aktor yang berada pada halaman utama akan memilih menu data *user*, kemudian sistem akan menampilkan data *user*. Aktor akan memilih tambah data yang kemudian dilanjutkan mengisi *form* yang disediakan oleh sistem. Sistem akan memvalidasi isian *form* tersebut termasuk kesesuaian tipe data ataupun data yang kosong, kemudian jika *form* valid maka data akan disimpan dalam database, dan

jika tidak valid maka akan muncul *pop-up* “gagal” dan aktor harus memperbaiki isi dari *form* tersebut.

12. Ubah Data *User*

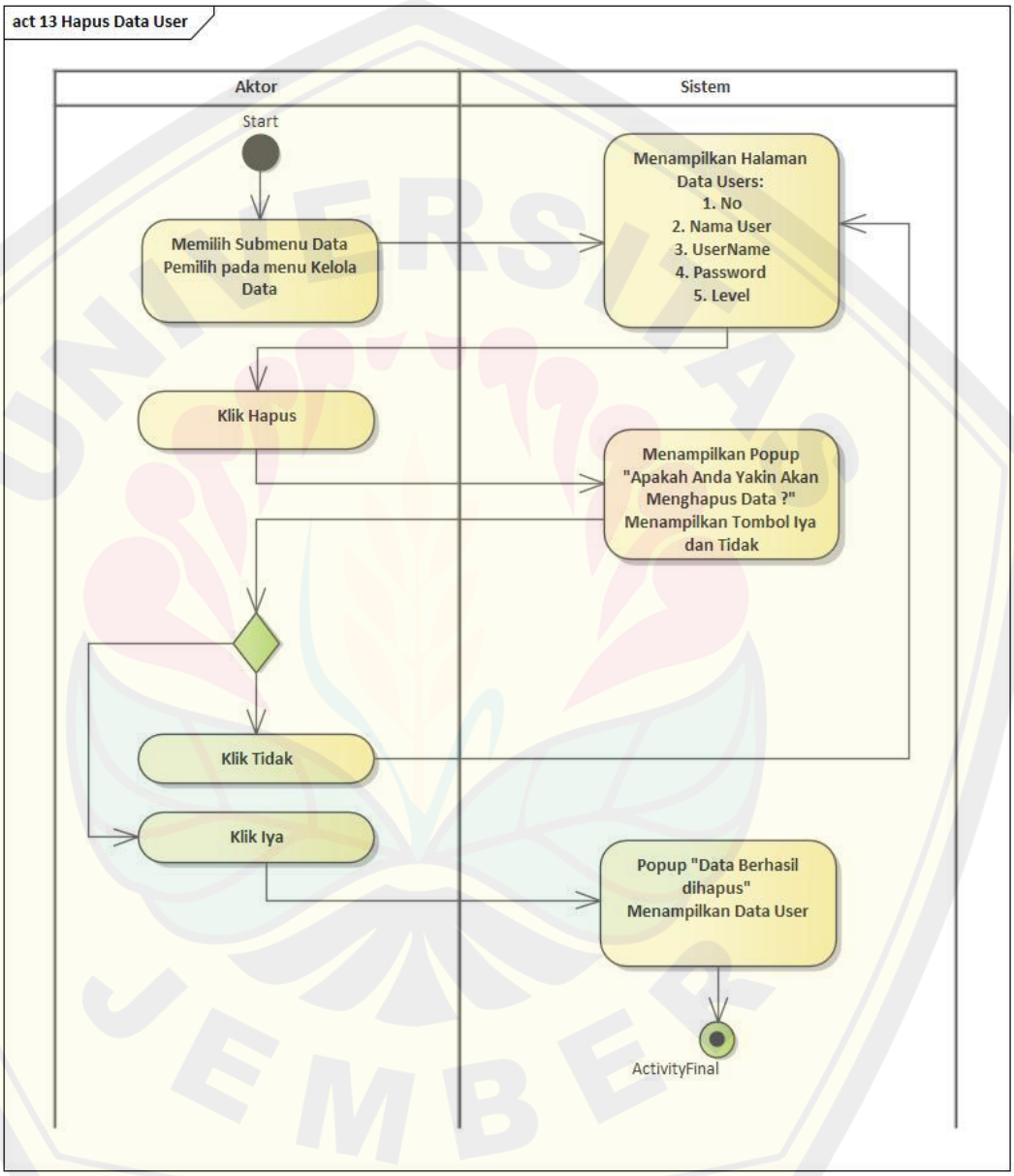


Gambar 4.13 *Activity Diagram* Ubah Data User

Aktor yang berada pada halaman utama akan memilih menu data *user*, kemudian sistem akan menampilkan data *user*, aktor akan memilih satu data *user* yang akan diubah. Selanjutnya aktor akan memilih tombol ubah, kemudian aktor akan mengubah data yang telah tersedia pada *form* yang ditampilkan. Sistem akan

memvalidasi isian *form* tersebut termasuk kesesuaian tipe data ataupun data yang kosong, kemudian jika *form* valid maka data akan disimpan dalam database, dan jika tidak valid maka akan muncul *pop-up* "gagal" dan aktor harus memperbaiki isi dari *form* tersebut.

13. Hapus Data *User*

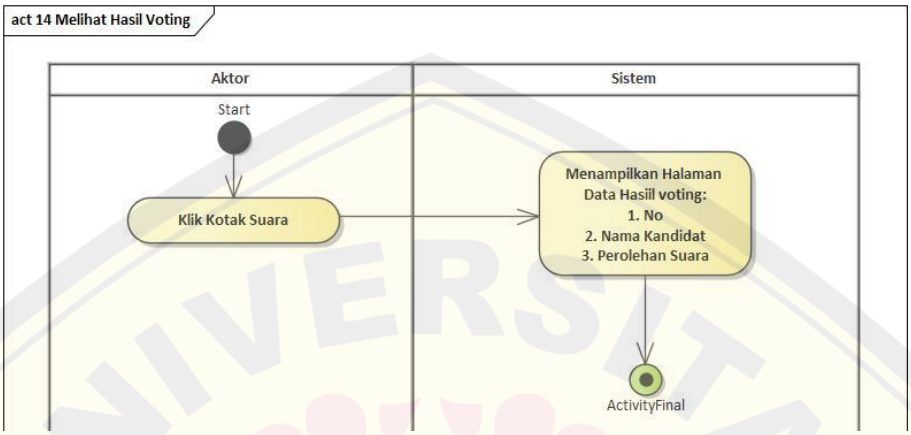


Gambar 4.14 *Activity Diagram* Hapus Data User

Aktor yang berada pada halaman utama akan memilih menu data *user*, kemudian sistem akan menampilkan data u*ser*, aktor akan memilih satu data *user*

yang akan diubah. Kemudian aktor memilih tombol hapus. Sistem akan menghapus data yang dipilih oleh aktor pada database, data yang telah dihapus bersifat permanen dan tidak dapat dikembalikan

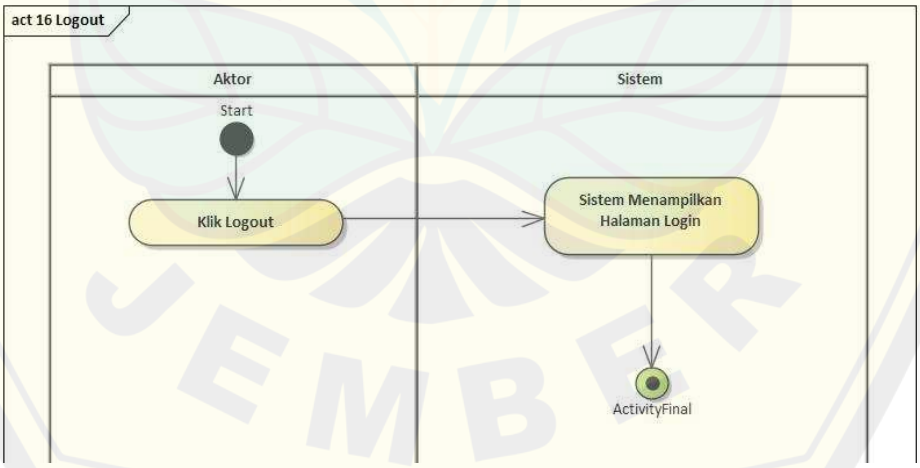
14. Melihat Hasil voting



Gambar 4.15 Activity Diagram Melihat Hasil Voting

Aktor hanya perlu mengakses alamat web, kemudian hasil voting akan ditampilkan pada halaman log in. Hasil voting akan ditampilkan pada hari dan jam yang telah ditentukan atau setelah seluruh pemilih melakukan voting.

15. Log out

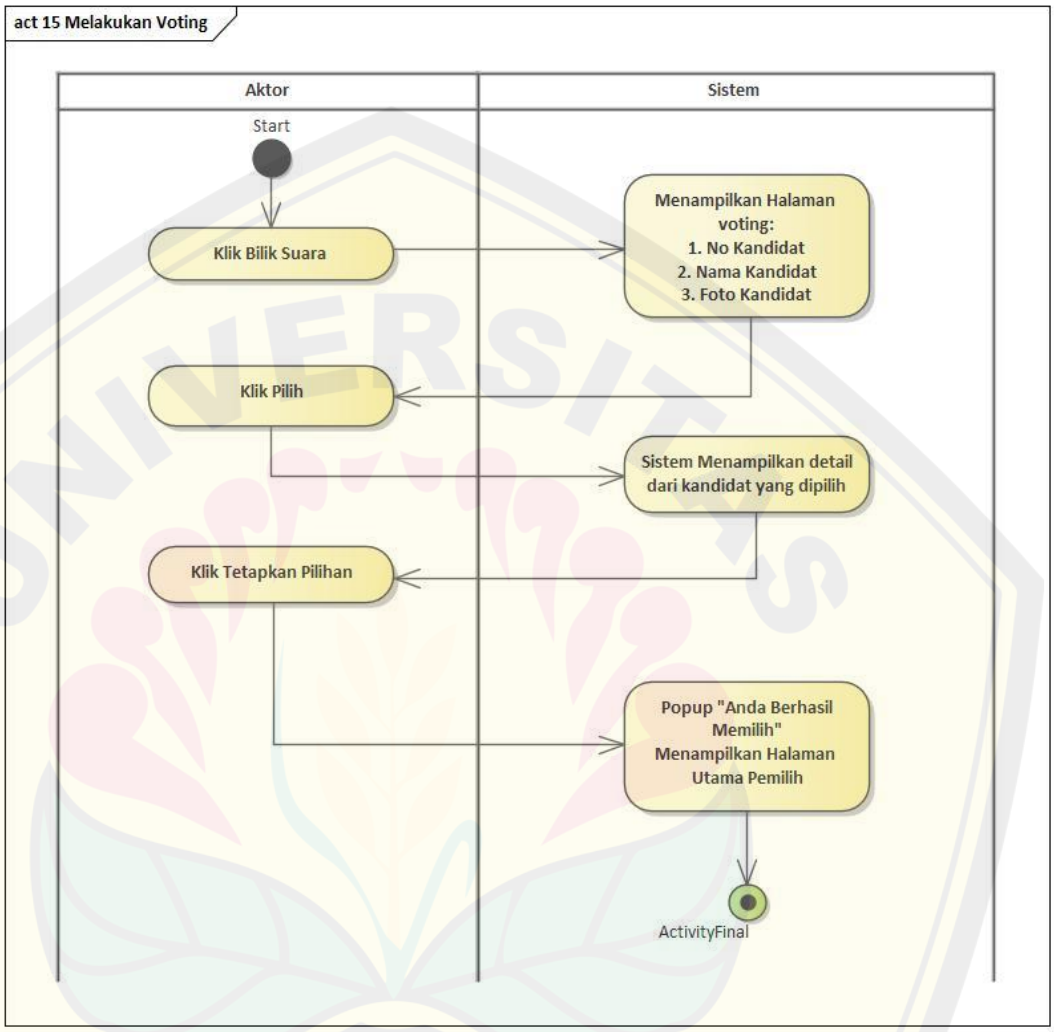


Gambar 4.16 *Activity Diagram* Logout

Aktor memilih tombol log out, kemudian sistem akan mengembalikannya pada halaman log in. Log out juga terjadi secara otomatis sesaat setelah aktor

melakukan voting, dan juga aktor yang telah melakukan voting tidak dapat log in kembali pada web.

16. Melakukan voting (tanpa RSA dan *Interlock Protocol*)



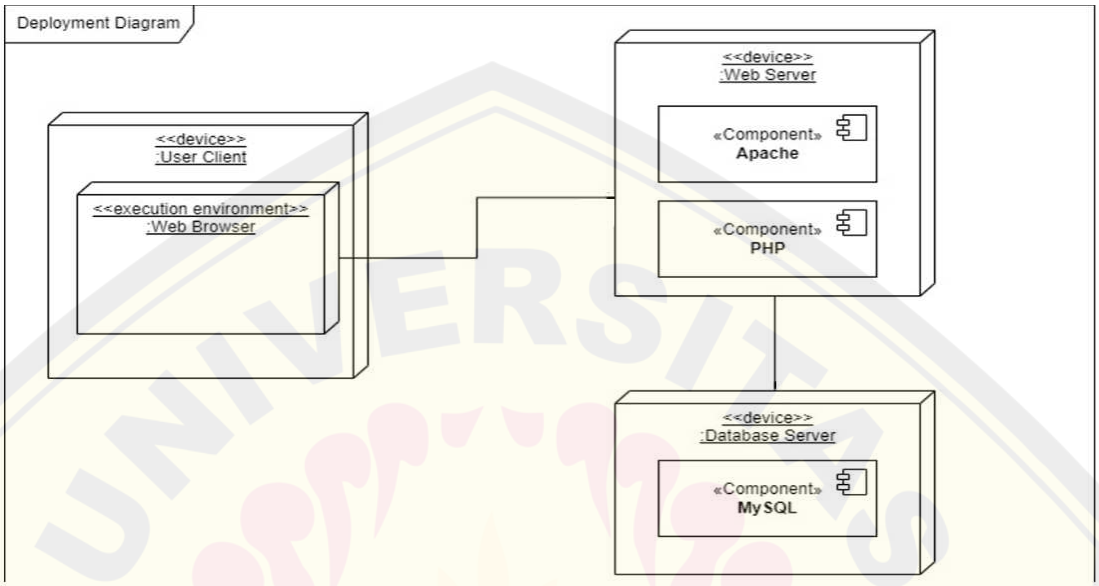
Gambar 4.17 *Activity Diagram* Melakukan Voting

Aktor akan memilih menu bilik suara yang kemudian aktor menentukan pilihan pada kandidat yang dikehendaki, kemudian sistem akan menyimpan hasil voting tersebut dalam database. Setiap pengguna hanya dapat memberikan hak suaranya 1 kali.

Diagram aktivitas melakukan voting ini yang akan disisipi oleh modul RSA dan *Interlock Protocol* pada tahapan penelitian berikutnya. Dengan acuan *flow chart* pada Gambar 3.2 dan diagram aktivitas pada Gambar 4.17 diharapkan

menghasilkan sistem yang aman dengan alur yang mudah dipahami oleh pengguna seperti pada gambar 4.16.

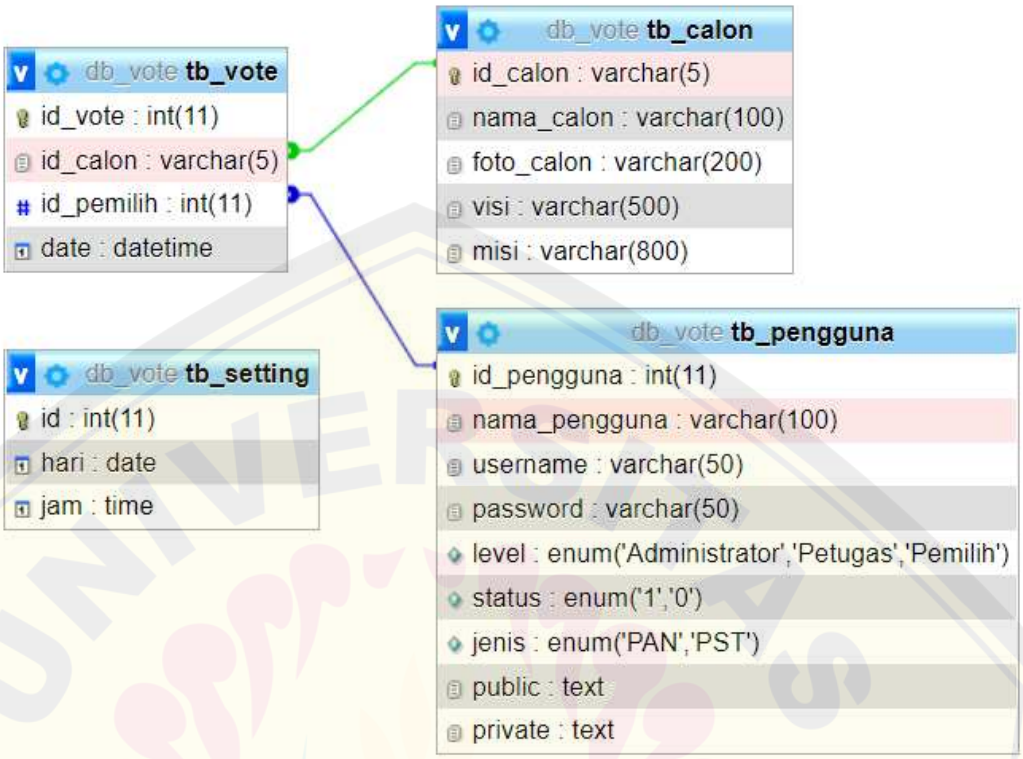
4.2.1.3 Deployment Diagram



Gambar 4.18 *Deployment Diagram*

Deployment menunjukkan *hardware* dan *software* yang digunakan dalam pembuatan sistem e-voting. *Client* akan menggunakan *device* masing-masing yang merupakan *execution environment*, artinya menggunakan *software* tertentu untuk menghubungkan dengan server yaitu web browser. Web browser nantinya akan berelasi (ditandai dengan garis hitam solid) dengan web server, selain itu web server juga berperan sebagai penghubung antara web browser dan database server.

4.2.1.4 Pemodelan Basis Data



Gambar 4.19 *Logical Record Structure*

Dalam pemodelan database menggunakan *logical record structure* untuk mengetahui struktur record pada tabel. Ada 4 tabel utama pada database yang akan digunakan, yaitu tb_calon, tb_pengguna, tb_vote dan tb_seting. Tb_calon untuk menampung data kandidat, tb_pengguna untuk menyimpan data seuruh user, tb_vote untuk menyimpan data hasil voting dan tb_setting digunakan untuk menyimpan aturan jam hari dalam menampilkan hasil voting.

4.2.2 Desain Protocol E-voting

Implementasi dari Algoritma *Rives Shamir Adleman* dan *Interlock Protocol* akan ditujukan untuk mengamankan hasil pilihan pengguna yang dikirim pada server. Hal ini bertujuan untuk meningkatkan keamanan dan kerahasiaan data yang dikirim. Pembangkitan *private key* dan *public key* akan dilakukan server sejak Admin mendaftarkan akun pemilih pada web E-voting. Hasil pembangkitan *public key* dan *private key* berupa bilangan prima acak antara 100-2000. Batas tersebut dianggap seimbang antara tingkat keamanan dan beban yang computer terima saat menjalankan. Hasil pembangkitan berupa 3 buah variabel yaitu d,e,n. Varabel d dan e digunakan sebagai *private key* sedangkan variabel 'e' dan 'n' digunakan sebagai *public key*. berikut contoh *public key* dan *private key* yang telah di simpan dalam database.

	id_pengguna	nama_pengguna	username	password	level	status	jenis	public 1	private
te	15	coba	coba	coba	Pemilih	1	PST	83441.7	83441.59143
te	11	emon	emon	emon	Pemilih	0	PST	400399.3	400399.266027
te	14	cv	cv	2015	Pemilih	1	PST	400399.3	400399.266027
te	16	wafi	wafi	1	Pemilih	1	PST	2729273.5	2729273.1635581

Gambar 4.20 Hasil Pembangkitan Kunci

Pada Gambar 4.20tersebut terdapat kolom *private key* dan *public key*. Pada tiap kolom akan berisi angka yang tidak akan sama ada setiap akun. pada kolom *private* dan *public* merupakan pasangan kunci yang digunakan RSA dalam melakukan enkripsi dan dekripsi data. Hasil pembangkitan *private key* dan *public key* berupa angka yang dipisahkan menggunakan titik, hal tersebut bertujuan untuk memudahkan proses pemanggilan . untuk penjabarannya ada pada Tabel 4.1:

Tabel 4.2 Hasil Pembangkitan Kunci

Public key(n.e)	*Private key*(n.d)	n	d	e
83441.7	83441.59143	83441	59143	7
400399.3	400399.266027	400399	266027	3
2729273.5	2729273.1635581	2729273	1635581	5

RSA akan menggunakan ketiga variabel tersebut untuk melakukan proses enkripsi dan dekripsi. Variabel n dan e yang merupakan *public key* akan digunakan

untuk proses enkripsi pada browser sesaat setelah pemilih menentukan pilihan pada voting yang pemilih lakukan, untuk proses dekripsi oleh server menggunakan variabel d dan n.

4.3 Implementasi

Dalam implementasi penulisan program (*coding*) menggunakan *tools*/aplikasi Sublime dengan Bahasa pemrograman *HTML*, *Java Script* dan *PHP* . HTML merupakan bahasa pemrograman yang digunakan untuk membuat sebuah halaman web. Proses Enkripsi pada penelitian ini ditulis menggunakan bahasa pemrograman *javascript* , dan untuk proses dekripsi ditulis menggunakan bahasa pemrograman PHP karena memang bahasa pemrograman PHP berada pada sisi server. Berikut adalah contoh tampilan dari hasil *source code* pada penelitian ini.

Gambar 4.21 Tampilah Halaman Utama Pemilih

Setelah pengguna menetapkan pilihan, sistem akan mengambil data *public key* dan *private key* pada database untuk proses Enkripsi. Proses enkripsi berada pada sisi *client* dengan menggunakan bahasa pemrograman *java script*. Berikut adalah *source code* proses enkripsi .

```
function enkrip() {
        var msg = "<?php echo $_GET['kode']; ?>";
        var public = "<?php echo $_GET['public']; ?>";
        const myArray = public.split(".");
        var n =myArray[0];// n
        var e =myArray[1];// e
```

```
        var hasilenkrip='';
        var vardata1="";
        var vardata2="";
        var getpow=(Math.pow(msg,e))%n;
        hasilenkrip = hasilenkrip.concat(getpow);
        const array = hasilenkrip.split('');
        vardata1=array[0]+""+array[1];
        for (var i = 2; i < array.length; i++) {
                vardata2 = vardata2.concat(array[i]);
        }
        var a = document.getElementById('link');
        a.href="?page=PsSQBpL&kode1="+vardata2+"&kode2="+vardat
        a1;
        }
window.onload = enkrip();
```

Hasil dari proses enkripsi tersebut berupa 2 *ciphertext* (dipecah oleh *interlock protocol*) yang kemudian akan dikirimkan ke server. Server akan menggabungkan kedua *ciphertext* tersebut kemudian didekripsi untuk disimpan pada database. Berikut adalah *source code* dekripsi yang ditulis menggunakan bahasa pemrograman PHP.

```
function dekrip_withkey($data, $user_n, $user_d) {
        $hasildekrip="";
        $teks = explode(".", $data);
        foreach ($teks as $nilai) {
                $hasildekrip=gmp_strval(gmp_mod(gmp_pow($nilai,
                $user_d), $user_n)));
        }
        return $hasildekrip;
}
```

4.4 Testing

Pengujian dilakukan untuk mengetahui apakah sistem yang dibangun sudah sesuai dengan kebutuhan pengguna. Uji yang pertama adalah uji sistem yang mengetahui kesesuaian fungsional sistem, Selanjutnya adalah uji algoritma untuk mengukur tingkat keamanan dari Algoritma RSA dan *Interlock Protocol*.

4.4.1 Black Box Testing

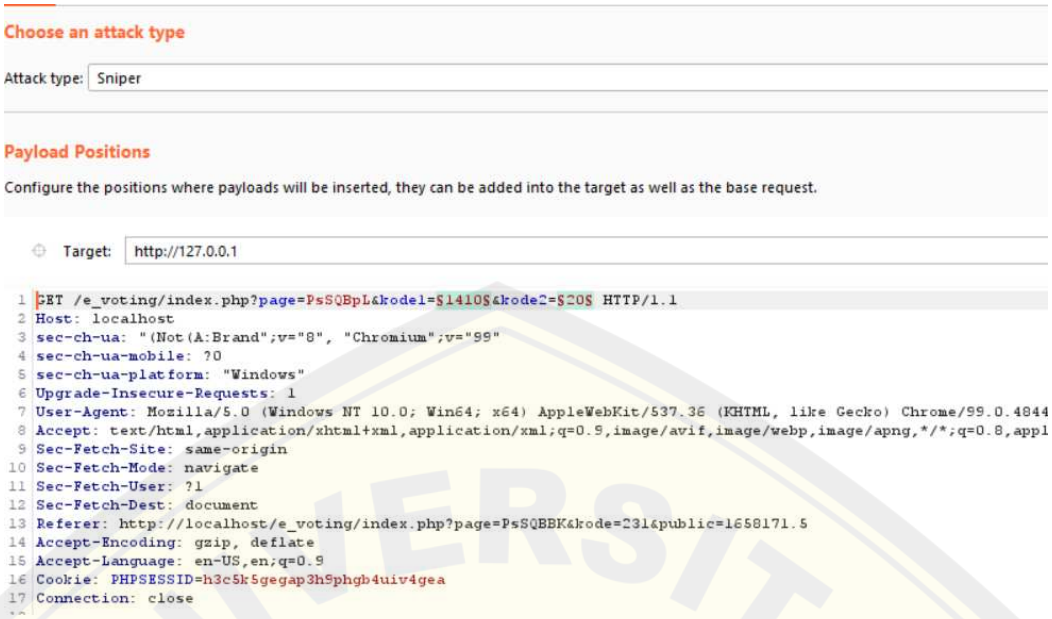
Pengujian dilakukan dengan *Black box* (uji fungsional) pada sistem yang telah dibuat. Uji fungsional artinya adalah menguji seluruh kesesuaian fungsional pada aplikasi dengan yang diharapkan. Berikut adalah Tabel hasil uji fungsional.

Tabel 4.3 Tabel Hasil Uji Fungsional

Fungsi	Hasil Pengujian
Log in	Berhasil
Melihat Data Kandidat	Berhasil
Tambah Data Kandidat	Berhasil
Ubah Data Kandidat	Berhasil
Hapus Data Kandidat	Berhasil
Melihat Data Pemilih	Berhasil
Tambah Data Pemilih	Berhasil
Ubah Data Pemilih	Berhasil
Hapus Data Pemilih	Berhasil
Melihat Data User	Berhasil
Tambah Data User	Berhasil
Ubah Data User	Berhasil
Hapus Data User	Berhasil
Melihat Hasil Voting	Berhasil
Melakukan Voting	Berhasil
Log out	Berhasil

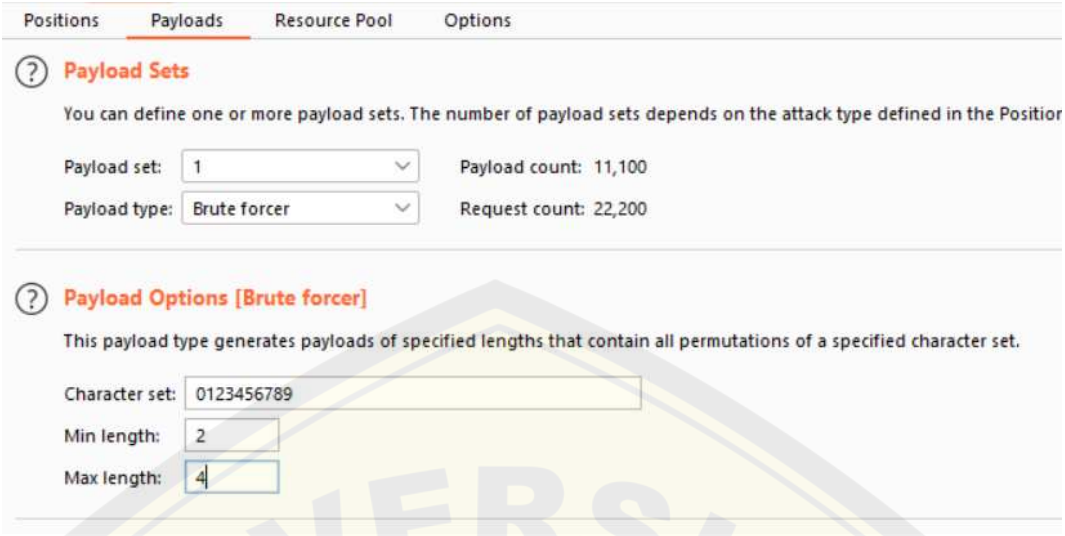
4.4.2 Pengujian Brute Force

Untuk pengujian *Brute force* digunakan *software* Burpsuite. Tahap pertama adalah menentukan tipe serangan, dalam hal ini menggunakan *Sniper* yang artinya serangan tunggal terus menerus. Selanjutnya adalah menentukan target dengan memasukkan alamat *localhost*. Kemudian kita ambil data *intercept* pemilih e-voting untuk *record* proses pada saat pemilihan berlangsung. Data tersebut merupakan acuan variabel mana yang akan di-*generate* secara berulang.



Gambar 4.22 Payload Position

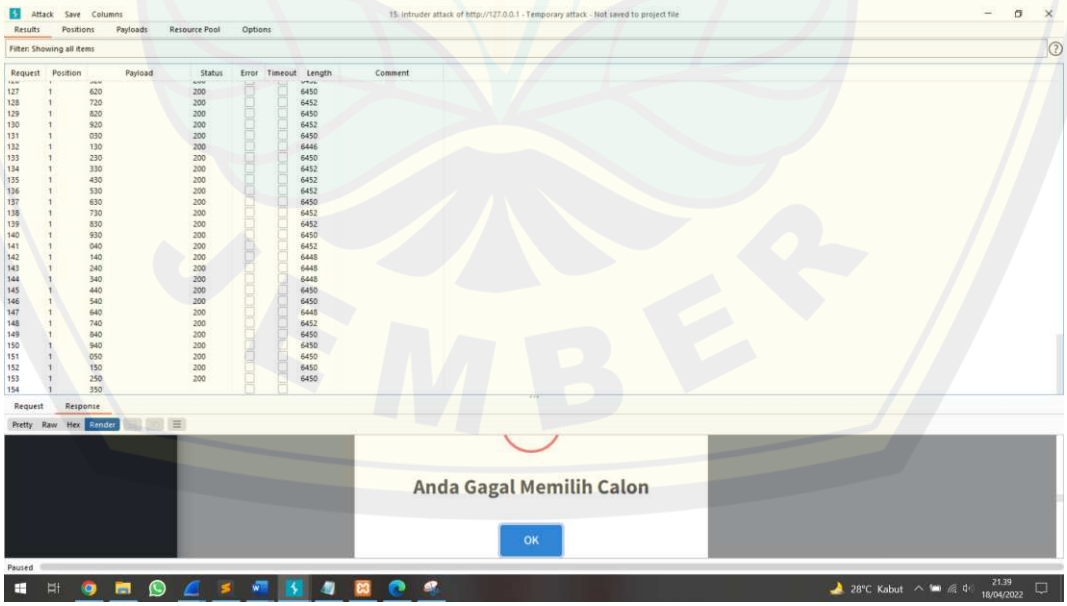
Pada Gambar 4.22 pada Baris pertama kode terlihat dua variabel yang diberi tanda $ hal tersebut merupakan penanda bahwa dua variabel tersebut akan di-*generate* secara terus menerus sampai menemukan angka yang sesuai. Kemungkinan terburuk dari tertebaknya dua variabel tersebut adalah peretas dapat mendapatkan data Id_Kandidat yang terpilih oleh pengguna, akan tetapi penambahan validasi Id_Pengguna *unique*(tidak boleh ganda) pada database maka akan memutus tindakan peretas, selain itu Id_Kandidat yang tidak berurutan juga menjadi sebuah kesulitan bagi peretas, karena hal tersebut peretas akan sulit mengganti data kandidat yang terpilih dengan id_kandidat kehendak peretas. Jika peretas ingin mendapatkan Id_Kandidat dari transaksi data yang lain juga harus melewati tahapan *brute force* yang tentunya dengan kemungkinan <1% dapat menebaknya.



Gambar 4.23 Payload Option

Pada Gambar 4.23 merupakan aturan serangan untuk jenis karakter yang akan dimasukkan serta panjang minimal dan maksimal dari digit dari variabel yang akan terus di-*generate* (tanda dolar pada Gambar 4.22).

Setelah seluruh data terisi maka kita dapat klik *start attack*. serangan akan dilakukan pada 2,5 menit pertama. Hal itu dikarenakan waktu *session* akan habis setelah 2,5 menit pada setiap akun, sehingga jika pemilih melakukan voting lebih dari 2,5 menit maka pengguna akan otomatis log out, dan pemilih diwajibkan untuk log in kembali.



Gambar 4.24 Hasil *Brute Force*

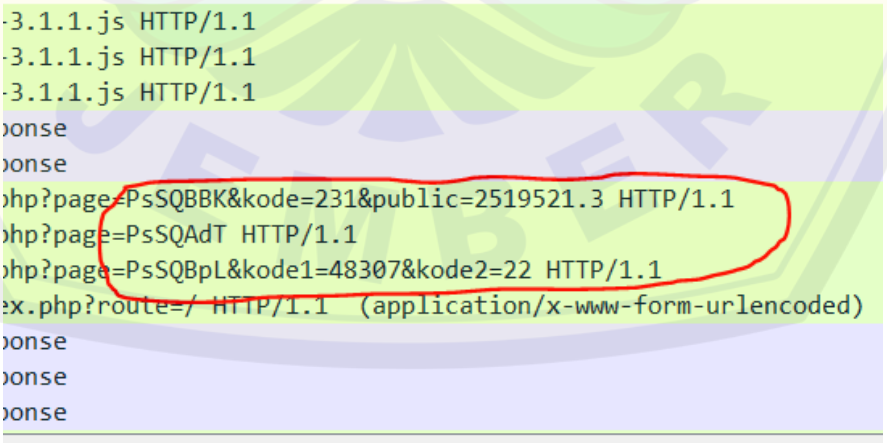
Pengujian *brute force* dilakukan selama kurang lebih 2,5 menit menghasilkan 154 kali percobaan dengan keterangan 'anda gagal memilih calon', hal itu disebabkan karena proses enkripsi pada *ciphertext* yang di hasilkan oleh *brute force* tidak sesuai dengan id calon pada database.

Pengujian hanya dilakukan selama 2.5 menit sesuai dengan waktu *session* pada setiap pemilih. Dalam 2.5 menit pertama Burpsuite berhasil mengirimkan 154 kali percobaan dari 22.200 kemungkinan, hal ini juga bukan patokan pasti karena dalam percobaan ini menggunakan *localhost* yang tentunya lebih cepat dari *hosting* internet pada umumnya. Sehingga dapat ditarik kesimpulan 0,64 % kemungkinan untuk menebak dengan sesuai dan dapat dikatakan bahwa sistem ini sangat aman dengan tingkat keamanan 99,36%.

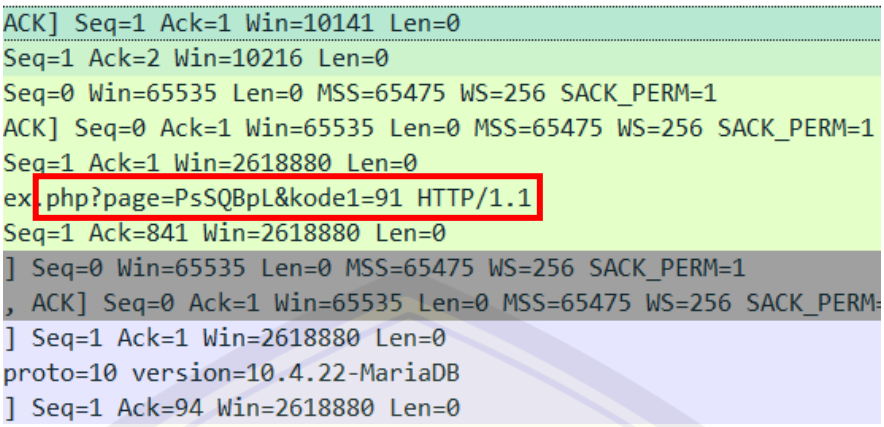
Disisi lain penggunaan *interlock protocol* dalam memecah *ciphertext* menjadi 2 bagian juga menambah kombinasi dari setiap karakter *ciphertext* tersebut. Dalam hal ini diperoleh 6 digit *ciphertext* yang seharusnya ada 11100 kemungkinan, tetapi setelah penggunaan *interlock protocol* maka kemungkinan percobaan menjadi 22200, sehingga dapat disimpulkan menambah keamanan sampai 2 kali lipat.

4.4.3 Pengujian MITM

Pengujian dilakukan menggunakan *software* Wireshark. Dengan melihat *traffic* yang ada pada komputer maka akan terlihat proses transaksi data Ketika pemilih melakukan voting.



Gambar 4.25 *Capture Traffic* data setelah Implementasi RSA dan *Interlock Protocol*



Gambar 4.26 *Capture Traffic* data sebelum Implementasi RSA dan *Interlock Protocol*

Pada Gambar 4.25 dapat diketahui transaksi data yang didapatkan antara *client* dan server. Dari gambar tersebut dapat diperoleh informasi tentang *public key*, kode1 dan kode2, kedua kode tersebut merupakan *ciphertext* yang telah dipisah berdasarkan interlock protocol sehingga selain tidak bermakna, peretas juga harus menyusun angka tersebut sampai menjadi *ciphertext* yang benar.

Pada Gambar 4.26 menunjukkan transaksi data pada sistem sebelum adanya implementasi algoritma RSA dan *interlock protocol* dari gambar tersebut kita akan mendapatkan data Id Kandidat (kode1=91), dengan ini kita dapat menyimpulkan bahwa peretas akan langsung mendapatkan apa yang pengguna pilih, dengan mempertimbangkan kemungkinan terburuk yaitu mengganti data yang akan dikirim maka akan diperoleh hasil yang tidak sesuai dengan tidak ada eror pada aplikasi. Jika kita bandingkan dengan Gambar 4.25 transaksi yang kita dapatkan berupa dua buah *ciphertext* (kode1=48307 dan kode2=22) sehingga dapat ditarik kesimpulan bahwa keamanan pada sistem akan sangat meningkat, mengingat hasil tes *brute force* pada subbab 4.4.2 sebesar 99,36 % sistem dapat dikatakan aman.

4.5 Pembahasan

Dari penelitian yang dilakukan didapat hasil sebuah modul pengamanan transaksi data menggunakan Algoritma RSA dan *Interlock protocol*. Modul ini tentunya dapat diaplikasikan pada seluruh *website* yang memiliki transaksi data yang dirahasiakan. Penerapan kedua metode ini dalam sistem e-voting memiliki dampak yang sangat baik terbukti dengan uji *brute force* dan *man in the middle* data transaksi tidak dapat terbaca.

Dalam uji *brute force* pada penelitian ini *software* Burpsuite berhasil mengirimkan 153 kemungkinan dari 22200 kemungkinan yang ada dalam waktu 2.5 menit. Yang artinya peretas memiliki 0.64% kemungkinan untuk menebak data transaksi tersebut. Banyaknya kemungkinan tersebut juga dipengaruhi oleh seberapa cepat server memproses, koneksi internet serta kecepatan komputer yang digunakan. Dalam penelitian ini karena berbasis *localhost* maka sudah dapat menggambarkan koneksi internet yang sangat cepat.

Untuk pengujian *man in the middle* saat membaca *traffic record* data hasil voting berhasil terenkripsi dengan baik. Pemanfaatan algoritma RSA serta pemecahan data menjadi 2 bagian oleh *interlock protocol* menjadi sangat aman, hal itu dikarenakan penyadap juga harus merangkai *ciphertext* sebelum dia berusaha untuk mendekripsi *ciphertext* tersebut. Hasil enkripsi dari transaksi data akan berbeda beda walaupun dengan basis *plaintext* sama hal tersebut dikarenakan *public key* dan *private key* akan di-*generate* 2.5 menit setelah pemilih log in tanpa voting.

BAB 5. PENUTUP

5.1 Kesimpulan

Hasil dari penelitian yang dilakukan, dapat diambil kesimpulan sebagai berikut :

1. Algoritma RSA diimplementasikan untuk mengubah data yang dikirim pemilih menjadi *ciphertext* agar menjadi tidak bermakna ketika terbaca oleh peretas. Kemudian *Interlock Protocol* akan memecah *ciphertext* tersebut sehingga menjadi 2 bagian dengan tujuan peretas masih harus menyusun *ciphertext* tersebut sebelum mendekripsinya. Kunci privat yang hanya dimiliki oleh server akan menjaga kerahasiaan data yang dikirim sehingga data tersebut hanya dapat dibaca oleh server.
2. Keamanan pada sistem E-voting diuji menggunakan simulasi serangan *brute force* dan *man in the middle*. Uji keamanan menggunakan Teknik *brute force*, *software* Burpsuite yang digunakan sebagai alat bantu uji mampu mengirimkan 153 percobaan dari 22200 kemungkinan yang ada dalam waktu 2.5 menit, dengan hasil pengujian 99,36% sistem dapat dikatakan aman. Pemecahan *ciphertext* menjadi 2 bagian saat uji *man in the middle* mengharuskan peretas untuk menyusun *ciphertext* tersebut sebelum mendekripsinya, sehingga ada 2 proses yang harus dilakukan oleh peretas sebelum mendekripsi *ciphertext* tersebut, yaitu menyusun dan selanjutnya adalah mendekripsi *ciphertext* tersebut. Dari hasil kedua percobaan tersebut sistem yang telah diimplementasi menggunakan algoritma RSA dan *Interlock protocol* dapat dikatakan aman.

5.2 Saran

Saran penelitian untuk mengembangkan sistem e-voting yang telah dibuat:

1. Modul RSA yang dibuat masih dapat dimodifikasi untuk menghasilkan *ciphertext* yang lebih panjang dan rumit hal ini tentu diselaraskan dengan data yang akan dienkripsi..
2. Untuk segi keamanan secara keseluruhan diharapkan untuk mengembangkan tingkat keamanan pada fitur log in dengan tujuan keamanan yang lebih optimal,

seperti penambahan kode OTP atau sidik jari untuk menjamin asas langsung dalam sistem voting.

3. Pembangkitan ulang kunci publik dan kunci privat dalam interval waktu beberapa menit sangat mengurangi kecepatan pada sistem, sehingga perlu dilakukan penelitian lanjutan.

DAFTAR PUSTAKA

Ahmad, A. (2012). Perkembangan Teknologi Komunikasi dan Kesenjangan Informasi: Akar Informasi dan Berbagai Standarnya. *Jurnal Dakwah Tabligh*, *13*(1), 137–149.

Basri. (2016). Kriptografi Simetris Dan Asimetris Dalam Perspektif Keamanan Data Dan Kompleksitas Komputasi. *Jurnal Ilmiah Ilmu Komputer*, *2*(2), 17–23. http://ejournal.fikom-unasman.ac.id

Carolina, I., & Supriyatna, A. (2019). Penerapan Metode Extreme Programming dalam Perancangan Aplikasi Perhitungan Kuota SKS Mengajar Dosen. *Jurnal IKRA-ITH Informatika*, *3*(1), 106–113.

Ginting, A., Isnanto, R. R., & Windasari, I. P. (2015). 144706-ID-implementasi-algoritma-kriptografi-rsa-u. *Jurnal Teknologi dan Sistem Komputer*, *3*(2), 253–258. https://media.neliti.com/media/publications/144706-ID-implementasi-algoritma-kriptografi-rsa-u.pdf

Hutchison, D., & Mitchell, J. C. (2011). E-Voting and Identity: Third International Conference, VoteID 2011. In *E-Voting and Identity: Third International Conference, VoteID 2011, Tallinn, Estonia, September 2011, Revised Selected Papers: Vol. LNCS 7187*. http://www.scopus.com/inward/record.url?eid=2-s2.0-84867441505&partnerID=40&md5=7e0e6117592c287f7f992f4aa9a8469c%5Cnhttp://link.springer.com/10.1007/978-3-642-32747-6

Karandikar, R. L. (2007). Introduction to cryptography. *E-Business Process Management: Technologies and Solutions*, 28–44. https://doi.org/10.4018/978-1-59904-204-6.ch002

Kurniawan, S. T. C., Dedih, D., & Supriyadi, S. (2018). Implementasi Kriptografi Algoritma Rivest Shamir Adleman dengan Playfair Cipher pada Pesan Teks Berbasis Android. *Jurnal Online Informatika*, *2*(2), 102. https://doi.org/10.15575/join.v2i2.113

Nabilah, A., & Amrozi, Y. (2019). Rancang Bangun E-Voting Berbasis Web Pada

Organisasi Karang Taruna Kelurahan Kedurus. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, *2*(3), 105. https://doi.org/10.32493/jtsi.v2i3.2751

Ridwan, M., Arifin, Z., Studi Ilmu Komputer, P., Ilmu Komputer dan Teknologi Informasi, F., Mulawarman Jalan Barong Tongkok Kampus Gunung Kelua Samarinda, U., & Timur, K. (2016). RANCANG BANGUN E-VOTING DENGAN MENGGUNAKAN KEAMANAN ALGORITMA RIVEST SHAMIR ADLEMAN (RSA) BERBASIS WEB (STUDI KASUS : PEMILIHAN KETUA BEM FMIPA). *Jurnal Informatika Mulawarman*, *11*(2). http://doc.google.com

Risnanto, S. (2018). Aplikasi Pemungutan Suara Elektronik / E-Voting Menggunakan Teknologi Short Message Service Dan At Command. *Jurnal Teknik Informatika*, *10*(1), 17–26. https://doi.org/10.15408/jti.v10i1.6811

Rizal, M. Y. (2012). PERANCANGAN SIMULASI MAN IN THE MIDDLE ATTACK PADA ALGORITMA KRIPTOGRAFI RSA DAN PENCEGAHANNYA DENGAN INTERLOCK PROTOCOL. *Junrnal AMIKOM*, 1–20.

Satrya, F., Kusumah, F., Guritman, S., Giri, E. P., Ibn, U., & Bogor, K. (2015). Desain E - Voting Pilkada Kota Bogor Menggunakan Protokol Two Central. *Jurnal Krea-TIF*, *03*, 24–37.

Wijaya, J. H., Zulfikar, A., & Permatasari, I. A. (2019). Implementasi Sistem E-Voting Untuk Meningkatkan Kualitas Demokrasi di Indonesia. *Jurnal Pemerintahan dan Kebijakan (JPK)*, *1*(1), 51–59. https://doi.org/10.18196/jpk.v1i1.7841

Yusmiarti, K. (2020). E-Voting Pemilihan Kepala Desa Berbasis Android. *Jurnal Informatika*, *8*(2), 1–7.