# Pengembangan Aplikasi E-Voting Menggunakan Enkripsi Homomorfik

Muhtar Hartopo
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
muhtarhartopo@gmail.com

Dr. Ir. Rinaldi Munir, S.T., M.T.
Teknik Informatika
Institut Teknologi Bandung
Bandung, Indonesia
rinaldi@informatika.org

***Abstrak*—E-voting memiliki banyak keuntungan diantaranya menambah efisiensi waktu dan biaya pemungutan suara serta mengurangi kemungkinan kesalahan perhitungan. Namun selain kelebihan tersebut terdapat risiko keamanan. Risiko tersebut berupa risiko pencurian dan manipulasi data. Aplikasi e-voting yang dibangun menggunakan enkripsi homomorfik dengan algoritma Pailier untuk menangani permasalahan tesebut. Enkripsi homomorfik memungkinkan komputasi pada ciphertext. Aplikasi dibangun dengan menggunakan bahasa pemrograman Java dengan dua bagian yaitu bagian client dan bagian server. Hasil pengujian EVSSO menunjukkan bahwa aplikasi e-voting yang dibangun memiliki kualitas keamanan berupa aspek kriptografi dan kesesuaian dengan asas-asas pemilihan dengan nilai masing-masing 5 dan 6.**

***Keywords—enkripsi; homomorfik; pailier; e-voting; aplikasi***

## I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah banyak mengubah cara hidup manusia. Perubahan tersebut dapat dirasakan mulai dari cara berkomunikasi, cara berbelanja, cara belajar mengajar dan masih banyak lagi. Perubahan-perubahan tersebut didasari karena keinginan manusia untuk melakukan pekerjaan dengan lebih efisien. Dalam aspek kehidupan berdemokrasi pun telah dipengaruhi oleh perkembangan teknologi, salah satunya adalah penerapan *electronic voting* pada pemilihan. Penggunaan *e-voting* tersebut dinilai menguntungkan karena dapat menghemat anggaran, menghemat waktu pemilihan, mempermudah perhitungan suara dan lain-lainnya.

Disamping banyaknya manfaat penggunaan sistem *e-voting* pada proses pemungutan suara, terdapat pula risiko yang mengancam. Beberapa diantara risiko tersebut adalah masalah keamanan dan masalah kerahasiaan pemilihan. Masalah keamanan yang dapat terjadi contohnya pencurian data pemilihan, penyadapan dan manipulasi suara. Kemudian pada masalah privasi juga menjadi masalah serius. Data yang tersimpan pada basis data bisa saja data *plaintext* , pemilih dan pilihannya akan terpampang jelas di basis data. Hal ini tidak sesuai dengan prinsip pemilihan yang bersifat rahasia. Kemudian muncul solusi agar data pemilih tersebut dienkripsi sebelum disimpan di basis data. Solusi tersebut cukup bagus, namun akan sulit apabila akan dilakukan proses pengolahan pada data tersebut karena data tersebut harus didekripsi menjadi *plaintext* terlebih dahulu kemudian dapat diolah. Proses dekripsi data di komputer sebelum diolah akan memiliki risiko keamanan.

Salah satu solusi untuk mengatasi permasalahan privasi dan keamanan pada *e-voting* adalah dengan menyimpan data yang terenkripsi lalu melakukan pengolahan pada data yang terenkripsi tersebut tanpa perlu melakukan proses dekripsi terlebih dahulu. Dengan menyimpan dan mengolah data dalam bentuk terenkripsi kita tidak perlu takut apabila data tersebut dicuri. Konsep untuk melakukan proses komputasi tertentu pada pesan terenkripsi disebut enkripsi homomorfik [10].

Enkripsi homomorfik merupakan suatu bentuk enkripsi yang memungkinkan dilakukannya komputasi pada *ciphertext* tanpa mendekripsi terlebih dahulu *ciphertext* tersebut. Operasi yang dilakukan pada *ciphertext* yang menggunakan enkripsi homomorfik akan menghasilkan *ciphertext* yang jika didekripsi akan menghasilkan hasil yang sama dengan operasi serupa pada *plaintext*. Cara ini dapat dilakukan untuk menambah keamanan pada *voting system* dan *cloud computing* [6].

Skema yang dapat digunakan untuk enkripsi homomorfik ada dua macam yaitu *partially homomorphic encryption* (PHE) dan *fully homomorphic encryption* (FHE). PHE merupakan jenis enkripsi homomorfik yang memungkinkan dilakukannya satu jenis operasi tertentu pada *ciphertext*. Sementara itu FHE merupakan jenis enkripsi homomorfik yang memungkinkan kedua jenis operasi penjumlahan dan perkalian dilakukan pada *ciphertext* [10]. Layanan yang ada saat ini belum menyediakan penggunaan enkripsi homomorfik. Bahkan masih sangat sulit untuk menemukan aplikasi di dunia nyata yang menggunakan enkripsi homomorfik. Makalah ini akan membahas mengenai Aplikasi *E-voting* Menggunakan Enkripsi Homomorfik. Sesuai dengan judulnya, makalah ini akan membahas mengenai pengaplikasian konsep enkripsi homomorfik yang diterapkan pada sistem *e-voting*.

## II. LANDASAN TEORI

Pada bab ini berisi landasan teori yang dipakai dalam makalah ini

## A. *Enkripsi Homomorfik*

Enkripsi Homomorfik (*homomorphic encryption*) adalah suatu bentuk enkripsi yang memungkinkan dilakukannya komputasi pada *ciphertext* tanpa mendekripsi terlebih dahulu *ciphertext* tersebut. Operasi yang dilakukan pada *ciphertext* yang menggunakan enkripsi homomorfik akan menghasilkan *ciphertext* yang jika didekripsi akan menghasilkan hasil yang sama dengan operasi serupa pada *plaintext* [6].

Secara matematis, *homomorphic cryptosystem* adalah sebuah *cryptosystem* yang menggunakan fungsi enkripsi yang bersifat homomorfik dan memungkinkan dilakukannya operasi pada *ciphertext*. Terdapat dua jenis operasi utama yaitu penjumlahan dan pengurangan [10].

Suatu kriptosistem dikatakan bersifat aditif jika dan hanya jika :

$$\exists \Delta:\ \varepsilon(x1)\ \Delta\ \varepsilon(x2) = \varepsilon(x1 + x2)$$

Dengan $x1$ dan $x2$ adalah *plaintext*, $\varepsilon$ adalah fungsi enkripsi dan $\Delta$ adalah suatu operasi yang bergantung pada sifat algoritma enkripsi yang digunakan. Kemudian suatu kriptosistem dikatakan bersifat multiplikatif jika dan hanya jika :

$$\exists \Delta:\ \varepsilon(x1)\ \Delta\ \varepsilon(x2) = \varepsilon(x1.x2)$$

Terdapat dua jenis enkripsi homomorfik yaitu *partially homomorphic encryption* (PHE) dan *fully homomorphic encryption* (FHE). PHE merupakan jenis enkripsi homomorfik yang memungkinkan dilakukannya satu jenis operasi tertentu pada *ciphertext*. Sementara itu FHE merupakan jenis enkripsi homomorfik yang memungkinkan kedua jenis operasi penjumlahan dan perkalian dilakukan pada *ciphertext* [10].

### 1) *Partially Homomorphic Encryption*

Suatu *cryptosystem* dikatakan bersifat *partially homomorphic* jika *cryptosystem* tersebut memiliki salah satu dari sifat additif atau multiplikatif tapi tidak keduanya (Gentry C, 2009). Maksudnya dapat dilakukan salah satu operasi penjumlahan atau perkalian pada *ciphertext*. Beberapa contoh *cryptosystem* yang bersifat *partially homomorphic* yaitu RSA, ElGamal dan Pailier. RSA memiliki sifat multiplikatif, ElGamal memiliki sifat multiplikatif dan Pailier memiliki sifat aditif.

Sifat homomorfik algoritma RSA adalalah sebagai berikut. Diberikan dua buah *ciphertext* c1 dan c2

$$c1 = a^e \bmod n$$

$$c2 = b^e \bmod n$$

Lakukan perkalian pada c1 dan c2 sehingga diperoleh

$$c1.c2 = (a.b)^e \bmod n$$

Nilai tersebut sama dengan hasil enkripsi plaintext a x b.

Sifat homomorfik algoritma ElGamal adalah sebagai berikut. Diberikan dua buah *ciphertext* (c1,c2) dan (c3,c4).

$$c1 = g^{k1} \bmod p, \qquad c2 = y^{k1}.a \bmod p$$
$$c3 = g^{k2} \bmod p, \qquad c4 = y^{k2}.b \bmod p$$

Lakukan perkalian dua buah pasangan *ciphertext* sehingga diperoleh :

$$(c1, c2).(c3, c4) = (c1.c3\ , c2.c4)$$
$$= (g^{k1}.g^{k2}, y^{k1}.a.y^{k2}.b\ )$$
$$= (g^{k1+k2}, a.b.y^{k1+k2})$$

Nilai tersebut sama dengan hasil enkripsi plaintext a x b.

Sifat homomorfik algoritma Pailier adalah sebagai berikut. Diberikan *ciphetext* c1 dan c2

$$c1 = g^a r_1^n \bmod n^2$$
$$c2 = g^b r_2^n \bmod n^2$$

Dengan mengalikan c1 dan c2 maka akan diperoleh hasil :

$$c1.c2 = g^a r_1^n . g^b r_2^n \bmod n^2$$
$$= g^{a+b}(r_1 r_2)^n \bmod n^2$$

Hasil tersebut sama dengan enkripsi plaintext a+b

### 2) *Fully Homomorphic Encryption*

Suatu *cryptosystem* dikatakan bersifat *fully homomorphic* jika *cryptosystem* tersebut memiliki sifat additif atau multiplikatif [3]. Maksudnya terdapat operasi pada *ciphertext* yang dapat mewakili operasi penjumlahan dan pengurangan pada *plaintext*.

Skema FHE yang ada saat ini yaitu menggunakan *cryptosystem* yang dikembangkan oleh Craig Gentry [2,3] pada tahun 2009. Skema tersebut menggunakan *ideal lattice* untuk merepresentasikan kunci dan *ciphertext*-nya [6].

## B. *E-voting*

*Electronic voting* atau *e-voting* adalah penggunaan komputer atau komputerisasi pada proses pemungutan suara pada pemilihan [4]. Teknologi *electronic voting* dimulai pada tahun 1970 yang disebut teknologi pencatatan langsung secara elektronik atau lebih dikenal dengan istilah DRE (*direct recording electronic*). Cara memilih dengan sistem ini adalah dengan memilih kandidat yang sudah tercetak pada layar komputer. Pemilih hanya menekan tombol untuk memilih pilihan yang diinginkan.

*E-voting* dapat dilakukan pada suatu tempat tertentu contohnya tempat pemungutan suara dengan menggunakan DRE. Mekanisme ini sama dengan voting konvensional, hanya saja menggunakan alat elektronik untuk merekam pilihan dari pemilih. Terdapat juga mekanisme *remote voting* atau voting yang dilakukan oleh pemilih dari jarak jauh. Pada voting konvensional, *remote voting* ini biasanya dilakukan dengan mengirimkan surat pernyataan dari pemilih. Pada sistem *e-voting*, pemilihan dari jarak jauh lebih dimungkinkan dengan cara :

1. Pemilihan melalui telepon. Pemilihan ini dilakukan dengan cara pemilih menelepon nomor tertentu

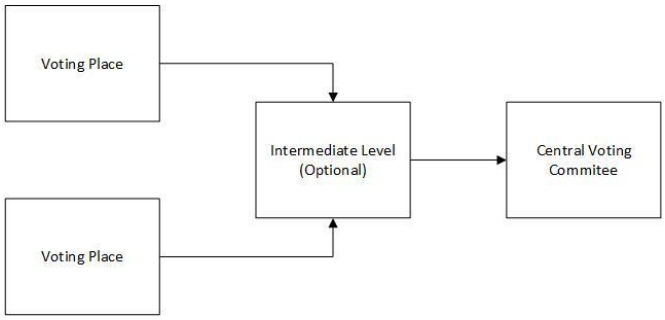
kemudian bot akan meminta masukan pilihan. Pemilih hanya perlu menekan tombol pada telepon lalu akan terekam dan disimpan. Kekurangan dari cara ini adalah tidak dapat melakukan banyak pemilihan dalam waktu bersamaan dan beberapa negara tidak memberlakukan enkripsi pada jaringan telepon sehingga rawan akan penyadapan.

2. Pemilihan melalui SMS. Pemilihan ini dilakukan dengan cara pemilih mengirimkan *username* dan PIN terlebih dahulu kemudian akan di konfirmasi oleh sistem. Jika berhasil maka sistem akan mengirimkan formulir pemilihan dan pemilih hanya perlu mengirimkan pilihannya lalu pilihan tersebut akan terekam oleh sistem. Masalah utama pada pemilihan model ini adalah kerahasiaan data pemilihan sebab beberapa negara tidak melakukan enkripsi terhadap jaringan telepon seluler.
3. Pemilihan melalui internet. Pemilih melakukan pemilihan melalui situs web tertentu. Pemilih harus *login* terlebih dahulu kemudian memilih melalui formulir yang disediakan. Data yang dikirim dienkripsi terlebih dahulu untuk menghindari kemungkinan penyadapan dan perubahan data

## III. PENELITIAN TERKAIT

Pada bab ini akan dipaparkan mengenai penelitian terkait yang yang digunakan sebagai acuan pada makalah ini.

### A. *Design and Development of Voting Data Security for Electronic Voting (E-voting)*




Gambar 1 Skema e-voting Djanali Supeno

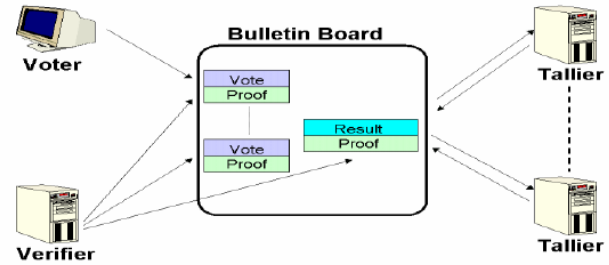
Sistem e-voting yang dijelaskan pada [1] mengadaptasi sistem pemilihan umum dalam skala yang luas, contohnya pemilihan umum di Indonesia. Terdapat beberapa level perhitungan hasil pemilihan mulai dari level paling bawah (*voting place*), kemudian level menengah (*intermediate level*) hingga level paling atas (*central voting comitee*).

*Voting place* merupakan bagian yang menerima hasil vote yang dilakukan oleh voter. Intermediate level merupakan bagian yang menerima hasil rekapituasi suara dari level yang ada di bawahnya. Central voting comite merupakan level paling atas yang mengakumulasikan hasil pemungutan suara secara keseluruhan. Skema e-voting yang digunakan dapat dilihat pada Gambar 1.

### B. *Secure E-voting using Homomorphic Technology*

Sistem e-voting yang diusulkan [12] menggunakan enkripsi homomorfik untuk melakukan perhitungan suara. Tahapan-tahapan pada sistem e-voting tersebut adalah :

1. Fase registrasi, yaitu pendaftaran identitas pemilih. Pemilih yang terdaftar akan mendapatkan *username* dan *password*.
2. Fase validasi, fase ini dilakukan saat pemilih akan melakukan pemilihan. Sistem akan mengecek apakah *username* dan *password* pemilih benar dan pemilih tersebut belum melakukan pemilihan.
3. Fase pemilihan, yaitu fase pemilihan kandidat oleh pemilih. Pemilih dan kandidat yang dipilih akan dienkripsi untuk menjaga kerahasiaan pemilihan.
4. Fase perhitungan, yaitu tahap perhitungan suara tiap kandidat.




Gambar 2 Skema e-voting dengan bulletin board

Sistem voting dengan enkripsi homomorfik yang diusulkan adalah sistem yang bekerja menggunakan model yang disebut bulletin board. Pada model tersebut terdapat empat komponen yaitu voter, verifier, tallier dan bulletin board. Semua informasi yang telah dikirim ke bulletin board dapat dibaca oleh semua komponen yang ada. Setiap user yang terautorisasi dapat menambahkan pesan pada areanya dan user lain tidak dapat menghapusnya. Skema tersebut dapat dilihat pada Gambar 2.

### C. *Analysis Partially Homomorphic Encryption and Fully Homomorphic Encryption*

Morris Liam pada makalahnya [6] membahas mengenai performa enkripsi homomorfik parsial dan enkripsi homomorfik penuh.

Skema partially homomorphic encryption menggunakan algoritma Pailier diuji dengan contoh studi kasus menjumlahkan data rekapitulasi jumlah pemilihan dari tiga

kandidat. Hasilnya adalah algoritma Pailier mendukung penjumlahan yang bersifat homomorfis pada dua buah ciphertext dengan mengalikan ciphertext tersebut. Waktu enkripsi berada pada orde 0.1 mili detik dan untuk operasi penjumlahan homomorfis memakan waktu 5 mili detik.

Pada skema *fully homomorphic encryption* dengan menggunakan *Gentry Cryptosystem*, operasi penjumlahan dan perkalian secara homomorfis dapat dilakukan. Namun masalahnya adalah ukuran kunci yang digunakan haruslah sangat besar agar aspek keamanan yang diharapkan bisa terpenuhi. Pada makalah tersebut ukuran *ciphertext* yang dihasilkan untuk parameter keamanan yang direkomendasikan adalah pada kisaran 128MB dan ukuran kunci publik 128PB. Aspek keamanan dapat diturunkan sehingga ukuran kunci publik hanya pada orde beberapa GB saja namun waktu untuk melakukan enkripsi tetap memakan waktu yang lama yaitu 30 menit untuk mengenkripsi satu bit.

## IV. ANALISIS DAN SOLUSI PERMASALAHAN

Bagian ini berisi analisis permasalahan berserta solusi dari permasalahan tersebut.

### *A. Permasalahan*

Sistem pemilihan elektronik atau *e-voting* memiliki kelebihan seperti kemudahan melakukan pemilihan, proses perhitungan dan rekapitulasi yang cepat, dapat menghemat anggaran untuk surat suara dan mengurangi kemungkinan kesalahan perhitungan. Namun dibalik keuntungan dan kemudahan yang ditawarkan sistem *e-voting* terdapat risiko yaitu ancaman dibidang keamanan. Risiko keamanan tersebut dapat terjadi pada proses *e-voting* antaran lain :

1. Keamanan data pemilih.
2. Keamanan pada proses pemilihan, yaitu bagaimana menjaga agar proses pemilihan berlangsung aman. Contohnya menjaga pemilihan agar tetap rahasia, menjaga agar pemilih terdaftar saja yang dapat memilih serta menjaga agar satu pemilih hanya dapat melakukan pemilihan sebanyak satu kali.
3. Keamanan saat rekapitulasi hasil, yaitu menjaga keamanan saat proses rekapitulasi seperti perhitungan suara dan rekapitulasi suara dari beberapa daerah pemilihan.
4. Keamanan pengiriman hasil pemilihan, yaitu menjaga keamanan saat dilakukan transmisi hasil pemilihan.

Dari permaslahan-permasalahan yang telah disebutkan, masalah yang akan menjadi lingkup bahasan pada makalah ini adalah masalah keamanan pada proses pemilihan yaitu bagaimana pilihan dari pemilih bersifat rahasia, keamanan saat rekapitulasi hasil pemilihan dan keamanan hasil pemilihan yang dikirim.

### *B. Solusi Permasalahan*

Permasalahan utama yang dibahas pada tugas akhir ini yaitu bagaimana menjaga keamanan data pemilihan pada saat proses pemilihan, saat rekapitulasi hasil pemilihan dan keamanan data hasil pemilihan yang dikirim. Untuk rekapitulasi hasil pemilihan perlu ditentukan skema enkripsi dan algoritma apa yang cocok untuk digunakan. Kemudian untuk menjaga keamanan pada proses pemilihan perlu ditentukan skema *e-voting* yang akan digunakan. Solusi untuk permasalahan-permasalahan tersebut akan dibahas lebih detail pada subbab selanjutnya.

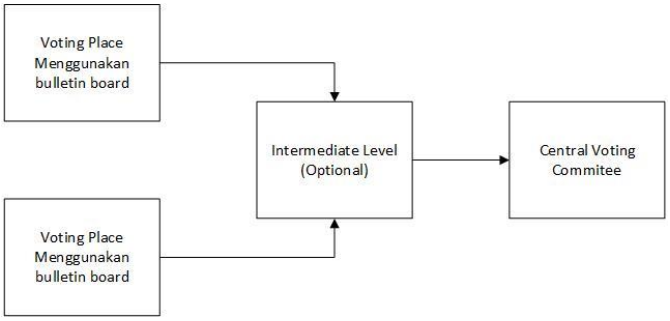
#### *1) Skema E-voting*

Rekapitulasi data hasil pemilihan dilakukan dengan mencari total suara tiap kandidat dari seluruh tempat pemilihan. Untuk data yang tidak terenkripsi tentu mudah untuk menghitung total suara tiap kandidat, kita hanya perlu menjumlahkannya langsung. Namun berbeda dengan data yang terenkripsi, untuk mencari jumlahnya kita harus mendekripsi data tersebut lalu menjumlahkannya kemudian mengenkripsinya lagi. Sama halnya dengan sistem *e-voting* yang dibuat oleh Djanali Supeno [1]. Kelemahan sistem tersebut adalah kunci untuk mendekripsi pesan harus tersebar ke seluruh *intermediate level* contohnya kabupaten/kota dan provinsi. Hal ini tentu menjadi risiko sebab semakin banyak yang mengetahui kuncinya maka akan semakin besar peluang kunci tersebut dicuri atau diketahui pihak lain yang tidak berhak.

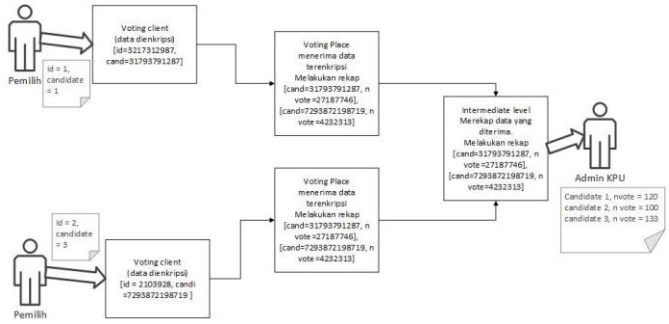
Tabel 1 Perbandingan skema e-voting Djanali dengan Shinde

| | **Algoritma Enkripsi** | **Kelebihan** | **Kekurangan** |
|---|---|---|---|
| Djanali Supeno dkk | RSA | Adaptif terhadap perbedaan metode pemilihan yang digunakan | kunci untuk mendekripsi pesan harus tersebar ke seluruh *intermediate level* |
| Shinde Shubhangi dkk | Pailier | Memungkinkan rekapitulasi hasil pemilihan tanpa perlu melakukan dekripsi terlebih dahulu terhadap data | Tersentralisasi, tidak adaptif terhadap perbedaan metode pemilihan |

Shinde Shubhangi dkk [12] mengembangkan sistem *e-voting* yang memanfaatkan enkripsi homomorfik. Sistem ini memungkinkan rekapitulasi hasil pemilihan tanpa perlu melakukan dekripsi terlebih dahulu terhadap data. Namun kekurangannya adalah sistem *e-voting* yang dikembangkan tersentralisasi. Hal tersebut tidak cocok diterapkan pada pemilihan dengan jumlah pemilih yang banyak dengan wilayah pemilihan yang banyak. Kedua skema *e-voting* tersebut dapat dilihat pada Tabel 1.




Gambar 3 Skema e-voting yang diusulkan




Gambar 4 Ilustrasi proses pemungutan suara

Untuk menutupi kelemahan dari kedua sistem *e-voting* tersebut kita dapat menggabungkan keduanya sehingga dapat saling menutupi kelemahan. Kita dapat menambahkan sifat homomorfis pada sistem *e-voting* Djanali Supeno [1] dengan menggunakan *bulletin board* pada *voting place* seperti yang dikembangkan Shinde Shubhangi dkk [12]. Gabungan dari kedua skema *e-voting* tersebut dapat dilihat pada Gambar 3.

Pada sisi klien, voter melakukan *vote* kepada suatu kandidat lalu data mengenai *vote* yang diberikan oleh voter dienkripsi kemudian dikirim ke *voting place*. Di *voting place*, data yang diterima adalah data dalam bentuk terenkripsi. Dari data terenkripsi tersebut dilakukanlah rekapitulasi hasil pemungutan suara di *voting place*. Proses rekapitulasi ini menggunakan penjumlahan secara homomorfik jadi data yang akan direkapitulasi tidak didekripsi sama sekali. Hasil rekapitulasi tersebut juga merupakan data dalam bentuk terenkripsi. Kemudian hasil rekapitulasi di *voting place* akan dikirim ke *intermediate level*. Disana dilakukan juga proses rekapitulasi terhadap data-data yang diterima dari *voting place* – *voting place* yang ada di bawahnya. Proses rekapitulasi yang dilakukan melibatkan penjumlahan homomorfik sama dengan proses yang terjadi di *voting place.* Ilustrasi tersebut dapat dilihat pada Gambar 4.

*2) Algoritma Enkripsi Homomorfik*

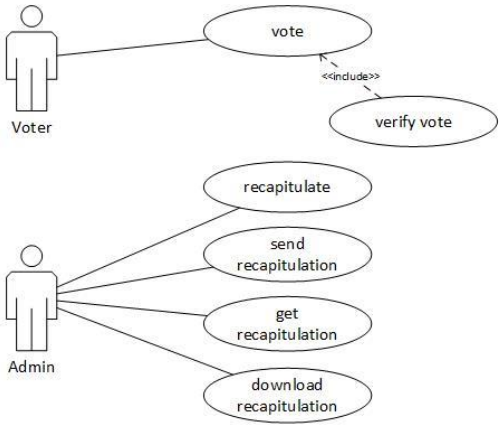
Pada sistem *e-voting*, skema enkripsi yang digunakan harus mendukung penjumlahan homomorfis pada *ciphertext*. Pada skema PHE, kita dapat memanfaatkan PHE dengan algoritma Pailier yang memiliki sifat additif homomorfik. Selain PHE Pailier, skema FHE juga memenuhi syarat tersebut sebab FHE mendukung perkalian dan penjumlahan homomorfik.

Berdasarkan penelitian pada [6], FHE saat ini masih membutuhkan *resource* yang sangat besar untuk melakukan enkripsi dan dekripsi sehingga tidak efisien untuk digunakan. Sementara itu PHE dengan algoritma Pailier memberikan hasil yang cukup memuaskan dalam hal enkripsi, dekripsi serta penjumlahan homomorfik. Walaupun PHE dengan algoritma Pailier hanya mendukung penjumlahan homomorfik saja, hal tersebut sudah cukup sebab rekapitulasi data hanya membutuhkan operasi penjumlahan saja. Jadi skema enkripsi yang dipilih adalah skema *partially homomorphic encryption* dengan algoritma Pailier.

## V. RANCANGAN SOLUSI

Bab ini berisi rancangan solusi berupa rancangan perangkat lunak yang akan dibangun. Rancangan solusi dibangun berdasarkan solusi permasalahan yang diuraikan sebelumnya. Rancangan solusi ini akan diterapkan menggunakan skema e-voting dan algoritma enkripsi homomorfik yang telah diuraikan pada sub bab sebelumnya.

### *A. Rancangan Use Case*




Gambar 5 Rancangan Use Case

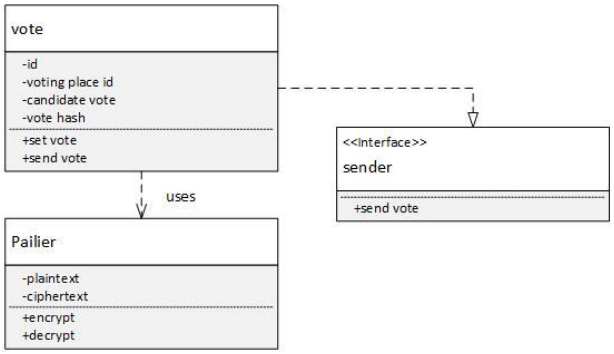
Perangkat lunak yang dikembangkan menggunakan metode pengembangan perangkat lunak berorientasi objek. Diagram *use case* perangkat lunak dapat dilihat pada Gambar 5.

Pada diagram *use case* terdapat dua aktor yaitu admin dan *voter*. *Voter* atau pemilih merupakan aktor yang dapat melakukan *vote* atau peserta pemilihan. Admin merupakan administrator perangkat lunak yang dalam konteks pemilihan merupakan panitia penyelenggara pemilihan.

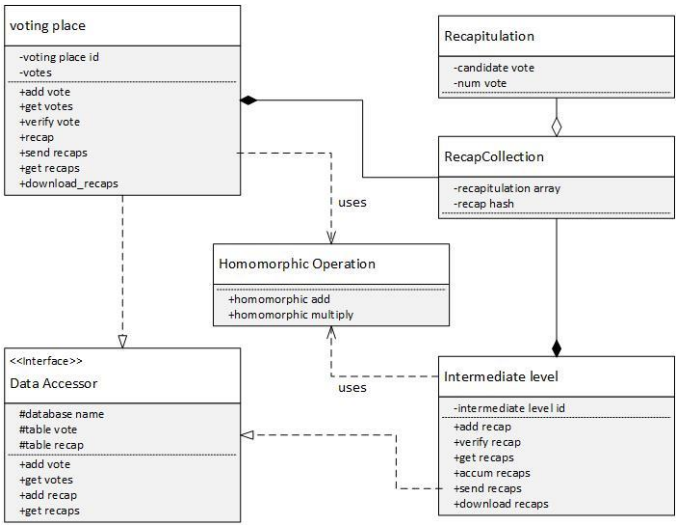
### *B. Rancangan Kelas*

Terdapat dua komponen yang akan dibangun. Komponen yang pertama yaitu komponen berjalan pada sisi klien yang berfungsi untuk mengenkripsi dan mengirimkan data pemungutan suara ke *voting place* . Diagram kelas komponen

tersebut dapat dilihat pada Gambar 6. Komponen yang kedua yaitu komponen yang berjalan pada sisi server. Komponen tersebut bertugas untuk mengumpulkan data pemungutan suara, memverifikasi kemudian melakukan perhitungan rekapitulasi data. Diagram kelas komponen tersebut dapat dilihat pada Gambar 7.




Gambar 6 Rancangan kelas klien




Gambar 7 Rancangan kelas server

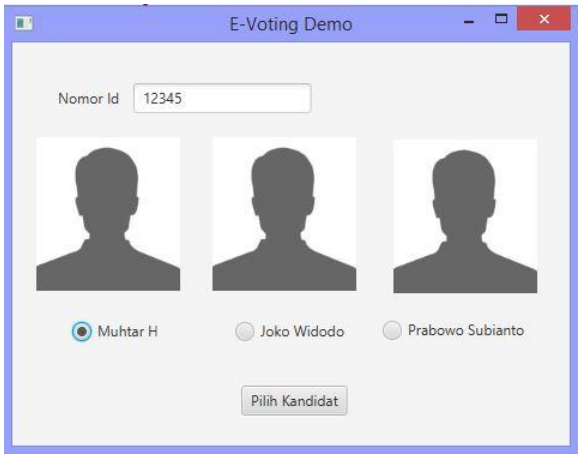
## VI. IMPLEMENTASI APLIKASI E-VOTING

Bagian ini berisi implementasi hasil perancangan aplikasi e-voting yang telah dilakukan sebelumnya. Hasil implementasi berupa aplikasi *e-voting* dan *file* konfigurasi yang berfungsi untuk menyimpan parameter-parameter yang digunakan terutama parameter enkripsi homomorfik.

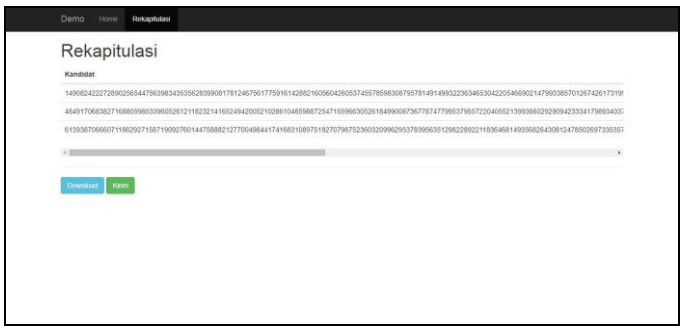
### *A. Aplikasi E-Voting*

Aplikasi *e-voting* yang dibangun berupa *e-voting* dengan tiga kandidat. Aplikasi tersebut terdiri dari tiga bagian. Bagian *client*, *voting place* dan *intermediate level*. Aplikasi tersebut dibangun berdasarkan rancangan *use case* dan rancangan kelas pada bab sebelumnya.

Bagian klien merupakan bagian yang berinteraksi langsung dengan pemilih. Bagian ini menerima masukan yang berupa data pemilihan yang dilakukan oleh pemilih. Data yang telah diterima kemudian dienkripsi menggunakan enkripsi homomorfik Pailier. Nomor *id* pemilih dienkripsi menggunakan variabel acak yang dibangkitkan di dalam aplikasi. Kandidat yang dipilih dienkripsi dengan algoritma yang sama tetapi menggunakan variabel acak yang telah ditentukan di *file* konfigurasi. Hal ini dilakukan agar pada saat rekapitulasi, kandidat yang terenkripsi tersebut dapat diidentifikasi. Setelah masukan diterima kemudian klien akan mengirim data voting yang terlebih dahulu dienkripsi ke *voting place* . Bagian *client* dibuat GUI untuk memudahkan melakukan pemilihan. Bagian ini dibuat dengan bahasa pemrograman Java dan menggunakan JavaFX untuk membuat GUI. Tampilan *client* dapat dilihat pada gambar 8.



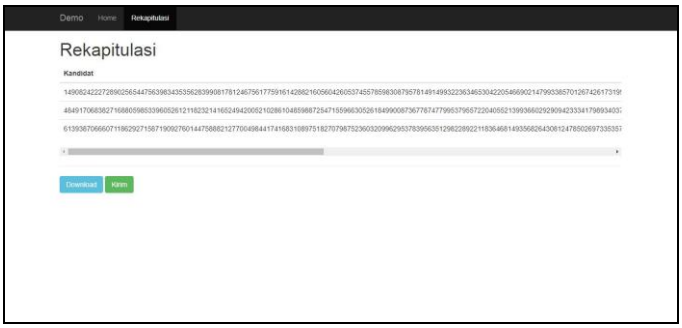

Gambar 8 Tampilan klien




Gambar 9 Tampilan voting place

Bagian *voting place* merupakan realisasi dari *voting place* yang dimaksudkan sebelumnya. Bagian ini menerima data hasil *vote* dari pemilih kemudian menyimpan ke basis data. Hasil *vote* yang diterima dan disimpan merupakan data dalam yang masih dalam bentuk terenkripsi. Bagian ini juga berfungsi untuk melakukan rekapitulasi dari *vote* yang telah dilakukan. Rekapitulasi dilakukan pada data yang masih dalam bentuk *ciphertext* menggunakan operasi penjumlahan homomorfik yang telah disediakan oleh aplikasi. Komponen

*Voting place* ini dibuat menggunakan *framework spring* untuk memudahkan implementasi. Contoh tampilan *voting place* yang dibuat dapat dilihat pada Gambar 9.




Gambar 10 Tampilan Intermediate level

Bagian ketiga adalah *intermediate level* yang pada pengujian kali ini juga sekaligus berperan sebagai *central voting comitee*. Bagian ini menerima hasil rekapitulasi dalam bentuk terenkripsi dari *voting place* kemudian melakukan rekapitulasi dari data-data yang diterima. Rekapitulasi dilakukan dengan menggunakan operasi penjumlahan homomorfik memanfaatkan aplikasi yang telah dibuat. Bagian ini dibuat menggunakan *framework spring* untuk memudahkan implementasi. Contoh tampilan *intermediate level* yang dibuat dapat dilihat pada Gambar 10.

### B. *File Konfigurasi*

Selain source code aplikasi, dihasilkan pula sebuah file konfigurasi. *File* ini berisi parameter-parameter yang dibutuhkan contohnya kunci privat dan kunci publik. Detail *file* konfigurasi dapat dilihat pada Tabel 2

Tabel 2 Parameter file konfigurasi

| No | Nama | Deskripsi |
|---|---|---|
| 1 | n | Nilai n pada kunci publik |
| 2 | nSquared | Nilai n kuadrat |
| 3 | g | Nilai g pada kunci publik |
| 4 | bits | Panjang bit kunci |
| 5 | Upperbound | Batas atas bilangan yang dapat dienkripsi |
| 6 | lambda | Nilai *lambda* $(\lambda)$ pada kunci privat |
| 7 | denominator | Nilai *denominator* $(\mu)$ pada kunci privat |
| 8 | hm_zero | *Ciphertext* untuk angka 0 |
| 9 | hm_one | *Ciphertext* untuk angka 1 |
| 10 | r | Nilai variabel acak yang digunakan |

Contoh penulisan file konfigurasi

```
#Private
lambda=167174079359597543689366628318212355698162582
623546069880037006481739518536O
denominator=331242388831021437001435260935073560180 6
8147908271642036299388431667134332240
```

```
#Public
n=735565949182229192233213164600134365077477597830008
724917507659118650212054869
nSquared=541057265596353778474261219978751436703533 8
925700698210557454329060182105932046866493295661467 4
608967881186864998784053520249730227443040073809674 6
6607161
g=363742120874166105715853703388525996027600141837 65
128623588690810268173338 82
bits=1024
#UpperBound
upperBound=2147483647
#number
hm_one=206852069678357392260893601789277438943203366
775067694360763599460818658638606998284879464881071 6
150864031608440876905099020792584814575559597759092 35
17262
hm_zero=353514319079847035804758882645720615739540 71
789187691510941246251988695649234562592270036203937 0
377887674736876153125791706085466719122843259085558 3
18466
```

## VII. PENGUJIAN

Bab ini berisi pengujian aplikasi yang dilakukan. Pengujian dilakukan dengan dua tahapan yaitu pengujian fungsionalitas aplikasi dan pengujian kualitas keamanan dengan metode EVSSO.

Pengujian fungsionalitas dilakukan dengan menggunakan metode *blackbox*. Aplikasi akan diuji dengan kasus-kasus uji yang telah dibuat berdasarkan *use case* yang telah didefinisikan. Hasi pengujian fungsionalitas menunjukkan bahwa aplikasi sudah lulus uji fungsionalitas ini.

### A. *Pengujian EVSSO*

Pengujian dengan EVSSO (*Electronic Voting System Security Optimization*) akan digunakan pada pengujian kali ini. Metode pengujian ini dipilih karena dapat menguantifikasikan kualitas keamanan pada sistem *e-voting*. Metode pengujian tersebut bersifat generik dan dapat diterapkan pada sistem *e-voting* yang berbeda-beda. Selain mengukur keamanan, metode ini juga dapat memberikan kita rekomendasi bagian mana dari sistem *e-voting* yang menjadi prioritas untuk diperbaiki demi menambah kualitas keamanan.

Pengujian dengan metode EVSSO dilakukan dengan menggunakan matriks EVSSO. Matriks ini berisi *core area* yaitu aspek keamanan yang diuji dan *maturity level* yang berarti posisi pada tiap level keamanan. Evaluasi dilakukan berdasarkan kriteria-kriteria yang telah ditentukan. Terdapat tiga level yaitu level A, B dan C. Matriks EVSSO dan kriteria evaluasinya dapat dilihat pada lampiran C.

Secara umum *core area* pada matriks EVSSO terbagi atas tiga yaitu *hardware*, *software* dan *human factor*. Pada pengujian ini hanya diambil *core area software*. Alasannya karena tugas akhir ini hanya berfokus pada pengembangan perangkat lunak aplikasi *e-voting* saja. *Hardware* dan sumber daya manusia pelaksana yang digunakan oleh sistem *e-voting* berada di luar skop pembahasan tugas akhir. Kemudian pada core area software, kategori *software engineering* tidak dimasukkan

karena tugas akhir ini tidak membahas mengenai metode pengembangan *software* yang digunakan.

Hasil pengujian EVSSO yang dilakukan dapat dilihat pada Tabel 3.

Tabel 3 Hasil Pengujian EVSSO

| Core Area | Maturity Level | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Software - Compliance with Election Principles | | A | | | | B | | | | | |
| Software - Data integrity | | | A | | | B | | | | | |
| Software - Cryptography | | | A | | | | B | | | C | |
| Software - Transparency | | | | A | | | | B | | | |
| Software - Protection of Software | | | | A | | | | | B | | C |

## VIII. EVALUASI

Berdasarkan pengujian yang telah dilaksanakan selama pengujian aplikasi *e-voting* pada sistem *e-voting* sederhana yang dibuat, diperoleh evaluasi sebagai berikut :

1. Berdasarkan hasil kebutuhan fungsional, dapat ditarik kesimpulan bahwa aplikasi e-voting yang dibuat sudah mencakup *use case – use case* yang telah didefinisikan sebelumnya. Aplikasi tersebut dapat digunakan untuk membangun sistem *e-voting*. Pada pengujian dilakukan integrasi aplikasi *e-voting* dengan *framework spring*. Dengan memanfaatkan aplikasi *e-voting*, pembangunan sistem *e-voting* menjadi lebih sederhana. Contohnya untuk membangun sistem sederhana seperti yang digunakan pada pengujian, hanya membutuhkan tambahan dua kelas *controller* dan sebuah kelas yang berisi definisi data yang menyatakan model pada basis data.
2. Berdasarkan matriks EVSSO pada Tabel 2 terlihat bahwa hal yang menjadi prioritas pengembangan saat pembuatan perangkat lunak e-voting yang memanfaatkan aplikasi yang telah dibuat adalah *data integrity* level B. Hal tersebut menjadi prioritas sebab pada matriks EVSSO menunjukkan bahwa level B pada *data integrity* berada pada *maturity level* dengan nilai 5, paling kecil di antara yang lainnya. Hal yang dapat dilakukan adalah melakukan proteksi untuk menjaga *reliability* data. Hal tersebut dapat dilakukan dengan melakukan *backup* data secara berkala di *non-volatile storage* contohnya *hard disk*. Hal tersebut dapat ditangani salah satunya dengan menggunakan DBMS (*Database Management System*) untuk menyimpan data.
3. Enkripsi dengan algoritma Pailier menggunakan variabel acak sehingga *plaintext* yang sama apabila dienkripsi dapat menghasilkan *cipher text* yang berbeda. Hal tersebut menjadi masalah pada saat rekapitulasi data sebab jumlah suara akan dihitung berdasarkan *id* kandidat. Permasalahan tersebut dapat diatasi dengan mengatur nilai variabel acak pada saat enkripsi untuk kasus-kasus yang memerlukan hasil enkripsi yang konsisten untuk tiap *plaintext.*
4. Operasi perhitungan suara lebih lambat dengan menggunakan enkripsi homomorfik menjadi lebih lambat dibandingkan dengan jika data tidak terenkripsi. Sebagai perbandingan, operasi penjumlahan secara homomorfik rata-rata memakan waktu 45330 ns, sedangkan operasi penjumlahan biasa hanya memakan waktu rata-rata 482 ns. Hal ini disebabkan karena operasi penjumlahan homomorfik sama halnya dengan perkalian dua buah *big integer*. Terlebih lagi *big integer* yang digunakan sangat besar yakni 1024 bit. Hal tersebut menjadi masalah jika sistem *e-voting* yang dibuat menampilkan hasil pemilihan secara *real time* dengan jumlah pemilih yang besar.
5. Aplikasi *e-voting* yang dibuat belum menyediakan fitur autentikasi dan autorisasi. Fitur autentikasi dan autorisasi dapat dibuat terpisah dengan sistem *e-voting* yang dibuat.

## IX. KESIMPULAN

Bersasarkan hasil pengerjaan dan pengujian aplikasi yang telah dilakukan dapat diambil kesimpulan bahwa :

1. Enkripsi homomorfik parsial yang bersifat *additive* dapat diterapkan pada sistem pemilihan elektronik (*e-voting*). Enkripsi homomorfik yang bersifat *additive* dipilih karena operasi perhitungan suara pada pemilihan menggunakan operasi penjumlahan. Selain itu enkripsi homomorfik memungkinkan tidak dilakukannya dekripsi data terlebih dahulu ketika melakukan rekapitulasi hasil pemilihan
2. Aplikasi yang dibangun telah menyediakan fitur keamanan yang baik dari sisi kesesuaian dengan asas-asas pemilihan dan dari sisi kriptografi. Kekurangan yang masih dialami adalah belum menyediakan fitur untuk *reliability* data.

## REFERENSI


[1] Djanali Supeno dkk (2016). *Design and Development of Voting Data Security for Electronic Voting (E-voting).* Institut Teknologi Sepuluh Nopember.

[2] Gentri C., Halevi S (2011). Implementing Gentry's Fully-Homomorphic Encryption Scheme. IBM Research.

[3] Gentry, Craig (2009). *A Fully Homomorphic Encryption Scheme.* Stanford University.

[4] Kahani, M. (2005), *Experiancing small-scale e-democracy in Iran.* The Electronic Journal On Information System in Developing Contries.

[5] Lauther, Kristin dkk (2011). *Can Homomorphic Encryption Be Practical?*. Microsoft.

[6] Morris, Liam (2013). *Analysis of Partial and Fully Homomorphic Encription.* Rochester Institute of Technology.

[7] Munir, Rinaldi (2005). *Diktat Kuliah IF5051 Kriptografi*. Departemen Teknik Informatika Institut Teknologi Bandung.

[8] Ondrisek, Barbara (2009). *E-Voting Security Optimization*. Vienna University of Technology.

[9] Pailier, Pascal (1999). Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Gemplus Card International.

[10] Potzelsberger (2013). KV Web Security: Application of Homomorphic Encryption. http://www.fim.uni-linz.ac.at/lva/Web_Security/Abgaben/Poetzelsberger-Homomorphic.pdf . Diakses pada 15 Oktober 2016

[11] Schneier, Bruce (1999). *Applied Cryptography 2nd edition*, John Wiley & Sons.

[12] Shinde Shubhangi dkk (2013). *Secure E-voting Using Homomorphic Technology*. Terna Engineering College.

[13] Sophan, M (2012). Design Model TPS Dalam Sistem Pemilihan Kepala Daerah. Universitas Trunojoyo Madura.