IMPLEMENTASI ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) 128 BIT UNTUK PENGAMANAN PESAN TEXT DALAM EMAIL

M. Irwan Ukkas [1)], Eka Arriyanti [2)], Relando Tri Hanggara [3)]

[1,2,3]Teknik Informatika, STMIK Widya Cipta Dharma
[1,2,3] Jl. M. Yamin No.25, Samarinda, 75123
E-mail : Irwan212@yahoo.com[1)], eka_arriyanti@yahoo.com [2)], relandoth@gmail.com[3)]

ABSTRAK

Penelitian dilakukan untuk dapat membuat sebuah aplikasi yang dapat enkripsi dan dekripsi pesan teks dalam *email* yang nantinya jika penelitian ini berhasil bisa membantu pengguna aplikasi ini dalam mengirim pesan secara rahasia tanpa takut pesan tersebut diketahui oleh orang yang tidak dikehendaki.

Dalam penelitian ini algoritma yang digunakan untuk enkripsi dan dekripsi adalah *Advanced Encryption Standard* (AES) 128 bit, dan metode pengembangan sistem yang digunakan yaitu *Waterfall* dengan perangkat lunak pendukung yang digunakan adalah *Visual Basic.Net* **.**

Adapun hasil akhir dari penelitian ini yakni berupa aplikasi pengamanan pesan teks dalam *email* berbasis *desktop* yang dapat merahasiakan sebuah pesan aslinya sehingga apabila pesan itu terbaca oleh orang yang tidak dikehendaki orang tersebut tidak akan mengetahui pesan sebenarnya karena telah terenkripsi.

Kata Kunci: *Kriptografi, Advanced Encryption Standard, AES, Enkripsi, Dekripsi*

---

1. PENDAHULUAN

Perkembangan teknologi yang begitu cepat membuat manusia dapat saling berkomunikasi dan bertukar informasi atau data, tanpa harus terhalangi oleh jarak dan waktu. Meskipun demikian, tidak semua perkembangan teknologi komunikasi memberikan dampak yang positif dan menguntungkan. Salah satu dampak negatif dalam perkembangan teknologi adalah adanya penyadapan data yang merupakan salah satu masalah yang paling ditakuti oleh para pengguna jaringan komunikasi. Oleh karena itu, dibutuhkan suatu metode yang dapat menjaga kerahasiaan informasi.

Kriptografi adalah salah satu metode untuk menjaga kerahasiaan informasi. Dalam kriptografi, terdapat dua konsep utama, yaitu enkripsi dan dekripsi. Enkripsi adalah suatu proses, di mana informasi atau data yang akan dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagaimana informasi awalnya dengan menggunakan suatu algoritma tertentu. Hasil enkripsi disebut *ciphertext*.

Advanced Encryption Standard (AES) merupakan algoritma kriftografi berupa blok *ciphertext* simetrik. AES dengan kunci kriptografi 128, 192, dan 256 bit dapat digunakan untuk mengenkripsi dan mendekripsi data pada blok 128 bit. Berdasarkan studi terdahulu terhadap beberapa jurnal, AES dianggap unggul dalam keamanan, kecepatan, dan karakteristik algoritma beserta implementasinya.

Dengan demikian, penelitian ini akan difokuskan pada pengembangan program aplikasi kriptografi untuk pengamanan pesan *text* (kata atau kalimat) dalam surat elektronik (email) dengan algoritma AES untuk menjaga kerahasiaan pesan (*text*) dalam surat elektronik (email), sehingga tidak dapat diketahui oleh pihak yang tidak berhak (*anauthorized persons*) dan memudahkan pengguna untuk mengirimkan surat elektronik (*email*) secara rahasia dan pribadi.

2. RUANG LINGKUP PENELITIAN

Dalam penelitian ini permasalahan mencakup:

1. Aplikasi yang dikembangkan akan dapat berfungsi sebagai pengirim *email* dan sebagai pendekripsi pesan *text* yang terenkripsi.
2. Aplikasi akan berfungsi untuk pemilik akun *email Gmail*, *Yahoo*, *Live* dan *Outlook*.
3. Untuk pengguna akun *Gmail*, *setting* (pengaturan) *email* mengijinkan aplikasi lain dapat mengakses pengiriman email.
4. *Text* yang dienkripsi dan dekripsi merupakan *text* hasil *input*-an *keyboard* bukan *attachment file*.
5. Kunci enkripsi sama dengan kunci dekripsi.
6. Panjang kunci maksimal 16 karakter
7. Diperlukan *Net Framework* 4.5 agar aplikasi dapat berjalan dengan baik.
8. Aplikasi dikembangkan dengan bahasa pemprograman *Visual Basic.Net*.

3. BAHAN DAN METODE

Adapun bahan dan metode yang digunakan dalam membangun sistem ini yaitu:

3.1 Kriptografi

Menurut Dony (2008), kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti rahasia (*secret*), dan *Graphia* berarti tulisan (*writing*). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain.

Dan menurut Munir (2006), kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti maknanya.

Sehingga dapat disimpulkan kriptografi adalah ilmu yang mempelajari tentang pengacakan pesan dengan fungsi matematika agar tidak dapat dibaca oleh pihak yang tidak berwenang.

Menurut Dony (2008), pada dasarnya kriptografi terdiri dari beberapa komponen, yaitu :

1. Enkripsi

Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Pesan asli yang disebut *plaintext*, yang bisa di ubah menjadi kode-kode yang tidak dimengerti.

2. Dekripsi

Dekripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi atau diacak dikembalikan ke bentuk asalnya.

3. Kunci

Kunci adalah sandi yang dipakai untuk melakukan proses enkripsi atau dekripsi. Kunci terbagi atas dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

4. *Ciphertext*

Ciphertext merupakan suatu pesan yang telah melalui proses enkripsi. Pesan ini tidak dapat dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti).

5. *Plaintext*

Plaintext merupakan pesan sebenarnya yang ditulis, diketik atau dibuat dan memiliki makna. Pesan inilah yang nantinya akan diproses menggunakan algoritma kriptografi untuk menjadi pesan berkode atau *ciphertext*.

6. Pesan

Pesan merupakan data atau informasi yang dikirim (melalui kurir, saluran komunikasi data) atau yang di simpan di dalam media penyimpanan (kertas berupa arsip, *harddisk*, USB, dan sebagainya).

7. *Cryptanalysis*

Cryptanalysis diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan pesan asli tanpa harus mengetahui kunci sebenarnya. Jika suatu pesan berhasil diubah tanpa menggunakan kunci sebenarnya, proses tersebut dinamakan *breaking code*. Hal ini dilakukan oleh para kriptanalis.

3.2 *Advanced Encryption Standard* (AES)

Data *Encryption Standard* (DES) mungkin akan berakhir masa penggunaannya sebagai standar enkripsi kriptografi simetri. Data *Encryption Standard* (DES) dianggap sudah tidak aman lagi karena dengan perangkat keras khusus kuncinya bisa ditemukan dalam beberapa hari.

National Institute of Standards and Technology (NIST), sebagai agensi Departemen Perdagangan Amerika Serikat mengusulkan kepada Pemerintah Federal Amerika Serikat untuk sebuah standard kriptografi yang baru.

Untuk menghindari kontroversi mengenai standard yang baru tersebut sebagaimana pada pembuatan DES (NSA sering dicurigai mempunyai "pintu belakang" untuk mengungkap *ciphertext* yang dihasilkan oleh DES tanpa mengetahui kunci), maka NIST mengadakan sayembara terbuka untuk membuat algoritma kriptografi yang baru sebagai pengganti DES. Standar tersebut kelak diberi nama *Advanced Encryption Standard* (AES).

Persyaratan yang diajukan oleh NIST tentang algoritma yang baru tersebut adalah:

1. Algoritma yang ditawarkan termasuk ke dalam kelompok algoritma kriptografi simetris berbasis *cipher* blok.
2. Seluruh rancangan algoritma harus publik (tidak dirahasiakan)
3. Panjang kunci fleksibel : 128, 192, dan 256 bit.
4. Ukuran blok yang dienkripsi adalah 128 bit.
5. Algoritma dapat diimplementasikan baik sebagai *software* maupun *hardware*.

NIST menerima 15 proposal algoritma yang masuk. Konferensi umum pun diselenggarakan untuk menilai keamanan algoritma yang diusulkan.

Pada bulan Agustus 1998, NIST memilih 5 finalis yang didasarkan pada aspek keamanan algoritma, keefisiensian (*efficiency*), fleksibilitas, dan kebutuhan memori (penting untuk *embedded system*). Finalis tersebut adalah :

1. Rijndael (dari Vincent Rijmen dan Joan Daemen – Belgia, 86 suara)
2. Serpent (dari Ross Anderson, Eli Biham, dan Lars Knudsen – Inggris, Israel, dan Norwegia, 56 suara)
3. Twofish (dari tim yang diketuai oleh Bruce Schneier – USA, 31 suara)
4. RC6 (dari Laboratorium RSA – USA, 23 suara)
5. MARS (dari IBM, 13 suara)

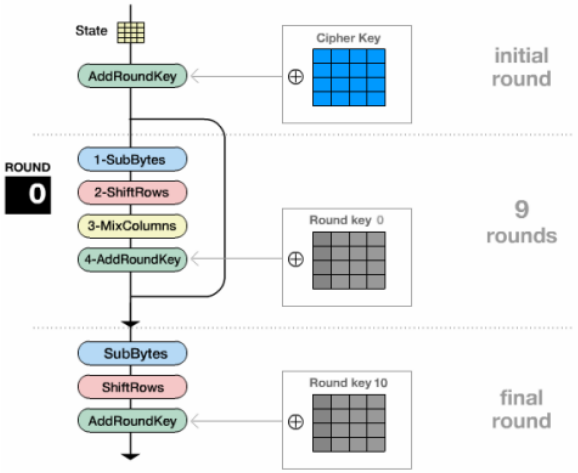
Pada bulan Oktober 2000, NIST mengumumkan untuk memilih Rijndael (dibaca : Rhine-doll), dan pada bulan November 2001, Rijndael ditetapkan sebagai AES, dan diharapkan Rijndael menjadi standard kriptografi yang dominan paling sedikit 10 tahun.

3.2.1 Proses Enkripsi *Advanced Encryption Standard* (AES)

Proses enkripsi *Advanced Encryption Standard* (AES) 128 Bit adalah sebagai berikut :

1. Pada awal enkripsi, *input plaintext* 128 bit akan disalinkan ke suatu *array* yang diberi nama *state array*.
2. Proses enkripsi dimulai dengan suatu proses yang melakukan XOR antara *state* awal (*plaintext*) dengan *cipher key*. Tahap ini disebut juga *initial round* .
3. Untuk kunci *cipher* 128 bit, jumlah ronde adalah 10.

4. Proses selanjutnya 9 ronde yang masing-masing ronde terdiri dari urut-urutan empat macam proses yaitu :
   1) *SubBytes* yaitu substitusi *byte* dengan menggunakan tabel substitusi (S-*box*).
   2) *ShiftRows* yaitu pergeseran baris-baris *array state* secara *wrapping*.
   3) *MixColumns* adalah mengacak data di masing-masing kolom *array state*.
   4) *AddRoundKey* adalah melakukan XOR antara *state* sekarang *round key*.
5. Diakhiri dengan ronde ke 10 yang hanya terdiri dari 3 proses yaitu *SubBytes*, *ShiftRows* dan *AddRoundKey*.



Gambar 1. Diagram Proses Enkripsi AES

Langkah kerja dari proses enkripsi sebagai berikut :

1. *SubBytes*

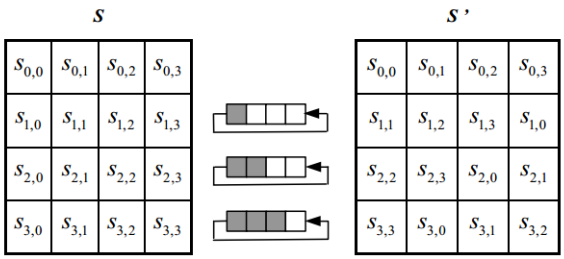
Transformasi *SubBytes* adalah memetakan setiap *byte* dari *array state* dengan menggunakan tabel substiusi S-*Box*.

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 2. Tabel S-*Box*

2. *ShiftRows*

Transformasi *ShiftRows* melakukan pergeseran secara *wrapping* pada 3 baris terakhir dari *array state*. Jumlah pergeseran bergantung pada nilai baris (r). Baris r = 1 digeser sejauh 1 *byte*, baris r = 2 digeser sejauh 2 *byte*, dan baris r = 3 digeser sejauh 3 *byte* . Baris r = 0 tidak digeser.

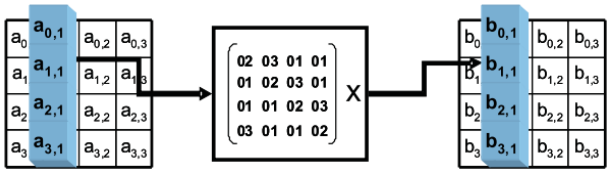


Gambar 3. Transformasi *ShiftRows*

3. *MixColumns*

Transformasi *MixColumns* merupakan proses mengalikan setiap kolom dari *array state* dengan polinomial a(x) mod (x4+ 1). Setiap kolom diperlakukan sebagai polinomial 4-suku pada GF(28).

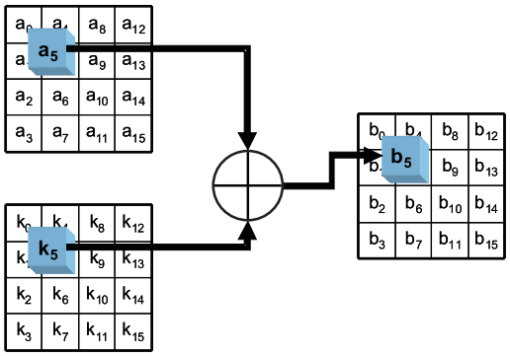
a(x) yang ditetapkan adalah : a(x) = {03}x3+ {01} x2+ {01}x+ {02}



Gambar 4. Transformasi MixColumns

4. *AddRoundKey*

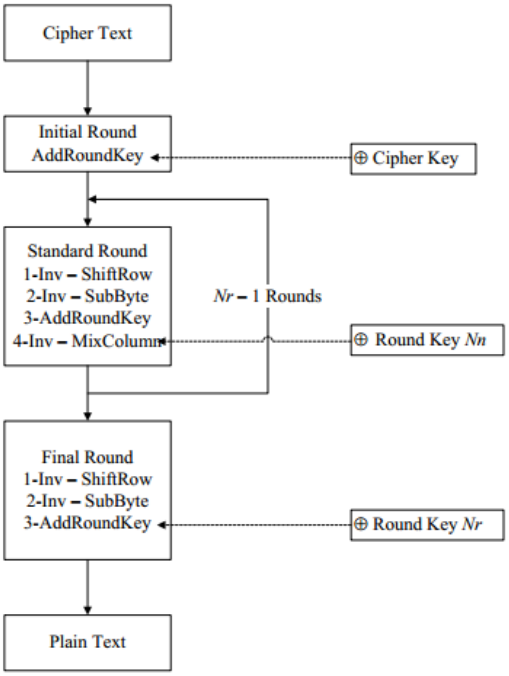
Transformasi *AddRoundKey* ini melakukan operasi XOR terhadap sebuah *round key* dengan *array state*, dan hasilnya disimpan di *array state*.



Gambar 5. Transformasi *AddRoundKey*

3.2.2 Proses Dekripsi *Advanced Encryption Standard* (AES)

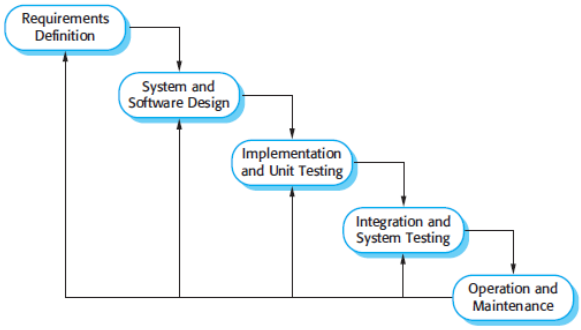
Urutan proses dekripsi *Advanced Encryption Standard* (AES) merupakan kebalikan dari proses enkripsinya. Namun ada proses yang dipertukarkan urutannya, namun penggunaan kuncinya sama. Jika urut-urutan proses enkripsi adalah *SubBytes*, *ShiftRows*, *MixColumns* dan *AddRoundKey*. Keempat proses tersebut adalah masing-masing *invers* pada proses enkripsi tetapi urutan adalah *InvShiftRows* dan *InvSubBytes* dipertukarkan dan *InvAddRoundKey* dan *InvMixColumns* juga dipertukarkan.



Gambar 6. Diagram Proses Dekripsi AES

3.3 *Waterfall*

Menurut Ian Sommerville (2011), model sekuensial linier (*waterfall*) atau air terjun merupakan paradigma rekayasa perangkat lunak yang paling tua (klasik) dan paling banyak dipakai. Model ini mengusulkan sebuah pendekatan pengembangan perangkat lunak yang sistematik hingga selesai untuk setiap tahapnya sebelum berpindah ke tahapan berikutnya.



Gambar 7. Model *Waterfall*

1. *Requirement Analysis and Definition*

Mengumpulkan kebutuhan secara lengkap kemudian kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

2. *System and Software Design*

Dalam tahapan ini akan dibentuk suatu arsitektur sistem berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar sistem perangkat lunak dan hubungan-hubungannya.

3. *Implementation and Unit Testing*

Dalam tahapan implementasi dan pengujian unit, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. *Integration and System Testing*

Dalam tahapan integrasi dan pengujian sistem, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu sistem yang utuh untuk memastikan sistem sudah memenuhi persyaratan yang ada. Setelah itu sistem akan dikirim ke pengguna sistem.

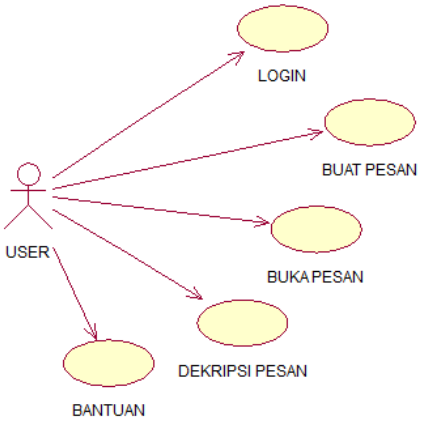
5. *Operation and Maintenance*

Dalam tahapan ini, sistem diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan sistem seperti penambahan fitur dan fungsi baru.

4. RANCANGAN SISTEM

Berikut ini merupakan perancangan sistem yang akan dibangun pada aplikasi pengamanan pesan text dalam email adalah sebagai berikut :

1. *Use Case Diagram*



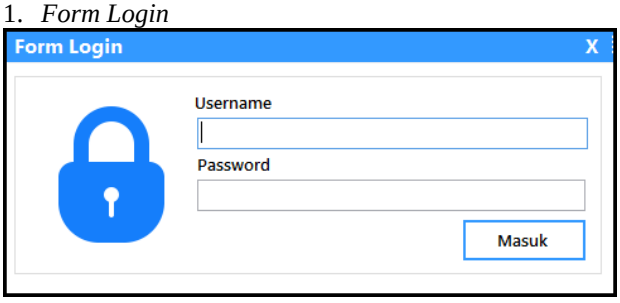
Gambar 8. *Use Case Diagram*

Gambar 8 merupakan *Use Case Diagram* yang menggambarkan segala aktifitas *user* yang dapat dilakukan pada aplikasi pengamanan pesan *text* dalam *email*.

5. IMPLEMENTASI

Hasil implementasi pada aplikasi pengamanan pesan *text* dalam *email* sebagai berikut :

1. *Form Login*

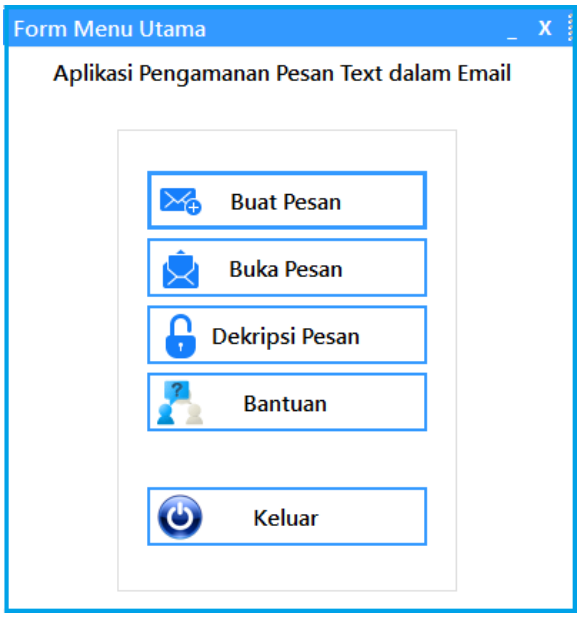


Gambar 9. *Form Login*

Gambar 9 merupakan tampilan *form login* untuk masuk ke dalam aplikasi, dimana pada *form* ini *user* akan menginputkan *username* dan *password* untuk bisa

menggunakan aplikasi pengamanan pesan *text* dalam *email*.

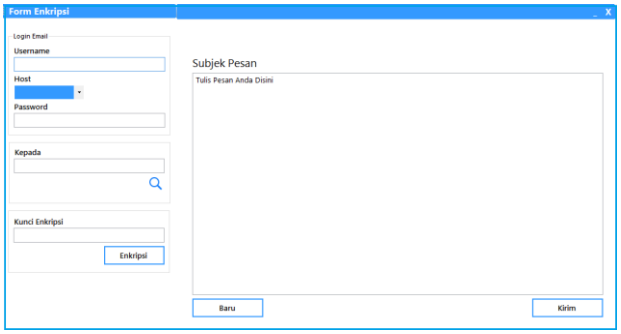
2. *Form* Menu Utama



Gambar 10. *Form* Menu Utama

Gambar 10 merupakan *form* menu utama, terdapat beberapa pilihan menu yang dapat dipilih oleh *user*, yaitu *user* dapat memilih menu buat pesan untuk membuat pesan, *user* dapat memilih menu buka pesan untuk membuka dan membaca pesan yang masuk, *user* dapat memilih menu dekripsi pesan untuk mendekripsi pesan yang terenkripsi, dan *user* dapat memilih menu bantuan untuk melihat tutorial penggunaan aplikasi, serta tombol keluar untuk keluar dari aplikasi.

3. *Form* Buat Pesan

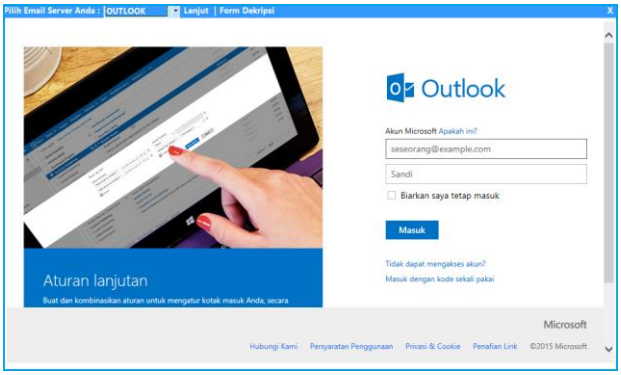


Gambar 11. Form Buat Pesan

Gambar 11 merupakan tampilan menu buat pesan. *Form* ini digunakan *user* untuk mengirim pesan *text email* yang dapat dienkripsi. Di *form* ini terdapat beberapa *input*-an yaitu, *username email*, *host server email*, *password email*, *email* tujuan, subjek pesan, isi pesan, *password* yang merupakan kunci untuk mengenkripsi pesan.

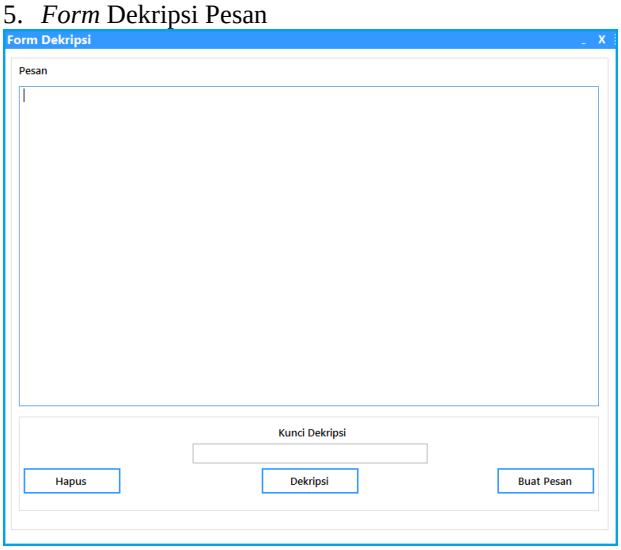
4. *Form* Buka Pesan

Pada gambar 12 merupakan tampilan menu buka pesan. *Form* ini digunakan untuk mengakses pesan masuk pada akun *email user*.



Gambar 12. Form Buka Pesan

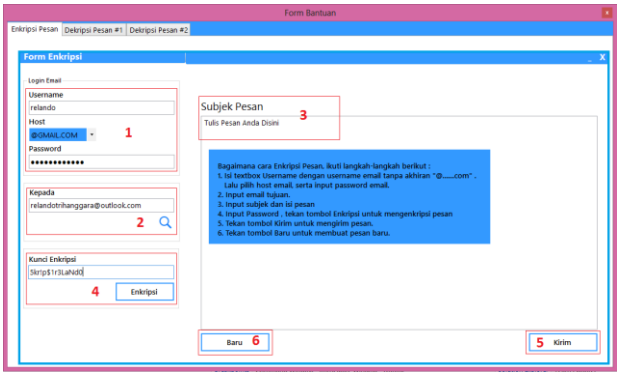
5. *Form* Dekripsi Pesan



Gambar 13. *Form* Dekripsi Pesan

Gambar 13 merupakan tampilan menu dekripsi pesan. *Form* ini merupakan tampilan menu untuk mendekripsi pesan *text* yang yang telah di salin (*copy*) dari *form* buka pesan. Pesan yang telah di salin (*copy*) lalu di tempel (*paste*) di *textbox* pesan enkripsi. *Textbox* kunci merupakan kunci yang disepakati saat mengenkripsi pesan.

6. *Form* Bantuan



Gambar 14. *Form* Bantuan

Form bantuan merupakan *form* yang digunakan *user* sebagai tutorial penggunaan aplikasi pengamanan pesan *text* dalam *email* dalam bentuk gambar-gambar.

6. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan, yaitu :

1. Pembuatan aplikasi pengamanan pesan *text* dalam *email* ini menggunakan *Microsoft Visual Studio* dengan bahasa pemrograman *Visual Basic.Net*.
2. *Use Case Diagram*, *Activity Diagram* dan *Sequence Diagram* sebagai alat bantu perancangan aplikasi pengamanan pesan *text* dalam *email*.
3. Aplikasi pengamanan pesan *text* dalam *email* ini dapat menjadi salah satu alat komunikasi yang bersifat rahasia dikarenakan pesan akan dienkripsi menggunakan algoritma *Advanced Encryption Standard* (AES) 128 Bit sebelum dikirimkan ke tujuan.

7. SARAN

Berdasarkan hasil penelitian yang telah dilakukan, disadari masih banyak kekurangan pada aplikasi pengamanan *text* dalam *email* ini. Diharapkan penelitian selanjutnya aplikasi ini dapat dikembangkan seperti saran-saran berikut :

1. Aplikasi pengamanan pesan *text* dalam *email* masih bersifat *standalone* dan masih berbasis *desktop*, diharapkan penelitian selanjutnya dapat dikembangkan menjadi aplikasi berbasis *web* ataupun aplikasi berbasis *smartphone*.
2. Aplikasi ini menggunakan algoritma *Advanced Encryption Standard* (AES) 128 bit dalam proses enkripsi dan dekripsinya, diharapkan penelitian kedepannya dapat menggunakan *Advanced Encryption Standard* (AES) 192 bit ataupun *Advanced Encryption Standard* (AES) 256 bit agar semakin sulit dipecahkan.
3. Diharapkan kedepannya aplikasi ini dapat dikembangkan dalam hal tampilan mukanya sehingga semakin menarik dan semakin mempermudah *user*.

4. DAFTAR PUSTAKA

Ariwisanto Sianturi, Fricles, 2013, Perancangan Aplikasi Pengamanan Data dengan Kriptografi Advanced Encryption Standard (AES), Jurnal Pelita Informatika Budi Darma. Vol : IV (1)

Ariyana, Yoki, 2011, Advanced Encryption Standard (AES). Bandung : PPPPTK IPA Bandung.

Ariyus, Dony, 2006, Computer Security, Yogyakarta: CV ANDI OFFSET.

Dhanta, Risky, 2009, Kamus Istilah Komputer Grafis & Internet, Surabaya : Indah.

Enterprise, Jubilee, 2014, Buku Pintar VB.NET, Jakarta : Elex Media Komputindo.

Henderi, 2007, Analysis and Design System with Unfied Modeling Language (UML), STMIK Raharja, Tangerang.

Kurniawan, Agus, 2008, Konsep dan Implementasi Cryptography Dengan .NET. Jakarta: PC Media.

Kurniawan, Yusuf, 2006, Kriptografi: Keamanan Internet dan Jaringan Komunikasi. Bandung: Informatika.

Munir, Rinaldi, 2006, Pengantar Kriptografi, Bandung: Informatika.

Munir, Rinaldi, 2005, Advanced Encryption Standard (AES). Institut Teknologi Bandung: Bandung.

Nugroho, Adi, 2010, Rekayasa Perangkat Lunak Berorientasi Objek dengan Metode USDP, Yogyakarta: CV ANDI OFFSET.

Prasetyo, Didik Dwi, 2005, Buku Pintar Internet, Jakarta : Elex Media Komputindo.

Pressman, Roger.S, 2005, Rekayasa Perangkat Lunak, Yogyakarta : CV ANDI OFFSET.

Rosyadi, Ahmad, 2012, Implementasi Algoritma Kriptografi AES untuk Enkripsi dan Dekripsi Email, Semarang : Universitas Diponegoro.

Sommerville, Ian. 2011. Software Engineering (Rekayasa Perangkat Lunak), Jakarta: Erlangga.

Texas Instruments, 2009, AES128 – A C Implementation for Encryption and Decryption.

Widodo, Wahyu , 2013, Aplikasi Pengamanan SMS dengan Metode Kriptografi Advanced Encryption Standard (AES) 128 Berbasis Android, Skripsi, Samarinda : STMIK Wicida.