ISSN : 2548-6985

Rancang Bangun Aplikasi Keamanan Data Menggunakan Metode AES Pada Smartphone

Amir Mahmud Hasibuan
STMIK Budi Darma, Jl. Sisingamangaraja No.338 Medan, Sumatera Utara, Indonesia
http : //www.stmik-budidarma.ac.id // Email : amirmahmudhasibuan@gmail.com

ABSTRACT
Cryptography is one technique used to improve the security aspects of information. Cryptography is the study of science and art to maintain a message or data information so that the data is safe. Cryptography supports the needs of two aspects of information security, namely secrecy (protection of the confidentiality of information data) and authenticity (protection against counterfeiting and changing unwanted information).
Along with the development of computer technology, the world of information technology requires a stronger and safer cryptographic algorithm. Currently the Advanced Encryption Standard (AES) is used as the latest standard cryptographic algorithm. For this reason, it is necessary to prepare an application that can secure a data and maintain its confidentiality so that it is not known by unauthorized parties. One alternative that can be used in making a data security application is by applying the Advanced Encryption Standard (AES) algorithm. hence the design of a data security application on a smartphone by designing the application of encryption and description of text data using the AES method. This method does a round of 10 rounds to get the results of encryption and decryption in the text.

Keywords: Cryptography, AES, Smartphone.

PENDAHULUAN

Keamanan data yang bersifat pribadi sangatlah penting pada teknologi smartphone sekarang ini, dimana sering terjadi pencurian data, menyalahgunakan data oleh pihak yang tidak berwenang pada smartphone. Seiring meningkatnya pencurian data, pembajakan data, serta penyalahgunaan data tersebut mengakibatkan kerugian bagi pemiliki data informasi tersebut. Agar data teks dapat dijaga kerahasiannya maka dirancang sebuah aplikasi kriptografi untuk menyandikan pesan teks menggunakan metode AES[5].

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika yang berkaitan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentifikasi entitas.

AES merupakan sistem penyandian blok yang bersifat non-feistel kerena AES menggunakan komponen yang selalu memiliki invers dengan panjang blok 128. Penyandian AES menggunakan proses yang berulang disebut dengan ronde. jumlah ronde yang digunakan oleh AES tergantung panjang kunci yang digunakan. Enkripsi pesan teks pada AES adalah transformasi terhadap state secara berulang dalam beberapa ronde. state menjadi keluaran ronde, k masukan untuk ronde untuk ronde k-k+1.

Masalah yang dibahas dalam skripsi ini yaitu:

1. Bagaimana Proses enkripsi dan dekripsi metode AES ?
2. Bagaimana menerapkan metodeAES dalam pengaman data ?
3. Bagaimana merancangperangkat lunak yang dapat diaplikasikan dengan kriptografi algoritma AES pada smartphone ?

Dalam penelitian ini yang menjadi batasan masalah adalah: Data yang digunakan dalam algoritma AES dalam bentuk teks dengan panjang maksimal 128 bit. Aplikasi yang dirancang dengan metode AES menggunakan model enkripsi simetris Block Cipher dan hanya berjalan pada smartphone yang memiliki sistem operasi android. Perancangan aplikasi yang dibangun menggunakan bahasa pemrograman java untuk smartphone samsung.

Adapun manfaat dari perancangan aplikasi ini yaitu: Pengguna dapat memanfaatkan media smartphone untuk penyandian data teks atau kode informasi ke dalam bentuk kriptografi. Dapat dijadikan alternatif dalam pengamanan data teks. Mencegah pencurian dan penyalahgunaan data teks pada smartphone agar tidak bisa dilihat oleh orang lain [1].

LANDASAN TEORI

ISSN : 2548-6985

2.1 Keamanan Data

Keamanan data telah menjadi bagian dari pengembangan teknologi informasi mengingat bahwa berjuta-juta bit informasi telah dipertukarkan dalam jaringan komputer terutama di internet. Masalah keamanan data dapat diklasifikasi ke dalam beberapa dimensi.

2.2 Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasian, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi.

2.3 Advanced Encryption Standard (AES)

Pada tahun 90-an, setelah beberapa tahun standar penyandian simetris DES dianggap tidak lagi aman, lembaga standar amerika serikat NIST (National Institute Of standars and Technology) membuat sayembara untuk menggantikan DES dengan sebuah sistem penyandian yang disebut dengan Advanced Encryption Standars pada tanggal 12 September 1997. NIST memberikan spesifikasi AES, yaitu harus memiliki panjang kunci 128 bit, 192 bit dan 256 bit.

Setelah beberapa seleksi, NIST memilih sistem penyandian Rijndael yang dikembangkan oleh Joan Daemen dan Vincent Rijment sebagai sistem penyandian AES pada tahun 2000. Pemilihan Rijndael berdasarkan pada kriteria:

1. Keamanan Sistem penyandian harus tahan terhadap serangan analisis sandi selain serangan secara brute force.
2. Biaya Sistem penyandian harus memiliki biaya komputasi dan memori yang efesien sehinggan dapat diimplementasikan secara perangkat keras maupun perangkat lunak.
3. Karakteristik algoritma dan implementasi Sistem penyandian harus bersifat terbuka, fleksibel, dan sederhana.

Pada tahun 2001 akhirnya NIST mempublikasikan AES sebagai standar pemrosesan dokumen pada dokumen FIPS-PUB 197.

2.3.1 Deskripsi AES

AES merupakan sistem penyandian blok yang bersifat non-Feistel karena AES menggunakan komponen yang selalu memiliki invers dengan panjang blok 128 bit. Kunci AES dapat memiliki panjang kunci bit 128, 192 dan 256 bit. Penyandian AES menggunakan proses yang berulang yang disebut dengan ronde. Jumlah ronde yang digunakan oleh AES tergantung dengan panjang kunci yang digunakan. Setiap ronde membutuhkan kunci ronde dan masukan dari ronde berikutnya. Kunci ronde dibangkitkan berdasarkan kunci yang diberikan (Rifki Sadikin, 2012).

Tabel 1 Hubungan antara jumlah ronde dan panjang kunci AES

Panjang Kunci AES (bit)	Jumlah *Ronde* (Nr)
128	10
192	12
256	14

Sumber: Rifki Sadikin, 2012, 152

2.3.2 Struktur Enkripsi AES

Proses di dalam AES merupakan transformasi terhadap state. Sebuah teks asli dalam blo (128 bit) terlebih dahulu diorganisir sebagai state. Enkripsi AES adalah transformasi terhadap state secara berulang dalam beberapa ronde. State yang menjadi keluaran ronde k menjadi masukan untuk ronde ke-k + 1 [4].

2.3.3 Transformasi-transformasi AES

Algoritma enkripsi AES menggunakan 4 jenis transformasi: substitusi yang disebut dengan *SubBytes*, permutasi yang dengan *ShiftRows*, pencampuran yang disebut *MixColumns*, dan penambahan kunci yang disebut dengan *AddRoundKey*. Pada bagian ini ke-4 transformasi ini akan dibahas.

2.3.4 *SubBytes*

AES menggunakan substitusi nonliniar pada ukuran *byte* yang dengan *SubBytes*. Setiap elemen pada *state* dari elemen $^{S}0,0$ sampai dengan $^{S}3,3$ dikenakan transformasi *SubBytes*. Ada 2 cara untuk mengkomputasi substitusi dengan *SubBytes*, yaitu dengan menggunakan substitusi atau dengan menggunakan komputasi pada GF(2^8).

2.3.5 *ShiftRows*

Selain dengan menggunakan substitusi untuk mengganti nilai pada elemen *state*, AES menggunakan permutasi pada *state*. Transformasi permutasi pada *state* disebut dengan transformasi *ShiftRows*. *ShiftRows* dilakukan dengan menjalankan operasi *circular shift left* sebanyak *i* pada baris ke-*i* pada *state*.

Transformasi *shiftRows* merupakan jenis transformasi permutasi, yaitu pengubahan posisi elemen pada *state* tanpa mengubah nilainya. Transformasi *ShiftRows* terlihat sederhana jika dilihat melalui representasi *state*. Namun, karena *state* adalah representasi blok dengan orientasi per kolom menjadikan transformasi *ShiftRows* menjadi rumit jika dilihat dari sudut pandang blok.

2.3.6 *MixColumns*

Tujuan transformasi *MixColumns* adalah mencampur nilai kolom-kolom pada *state* pada satu elemen pada *state* keluaran. Untuk melakukan pencampuran itu, transformasi *MixColumns2* menggunakan operasi perkalian matriks dengan operasi perkalian dan penjumlahan menggunakan operator pada GF(2^8) dengan *irreducible polynomial* ($x^8 + x^4 + x^3 + x + 1$).

2.3.7 *AddRoundKey*

Transformasi keempat yang digunakan pada penyandian AES adalah transformasi *AddRoundKey*. Transformasi *AddRoundKey* mencampur sebuah *state* masukan dengan kunci ronde dengan operasi eksklusif OR ($\oplus$). Setiap elemen pada *state* masukan yang merupakan sebuah *byte* dikenakan operasi ekslusif OR dengan *byte* pada posisi yang sama di kunci ronde (kunci ronde direpresentasikan sebagai *state*).

Transformasi *AddRoundKey* merupakan transformasi yang bersifat *self invers*, yaitu transformasi *invers* sama dengan transformasi aslinya asalkan menggunakan kunci ronde yang sama [4].

2.4 Struktur Dekripsi AES

Algoritma dekripsi merupakan kebalikan algoritma enkripsi AES. algoritma dekripsi AES menggunakan transformasi invers semua transformasi dasar AES memiliki transformasi *invers*, yaitu: *InvSubBytes*, *InvShiftRows* dan *InvMixColumns*. *AddRoundKey* merupakan transformasi yang bersifat *self-invers* dengan syarat menggunakan kunci yang sama [4].

2.4.1 *InvShiftRows*

I*nvShiftRows* adalah transformasi *byte* kebalikan dari transformasi *ShiftRows*. pada transformasi *InvShiftRows* dilakukan pergeseran bit ke kanan sedangkan pada *ShiftRows* dilakukan pergesean bit ke kiri.

2.4.2 *InvSubBytes*

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. pada *InvSubBytes* tiap elemen pada *state* dipetakan dengan menggunakan tabel *Invers S-Box*.

2.4.3 *InvMixColumns*

Setiap kolom dalam state dikalikan dengan matriks perkalian dalam AES. perkalian dalam matriks dapat dilihat di bawah ini :

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Hasil dari perkalian pada matrik adalah :

$$s'_{0,c} = (\{0E\} \bullet s_{0,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0D\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0B\} \bullet s_{0,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})$$

PEMBAHASAN

3.1 Analisa

Masalah keamanan merupakan salah satu aspek terpenting dari sebuah informasi. Android merupakan sebuah sistem informasi untuk perangkat mobile berbasis linux yang mencakup sistem informasi, android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi sendiri. Advanced Encryption Standard (AES) adalah algoritma enkripsi blok cipher dengan variabel panjang blok dan panjang kunci. Panjang blok dan panjang kunci yang digunakan adalah 128 bit. Dalam pengamanan data teks menggunakan algoritma kriptografi yang berguna untuk menyandikan pesan agar tidak dapat dibaca. Salah satu algoritma kriptografi yang digunakan adalah *Advanced Encryption Standart* (AES). sistem yang dirancang menggunakan algoritma AES dalam pengamanan data teks.

3.1.1 Enkripsi dan Dekripsi Algoritma AES

Langkah – langkah Enkripsi algoritma AES adalah sebagai berikut:

1. Ekspansi Kunci, langkah-langkahnya adalah:
   a. Kunci AES 128 bit diorganisir menjadi 4 *word* dan disalin ke *word* keluaran (*w*) pada 4 elemen pertama (*w*[0], *w*[1], *w*[2], *w*[3]).
   b. Untuk elemen keluaran selanjutnya *w*[*i*], dengan *i* = {4,... , 43} dihitung sebagai berikut:
      1) Salin *w*[*i* – 1] pada *word t*
      2) Jika *i* mod 4 = 0 (*i* habis dibagi 4) maka lakukan *w*[*i*] = *f* (*t*,*i*)$\oplus$ *w*[*i* -

4] dengan fungsi $f(t, i) = SubWord(RotWord(t)) \oplus RC[i/4]$.

Fungsi *RotWord* adalah fungsi geser sirkular ke kiri 1 *byte* pada satu *word*. Bila masukan *RotWord* adalah {b_0, b_1, b_2, b_3} maka keluarannya adalah {b_1, b_2, b_3, b_0}. Sedangkan fungsi *SubWord* menggunakan transformasi *SubBytes* untuk mensubstitusi tiap elemen pada *word* nilai *byte* baru. Sedangkan RC[*i*] adalah konstan yang didefinisikan oleh tabel 1.

Konstan RCon dapat dicari dengan cara lain dengan tidak menggunakan tabel 3.1 tetapi menggunakan kalkulasi pada GF(2^8) dengan modulus polinomial $x^8 + x^4 + x^3 + x + 1$.

I	1	2	3	4	5	6	7	8	9	10
RC[*I*]	01	02	04	08	10	20	40	80	1B	3C

RCon[*i*] = x^i

3) Jika *i* mod 4 ≠ 0, lakukan $w[i] = t \oplus w[i-4]$

2. *AddRoundKey*: melakukan XOR antara *state* awal (plainteks) dengan *cipher key*. Tahap ini disebut juga *initial round*.
3. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. *SubBytes*: substitusi *byte* dengan menggunakan tabel substitusi (S-box).
   b. ShiftRows: pergeseran baris-baris *array state* secara *wrapping*.
   c. *MixColumns*: mengacak data di masing-masing kolom *array state*.
   d. *AddRoundKey*: melakukan XOR antara *state* sekarang round key.
4. *Final round*: proses untuk putaran terakhir:
   a. *SubBytes*
   b. *ShiftRows*
   c. *AddRoundKey*

Langkah – langkah Dekripsi adalah sebagai berikut:

1. Mengekspansi kunci dan putaran kunci sama dengan yang digunakan dalam proses Enkripsi
2. Meng-XOR kan *ciphertext* dengan kunci putaran 10
3. Melakukan transformasi *InvShiftRows*
4. Melakukan transformasi *InvSubBytes*
5. Melakukan transformasi invers *AddRoundKey, InvMixColums, InvShiftRows, InvSubBytes* sebanyak 10 putaran
6. Meng-XOR kan hasil 10 putaran dengan kunci

Maka dapat dihasilkan *plaintext*.

3.1.2 Proses Pengamanan Data

Misalkan:

Plaintext: AMIRMAHMUDHASIBU
Cipherkey: ANSTMIKBUDIDARMA

Tahapan pertama yang dilakukan adalah:

1. Ubah *plaintext* dan *cipherkey* ke dalam bentuk heksadesimaldengan melihat tabel ASCII (*American Standard Code for Information Interchange*) maka hasilnya adalah:
   Plaintext: 41 4d 49 52 4d 41 48 4d 55 44 48 41 53 49 42 55
   Cipherkey: 41 4e 53 54 4d 49 4b 42 55 44 49 44 41 52 4d 41
   Dengan 16 *byte* blok data dan kunci yang berukuran 128 bit. maka blok data 128 bit untuk ukuran *state* adalah 4x4
2. Transformasikan *plaintext* yang sudah dirubah ke *hexadecimal* dalam bentuk tabel 4x4 seperti di bawah ini:

Tabel 2 *plaintext* ke *hexadecimal*

Plaintext			
W_0	W_1	W_2	W_3
41	4d	55	53
4d	41	44	49
49	48	48	42
52	4d	41	55

Untuk mencari putaran kunci digunakan rumus sebagai berikut:

$W_4 = W_0 \oplus SubWord(RotWord(W_3)) \oplus RCon[1]$

Karena W_3 = 53 49 42 55, maka
RotWord (geser posisi 1 bit)
= 49 42 55 53
SubWord = 49 42 55 53 di rubah dengan melihat tabel *S-Box*, maka
= 3b 2c fc ed

Untuk mendapatkan *RoundKey* kolom II sampai kolom IV tidak perlu di XOR kan dengan Rcon, Rcon hanya digunakan untuk mendapatkan *byte* kolom I tiap putaran.

Hasil nya adalah:

41 4d 49 52 $\oplus$ 3b 2c fc ed $\oplus$ 01 = 7b 60 b4 be

4d 41 48 4d $\oplus$ 7b 60 b4 be = 34 23 fe f1

55 44 48 41 $\oplus$ 34 23 fe f1 = 65 63 b2 b4

53 49 42 55 $\oplus$ 65 63 b2 b4 = 3e 2a f8 e1

Maka dapat lah putaran kunci algoritma AES

Tabel 3 *RoundKey*

Round 1				Round 2				Round 3				Round 4			
7b	34	65	3e	5b	2e	f3	Df	d6	14	c5	Bc	3c	a9	50	ad
60	23	63	2a	b1	07	63	8c	0a	1a	a7	a4	05	3d	93	fa
b4	fe	b2	f8	58	47	3d	Eb	06	43	30	1c	5f	0b	11	aa
Be	f1	b4	e1	11	52	1f	Fd	74	b8	ce	39	f9	4d	af	e3

Round 5				Round 6				Round 7				Round 8			
49	5a	4c	73	3b	c6	73	1b	90	54	86	76	c1	3f	0a	4d
54	43	49	75	83	0b	09	64	65	75	dc	0b	49	a7	90	74
Ac	48	41	d9	2f	92	9b	1b	73	7f	30	40	35	ea	fe	28
49	55	4e	83	29	58	27	73	44	bc	ec	45	56	5d	e7	34

Round 9				Round 10			
2f	Ba	fa	ef	31	46	1f	a4
29	73	5e	7d	50	ef	aa	b1
43	04	41	0c	b3	c0	9a	05
85	4b	81	c6	7c	78	e9	44

3. Meng-XOR kan *plaintext* dengan *cipherkey*
   41 = 01000001 (*plaintext*)
   41 = 01000001 (*cipherkey*)

Maka hasilnya = 00000000 dalam bilangan *hexadecimal* = 00

Hasil XOR

00	00	00	12
03	08	00	1b
1a	03	01	0f
06	0f	05	14

4. Melakukan transformasi *SubBytes*
   Hasil XOR *S-Box*
   00 = 63

SubBytes			
63	63	63	c9
7b	30	63	Af
a2	7b	7c	76
6f	76	6b	Fa

5. Melakukan transformasi *ShiftRows*
   a. Baris pertama tetap atau tidak terjadi pergeseran
   b. Baris ke dua bergeser 1 *byte* ke kanan
   c. Baris ke tiga bergeser 2 *byte* ke kanan
   d. Baris ke empat bergeser 3 *byte* ke kanan

ShifRows			
63	63	63	c9
30	63	af	7b
7c	76	a2	7b
Fa	6f	76	6b

6. Transformasi *MixColumns*
   Hasil dari *ShiftRows* di kalikan dengan matriks yang sudah ditentukan. Hasil perkalian tetap berada pada medan galois ($GF(2^8)$).
   63.(02) $\oplus$ 30.(03) $\oplus$ 7c.(01) $\oplus$ fa.(01)
   01100011.(00000010) $\oplus$
   0110000.(00000011) $\oplus$
   1111100.(00000001) $\oplus$
   1111010.(00000001)
   $x^6 + x^5 + x + 1$ (x) $\oplus$ $x^5 + x^4$ (x + 1) $\oplus$ $x^6 + x^5 + x^4 + x^3 + x^2$ (1) $\oplus$ $x^7 + x^6 + x^5 + x^4 + x^3 + x(1)$
   $= x^4 + x = 12$

MixColumns			
12	b3	01	1c
7d	89	78	d1
Bd	Cc	Dc	67
b8	22	f9	30

7. Melakukan transformasi *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns* sebanyak 10 putaran

ALGORITMA DAN IMPLEMENTASI

4.1 Algoritma

1. Enkripsi AES

Input:

P = *plaintext*
K = Kunci

P, K {*Teks* asli 16 *byte,* Kunci AES (128 bit)}
Output:
CT = *Cipher Text*
CT {*Teks* sandi 16 *byte*}
Proses:
Nr = Jumlah Putaran
w = *word*
(*Nr*, *w*) ← EkspansiKunci (K) {Nr : Jumlah Ronde, w : larik *bytes* kunci ronde }
CT = P
AddRoundKey (CT, *w* [0..3])
For *i* = 1 → *Nr* do
SubBytes (CT)
ShiftRows (CT)
if *i* ≠ Nr Then
MixColumns (CT)
end if
AddRoundKey (CT, *w* [(*i* * 4).. (*i* * 4) + 3])
end for

2. Dekripsi AES
Input:
CT = *Cipher Text*
K = Kunci
CT, K {Teks sandi 16 *bytes*, kunci AES (128 bit)}
Output:
P = *Plaintext*
P {Teks asli 16 *bytes*}
Proses:
Nr = Jumlah Putaran
w = *word*
(Nr, *w*) ← EkspansiKunci (K) {Nr: Jumlah Ronde, w : larik *bytes* kunci ronde}
P = CT
AddRoundKey (P, *w*[Nr * 4..Nr * 4 – 3])
for *i* = 1 → Nr do
InvSubBytes (P)
InvShiftRows (P)
AddRoundKey (P, *w* [(Nr – *i*) * 4.. ((Nr – *i*) * 4) + 3])
if *i* ≠ Nr then
InvMixColumns (P)
end if
end for

4.2 Implementasi

Gambar 1. Tampilan Utama

Gambar 2 Tampilan AES

Gambar 3 Tampilan Info

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah merancang dan mengaplikasikan rancangan perangkat lunak keamanan data menggunakan metode AES maka dapat diambil kesimpulan bahwa:

1. perangkat lunak ini dapat mengkodekan data teks asli menjadi data teks campuran dan mengembalikan data teks tersebut seperti semula.
2. Sistem yang penulis rancang sudah dapat dijalankan pada *Smartphone* android.
3. Sistem yang penulis rancang masih bersifat data teks dan belum sampai pada pengiriman data teks.

5.2 Saran

Penulis menyadari bahwa aplikasi ini masih banyak kekurangan yang diharapkan diibutuhkan pengembangan lebih lanjut pada aplikasi. Adapun saran – saran dari penulis antara lain:

1. Untuk penelitian selanjutnya diharapkan data yang digunakan dalam algoritma AES dalam bentuk dokumen.
2. Aplikasi yang dirancang dapat menggunakan *software* yang lain seperti *Visual Studio 2008* dan *Macromedia Flash*.
3. Pengembang dapat menambahkan tampilan halaman untuk *login*.

ISSN : 2548-6985

DAFTAR PUSTAKA

1. Gata, Windu. (2013).Sukses Membangun Aplikasi Penjualan Dengan Java.Jakarta: Penerbit PT Elex Media Komputindo.
2. Jogiyanto. (2005). Analisis dan Desain Sistem Informasi. Yogyakarta: Penerbit Andi.
3. Nugroho, Adi. (2009). Rekayasa Perangkat Lunak Menggunakan UML dan JAVA. Yogyakarta: Penerbit Andi.
4. Sadikin, Rifki. (2012). Kriptografi Untuk Keamanan Jaringanan. Yogyakarta: Penerbit Andi.
5. Safaat H, Nazruddin. (2012) Pemrograman Aplikasi Mobile Smartphone Dan Tablet PC Berbasis Android. Bandung: Penerbit Informatika.
6. Simarmata, Janner. (2006). Pengenalan Teknologi Komputer dan Informasi. Yogyakarta: Penerbit Andi.
7. https://prezi.com/cqq4id3jtfqu/pengertian-keamanan-komputer-dan-konsep-keamanan-jaringan.html
8. http://www.pintarkomputer.org/2015/03/pengertian-android-menurut-para-ahli.html

MEDIA INFORMASI ANALISA DAN SISTEM
MEANS
UNIKA ST. THOMAS
PROGRAM STUDI : SISTEM INFORMASI