DIGITAL SIGNATURE PADA KARTU TANDA PENDUDUK

Oleh :

Junaidi Surya, M.Kom

STMIK Nurdin Hamzah Jambi

ABSTRACT

Digital Signature in this time represent a inportant component in a digital document, requirement for this signature progressively strengthened because progressively expand technological of information, conducive of document spreading easy to and free. Digital signature follow the nature of applicable conventional signature in digital document. Such Signature perhaps non habit signature which in the scan into digital format, but signature applied the concepts Cryptography. Digital Signature gift can exploit the system of cryptography and message digest 5 (MD5) by using public key symmentrically in yielding same output 128 beet without your Exellency how many string include to get the value hashing.

Keyword : Digital Signature, Cryptography, MD 5 and Hashing

I. PENDAHULUAN

1.1 Latar Belakang

Kemajuan ilmu pengetahuan dan teknologi menjadi salah satu faktor penentu bagi suatu peradaban yang modern. Keberhasilan yang dicapai dalam bidang ilmu pengetahuan dan perkembangan teknologi tentu saja akan membawa suatu negara pada kesejahteraaan dan

kemakmuran rakyatnya. Namun sejalan dengan kemajuan yang telah dicapai secara bersamaan dalam bidang ekonomi, ilmu pengetahuan dan teknologi, perkembangan tindak pidanapun ikut berkembang semakin tinggi dengan mempaatkan peradaban ilmu pengetahuan dan teknologi.

Kejahatan mengenai pemalsuan atau kejahatan pemalsuan adalah kejahatan yang mana di dalamnya mengandung sistem ketidakbenaran atau palsu sesuatu (obyek), yang sesuatunya itu tampak dari luar seolah-olah benar adanya, padahal sesungguhnya bertentangan dengan yang sebenarnya. Kartu Tanda Penduduk (KTP) adalah kartu identitas resmi dari seseorang di Indonesia yang diperoleh setelah seseorang berusia di atas 17 tahun atau yang sudah menikah. KTP berlaku selama 5 tahun dan tanggal berakhirnya disesuaikan dengan tanggal dan bulan kelahiran yang bersangkutan. Dimana pada KTP berisi informasi mengenai sang pemilik kartu, termasuk: nama lengkap, Nomor Induk Kependudukan (NIK), alamat, tempat dan tanggal lahir, agama, golongan darah, kewarganegaraan, foto dan tanda tangan. KTP ini digunakan pada berbagai bidang sebagai bukti identitas resmi yang diakui. Pada dasarnya setiap orang hanya memiliki satu KTP dan bersifat unik. Tetapi ada pihak yang dengan sengaja memalsukan KTP ini untuk maksud-maksud tertentu. Jika KTP palsu tersebut digunakan untuk suatu tindak kejahatan dengan mengatas namakan orang lain, maka tentu saja perbuatan tersebut dapat merugikan orang lain.

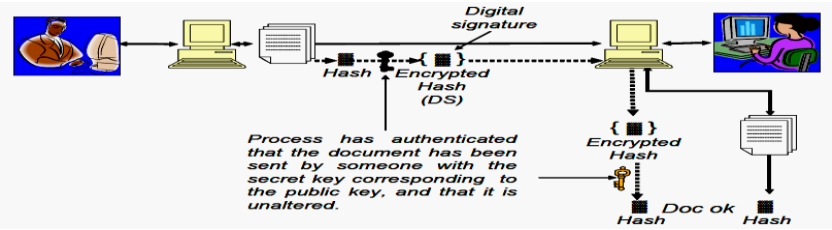
Oleh sebab itu maka penulis bermaksud untuk memperkecil ruang gerak pemalsuan KTP dengan menerapkan metoda digital signature pada setiap KTP resmi yang dikeluarkan oleh Pemerintah. Dalam hal ini yang menjadi digital signaturenya adalah Nomor Induk Kependudukan (NIK) dengan melakukan proses hashing satu arah. Algoritma yang digunakan untuk melakukan proses hashing adalah MD5. Dimana hasil dari MD5 berupa

message digest akan dienkripsi dengan menggunakan algoritma Advanced Encryption Standard (AES) dengan Cipher Mode Counter.

II. LANDASAN TEORI

2.1 Digital Signature

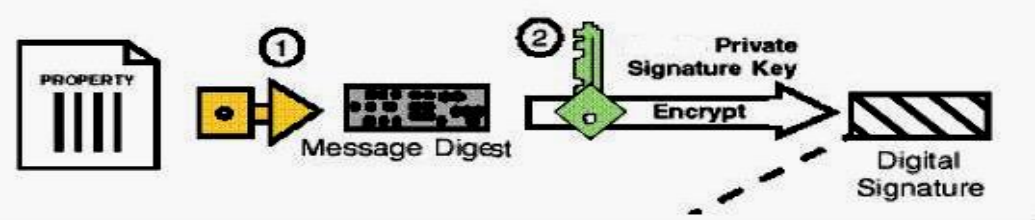
Digital signature atau tanda tangan digital adalah kode digital yang dapat ditempelkan pada pesan dikirim secara elektronis. Digital signature inilah yang menjadi identifikasi dari si pengirim pesan. Seperti halnya tanda tangan tertulis, tujuan digital signature adalah untuk menjamin bahwa yang mengirimkan pesan itu memang benar-benar orang yang seharusnya. Tanda-tangan digital berbeda dari tanda-tangan tinta tradisionil di mana banyak pengesahan percaya oleh berkwalitas forensik tanda-tangan dirinya**.** Berikut gambaran Digital Signature seperti berikut;



Gambar 1: Digital Signature

Untuk menandai dokumen, dokumen tersebut akan dihash dengan menggunakan suatu software dan disebut sebagai "Message digest". Kemudian Message Digest tersebut akan dienkripsi dengan kunci publik dan menghasilkan digital signature.

Dan digital signature ini akan ditambahkan pada KTP sebagai pengganti tanda tangan, sehingga semua data yang telah dihash telah ditandatangani. gambar berikut ini menjelaskan proses pembuatan digital signature.



Gambar 2: Digital Signature pada dokumen

2.2 Kriptografi

Kriptografi berasal dari bahasa yunani yang terdiri dari kata C*ryptos* yang berarti tersembunyi dan *grafo* yang berarti tulis. Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita. Selain pengertian tersebut kriptogragrafi juga merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data.

2.3 Enkripsi

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi sebuah data disandikan (encrypted) dengan menggunakan sebuah kunci (key). Untuk membuka (decrypt) data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi (untuk kasus private key cryptography) atau dengan kunci yang berbeda (untuk kasus public key cryptography). Berdasarkan cara memproses teks (plaintext), cipher dapat dikategorikan menjadi dua jenis:

- Block Cipher

Pada Block Cipher, sesuai namanya, data Plain Teks diolah per blok data. Block cipher bekerja dengan memproses data secara blok, dimana beberapa karakter / data digabungkan menjadi satu blok, setiap proses satu blok menghasilkan keluaran satu blok juga.

- Stream Cipher

Pada Stream Cipher, data Plain Teks diolah per satuan data terkecil, misalnya perbit atau per karakter. Stream cipher bekerja memproses masukan (karakter atau data) secara terus menerus dan menghasilkan data pada saat yang bersamaan.

2.4 Fungsi Hash Satu Arah

Seperti telah dijelaskan pada penjelasan mengenai tanda tangan digital, skema tanda tangan digital menggunakan fungsi *hash* satu arah. Fungsi *hash* adalah fungsi yang menerima masukan string yang panjangnya sembarang dan mengkonversinya menjadi *string* keluaran yang panjangnya tetap. Umumnya berukuran jauh lebih kecil daripada ukuran *string* semula. Fungsi *hash* menerima masukan *string* apa saja.

2.5 Message Digest-5 (MD5)

Message Digest-5 (MD5) merupakan fungsi hash satu arah yang dapat digunakan untuk mengetahui bahwa pesan yang dikirim tidak mengalami perubahan. Algoritma MD5 secara garis besar adalah mengambil pesan yang mempunyai panjang variabel diubah menjadi sidik jari atau intisari pesan yang mempunyai panjang tetap yaitu 128 bit. Sidik jari ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari sidik jari MD5.

III. ANALISIS PERANCANGAN

3.1 Analisis Perancangan Basis Data

Karena penelitian ini menggunakan menggunakan data penduduk yang bersumber pada kelurahan, maka struktur basis data yang digunakan adalah struktur basis data relasional,

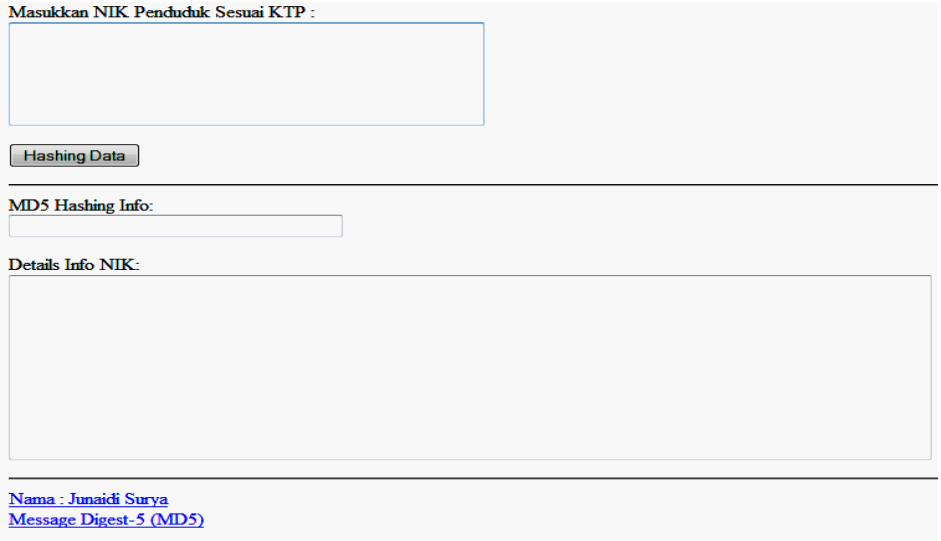
tetapi stuktur tabel tidak digunakan seceara keseluruhan. Dalam penelitian ini gambaran tabel yang digunakan sebagai berikut ;

Tabel 1 : Struktur Tabel Biodata

No Field	*Name Field*	*Type*	*Size*	*Dekarya ilmiah*
1	nik	varchar	15	No Induk Keluarga
2	nama	varchar	50	Nama Penduduk
3	tptlahir	varchar	50	Tempat Lahir penduduk
4	tgllahir	date	8	Tanggal Lahir Penduduk
5	Jenis kelamin	varchar	12	Jenis Kelamin
6	alamat	varchar	50	Alamat
7	RT/RW	varchar	50	RT dan RW Penduduk
8	Desa/kelurahan	varchar	35	Desa / Kelurahan
9	kecamatan	date	30	Kecamatan
10	Kab/kota	varchar	30	Kabupaten/Kota Madya
11	agama	varchar	20	Agama
12	pekerjaan	varchar	35	Pekerjaan Penduduk
13	berlaku_ktp		10	Masa Berlakunya KTP
14	md5hash	varchar	32	Info Hash MD5

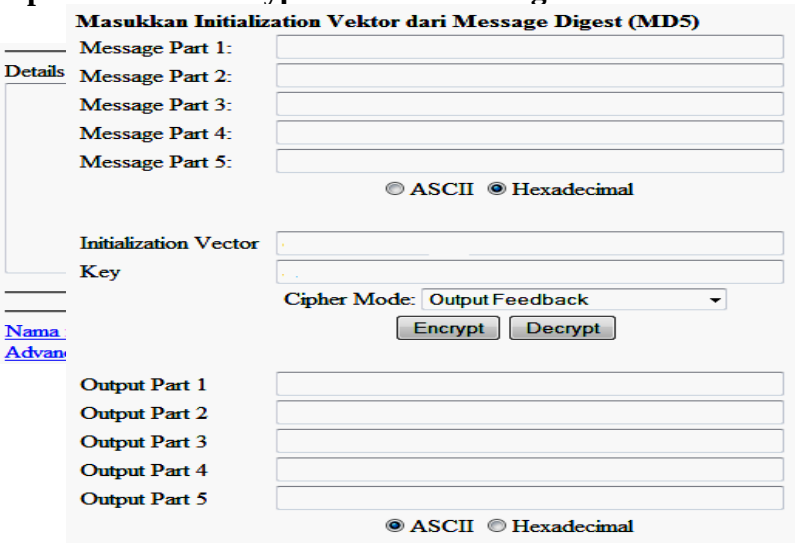
3.2 Analisis Perancangan Aplikasi

A. Aplikasi MD5



Gambar 3: Aplikasi MD5

B. Aplikasi AES Encryption Data Hashing MD5



Gambar 4: Aplikasi Encryption Data Hashing MD5

IV. PEMBAHASAN

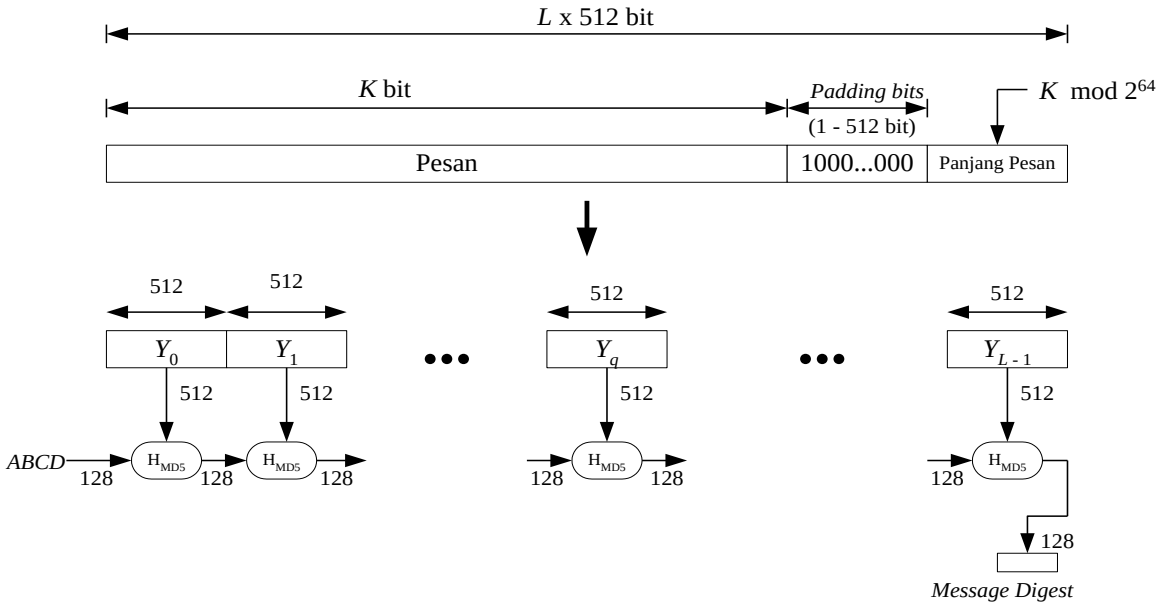
4.1 Cara kerja MD5

Setiap pesan yang akan dienkripsi, terlebih dahulu dicari berapa banyak bit yang terdapat pada pesan. Kita anggap sebanyak b bit. Di sini b adalah bit non negatif integer, b bisa saja nol dan tidak harus selalu kelipatan delapan. Pesan dengan panjang b bit dapat digambarkan seperti berikut :

m_0 m_1 …..m_(b-1)

Terdapat 4 langkah yang dibutuhkan untuk untuk menghitung intisari pesan.

Adapun langkah–langkah tersebut dijelaskan pada subbab-subbab berikut



Gambar 5 : Gambaran umum kerja MD5

Langkah-langkah pembuatan *message digest* secara garis besar:

1. Penambahan bit-bit pengganjal (*padding bits*).

2. Penambahan nilai panjang pesan semula.

3. Inisialisasi penyangga (*buffer*) *MD*.

4. Pengolahan pesan dalam blok berukuran 512 bit.

1. Penambahan Bit-bit Pengganjal

→ Pesan ditambah dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 modulo 512.

→ Jika panjang pesan 448 bit, maka pesan tersebut ditambah dengan 512 bit menjadi 960 bit. Jadi, panjang bit-bit pengganjal adalah antara 1 sampai 512.

→ Bit-bit pengganjal terdiri dari sebuah bit 1 diikuti dengan sisanya bit 0.

2. Penambahan Nilai Panjang Pesan

→ Pesan yang telah diberi bit-bit pengganjal selanjutnya ditambah lagi dengan 64 bit yang menyatakan panjang pesan semula.

→ Jika panjang pesan > 2^{64} maka yang diambil adalah panjangnya dalam modulo 2^{64}. Dengan kata lain, jika panjang pesan semula adalah *K* bit, maka 64 bit yang ditambahkan menyatakan *K* modulo 2^{64}.

→ Setelah ditambah dengan 64 bit, panjang pesan sekarang menjadi kelipatan 512 bit.

3. Inisialisai Penyangga MD

→ *MD5* membutuhkan 4 buah penyangga (*buffer*) yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 4 × 32 = 128 bit. Keempat penyangga ini menampung hasil antara dan hasil akhir.

→ Keempat penyangga ini diberi nama *A*, *B*, *C*, dan D. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut:

A = 01234567

B = 89ABCDEF

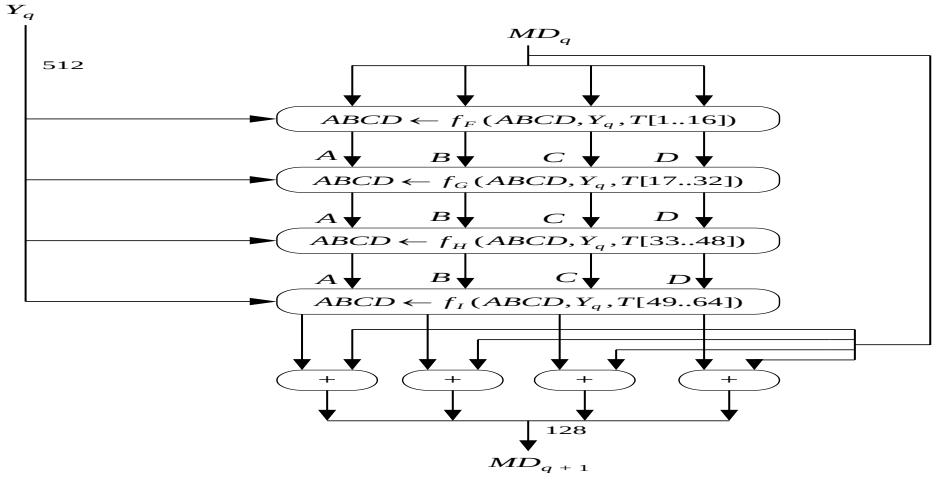
C = FEDCBA98

D = 76543210

4. Pengolahan Pesan dalam Blok Berukuran 512 bit

→ Pesan dibagi menjadi L buah blok yang masing-masing panjangnya 512 bit (Y_0 sampai Y_{L-1}).

→ Setiap blok 512-bit diproses bersama dengan penyangga *MD* menjadi keluaran 128-bit, dan ini disebut proses H_{MD5}. Gambaran proses H_{MD5} diperlihatkan pada Gambar 11.



Gambar 6 : Gambaran proses H_{MD5}

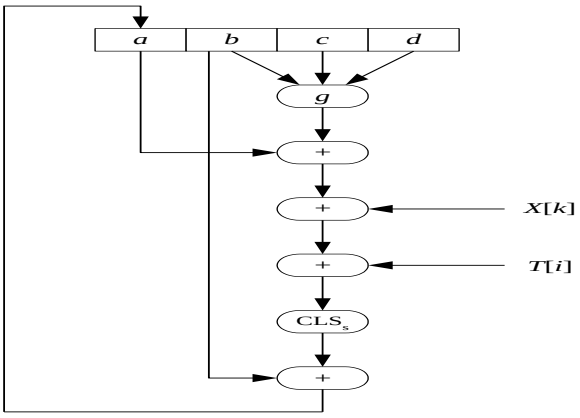
→ Pada Gambar .6, Y_q menyatakan blok 512-bit ke-q dari pesan yang telah ditambah bit-bit pengganjal dan tambahan 64 bit nilai panjang pesan semula.

→ MD_q adalah nilai *message digest* 128-bit dari proses H_{MD5} ke-q. Pada awal proses, MD_q berisi nilai inisialisasi penyangga *MD*.

→ Proses H_{MD5} terdiri dari 4 buah putaran, dan masing-masing putaran melakukan operasi dasar *MD5* sebanyak 16 kali dan setiap operasi dasar memakai sebuah elemen *T*. Jadi setiap putaran memakai 16 elemen Tabel *T*.

→ Fungsi-fungsi f_F, f_G, f_H, dan f_I masing-masing berisi 16 kali operasi dasar terhadap masukan, setiap operasi dasar menggunakan elemen Tabel *T*.

→ Operasi dasar *MD5* diperlihatkan pada Gambar 7.



Gambar 7 : Operasi Dasar MD5

Operasi dasar *MD5* yang diperlihatkan pada Gambar 7 dapat ditulis dengan sebuah persamaan sebagai berikut:

$$a \leftarrow b + \mathrm{CLS}_s(a + g(b, c, d) + X[k] + T[i])$$

yang dalam hal ini,

a, b, c, d = empat buah peubah penyangga 32-bit (berisi nilai penyangga A, B, C, D)

g = salah satu fungsi F, G, H, I

CLS_s = *circular left shift* sebanyak s bit

$X[k]$ = kelompok 32-bit ke-k dari blok 512 bit *message* ke-q. Nilai $k = 0$ sampai 15.

$T[i]$ = elemen Tabel T ke-i (32 bit)

$+$ = operasi penjumlahan modulo 2^{32}

Tabel 2 : Fungsi-fungsi dasar MD5

Nama	Notasi	$g(b, c, d)$
f_F	$F(b, c, d)$	$(b \wedge c) \vee (\sim b \wedge d)$
f_G	$G(b, c, d)$	$(b \wedge d) \vee (c \wedge \sim d)$
f_H	$H(b, c, d)$	$b \oplus c \oplus d$
f_I	$I(b, c, d)$	$c \oplus (b \wedge \sim d)$

Catatan: operator logika AND, OR, NOT, XOR masing-masing dilambangkan dengan $\wedge$, $\vee$, $\sim$, $\oplus$

Tabel 3 : Nilai *T*[i]

T[1] = D76AA478	T[17] = F61E2562	T[33] = FFFA3942	T[49] = F4292244
T[2] = E8C7B756	T[18] = C040B340	T[34] = 8771F681	T[50] = 432AFF97
T[3] = 242070DB	T[19] = 265E5A51	T[35] = 69D96122	T[51] = AB9423A7
T[4] = C1BDCEEE	T[20] = E9B6C7AA	T[36] = FDE5380C	T[52] = FC93A039
T[5] = F57C0FAF	T[21] = D62F105D	T[37] = A4BEEA44	T[53] = 655B59C3
T[6] = 4787C62A	T[22] = 02441453	T[38] = 4BDECFA9	T[54] = 8F0CCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D
T[8] = FD469501	T[24] = E7D3FBCB	T[40] = BEBFBC70	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = 289B7EC6	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = EAA127FA	T[58] = FE2CE6E0
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = D4EF3085	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = 04881D05	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = D9D4D039	T[61] = F7537E82
T[14] = FD987193	T[30] = FCEFA3F8	T[46] = E6DB99E5	T[62] = BD3AF235
T[15] = A679438E	T[31] = 676F02D9	T[47] = 1FA27CF8	T[63] = 2AD7D2BB
T[16] = 49B40821	T[32] = 8D2A4C8A	T[48] = C4AC5665	T[64] = EB86D391

Misalkan notasi

$$[abcd \quad k \quad s \quad i]$$

menyatakan operasi

$$a \leftarrow b + ((a + g(b, c, d) + X[k] + T[i])<<<s)$$

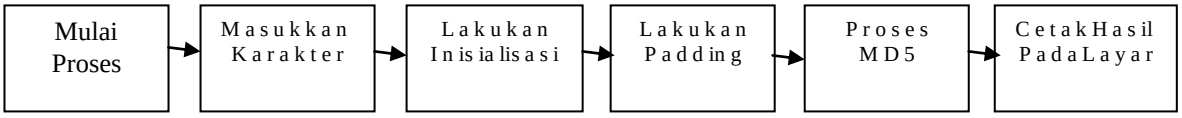
yang dalam hal ini <<< *s* melambangkan operasi *circular left shift* 32-bit, maka operasi dasar pada masing-masing putaran dapat ditabulasikan sebagai berikut:

→ Setelah putaran keempat, *a*, *b*, *c*, dan *d* ditambahkan ke *A*, *B*, *C*, dan *D*, dan selanjutnya algoritma memproses untuk blok data berikutnya (Y_{q+1}).

→ Keluaran akhir dari algoritma *MD5* adalah hasil penyambungan bit-bit di *A*, *B*, *C*, dan *D*.

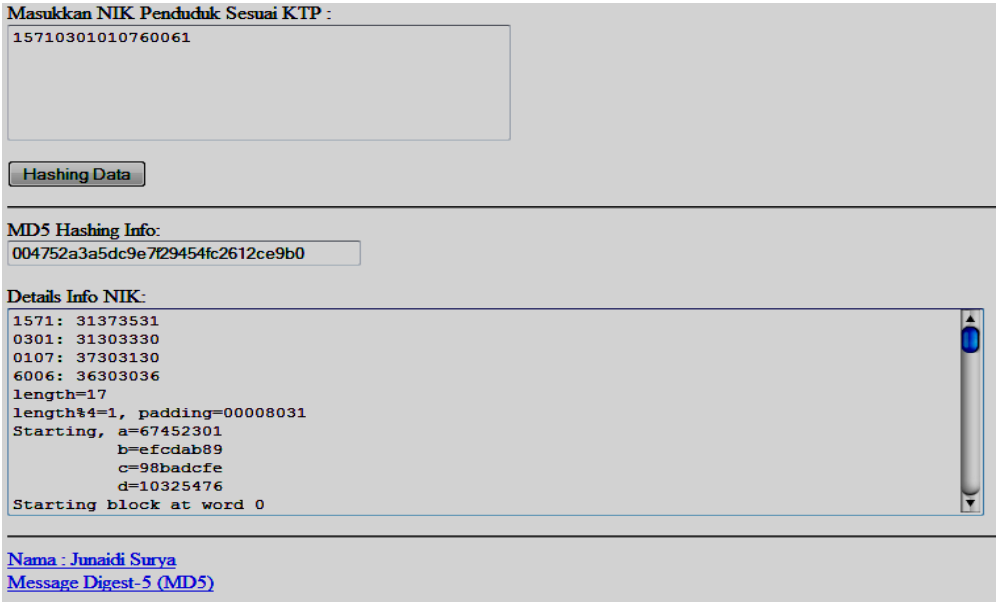
Proses MD-5 Dengan Masukan Berupa String Data NIK

Proses MD5 dengan masukan berupa string adalah proses yang masukan nya berupa karakter-karakter yang dimasukan melalui *keyboard.* Hal ini dapat dilihat pada gambar 8 berikut ini.



Gambar 8 : Proses MD-5 Masukkan String NIK Proses MD5

Contoh proses aplikasi MD5 data penduduk yang dihashing adalah data NIK yang menjadi private key KTP penduduk seperti gambar berikut;



Gambar 9 : Proses MD-5 NIK Penduduk

4.2 Cara kerja Proses Counter dengan *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) merupakan pengembangan dari DES (Data Encryption Standard). Salah satu yang membedakan AES dengan DES adalah operasi AES yang berorientasi byte, tidak seperti DES yang berorientasi bit. Selain itu, jaringan Feistel yang digunakan dalam DES tidak dipergunakan lagi dalam AES. Sebagai gantinya, AES menggunakan struktur SPN yang memiliki derajat paralelisme lebih besar. Garis besar algoritma Rijndael yang beroperasi blok 128-bit dengan kunci 128-bit adalah sebagai berikut:

1. AddRoundKey: melakukan XOR antara state awal (plainteks) dengan cipher key.

Tahap ini disebut juga initial round dari data NIK penduduk.

Encryption Algorithm (128-bit version)

```
Cipher(byte in[16], byte out[16], word w[44])
begin
byte state[4,4]
state = in
AddRoundKey(state, w[0, 3])
for round = 1 step 1 to 10
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w[round*4, (round+1)*4-1])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state, w[40, 43])
out = state
end
```

2. Putaran sebanyak Nr – 1 kali.

Proses yang dilakukan pada setiap putaran adalah:

ByteSub: substitusi byte dengan menggunakan tabel substitusi (Sbox). sehingga substitusi S-box seperti pada tabel 4 berikut ini;

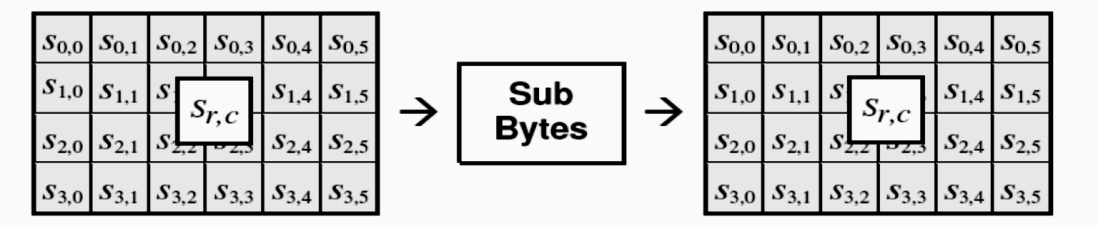
Tabel 4 : S-box yang digunakan dalam transformasi ByteSub() AES

hex	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

Dengan formulasi SubBytes Routine ;

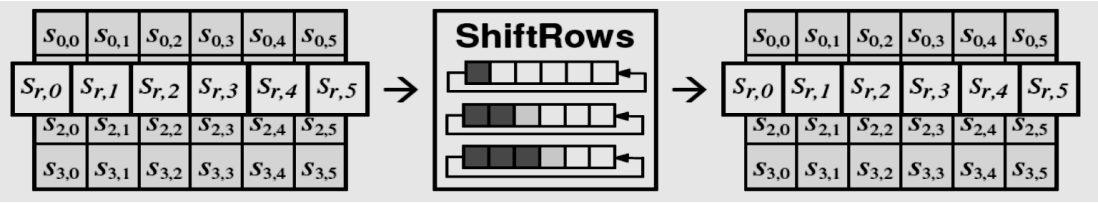
bit *i*, set b_i to b_i *xor* $b_{(i+4) \bmod 8}$ *xor* $b_{(i+5) \bmod 8}$ *xor* $b_{(i+6) \bmod 8}$ *xor* $b_{(i+7) \bmod 8}$ + c_i where c = 63 hex.

sedangkan ilustrasi ByteSub dapat dilihat pada gambar10.



Gambar 10: Ilustrasi Transformasi ByteSub() AES

b. ShiftRow: pergeseran baris-baris array state secara wrapping. Ilustarsi ShiftRow dapat dilihat pada gambar 11.



Gambar 11: Ilustrasi Transformasi ShiftRow() AES

c. MixColumn: mengacak data di masing-masing kolom array state.

MixColumns Routine formulasi

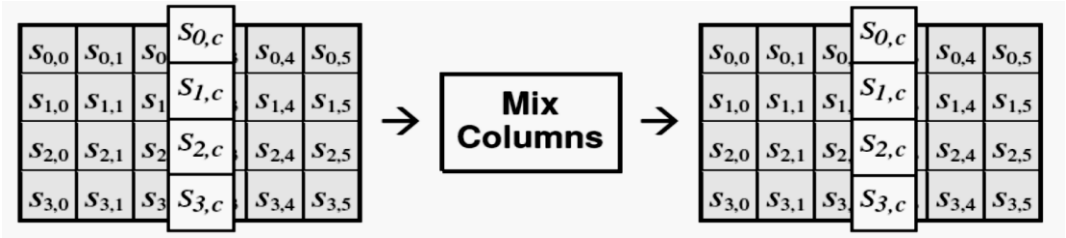
Set $s_{0,c}$ to *2**$s_{0,c}$ *xor 3**$s_{1,c}$ *xor* $s_{2,c}$ *xor* $s_{3,c}$

Set $s_{1,c}$ to $_{0,c}$ *xor* $2*s_{1,c}$ *xor* $3*s_{2,c}$ *xor* $s_{3,c}$

Set $s_{2,c}$ to $s_{0,c}$ *xor* $s_{1,c}$ *xor* $2*s_{2,c}$ *xor* $3*s_{3,c}$

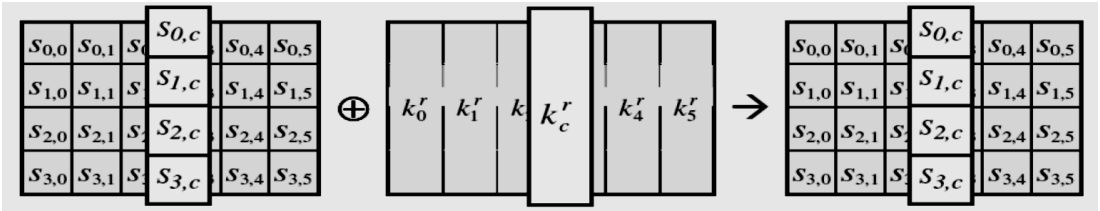
Set $s_{3,c}$ to $3*s_{0,c}$ *xor* $s_{1,c}$ *xor* $s_{2,c}$ *xor* $2*s_{3,c}$

Sehingga Ilustarsi MixColumn dapat dilihat pada gambar 12.



Gambar 12: Ilustrasi Transformasi MixColumn() AES

d. AddRoundKey: melakukan XOR antara state sekarang dengan round key. Ilustarsi AddRoundKey dapat dilihat pada gambar 13.



Gambar 13: Ilustrasi Transformasi AddRoundKey() AES

3. Final round: proses untuk putaran terakhir:

a. ByteSub.

b. ShiftRow.

c. AddRoundKey.

Diagram proses enkripsi AES dapat dilihat pada Gambar 14.

Algoritma Rijndael mempunyai 3 parameter sebagai berikut:

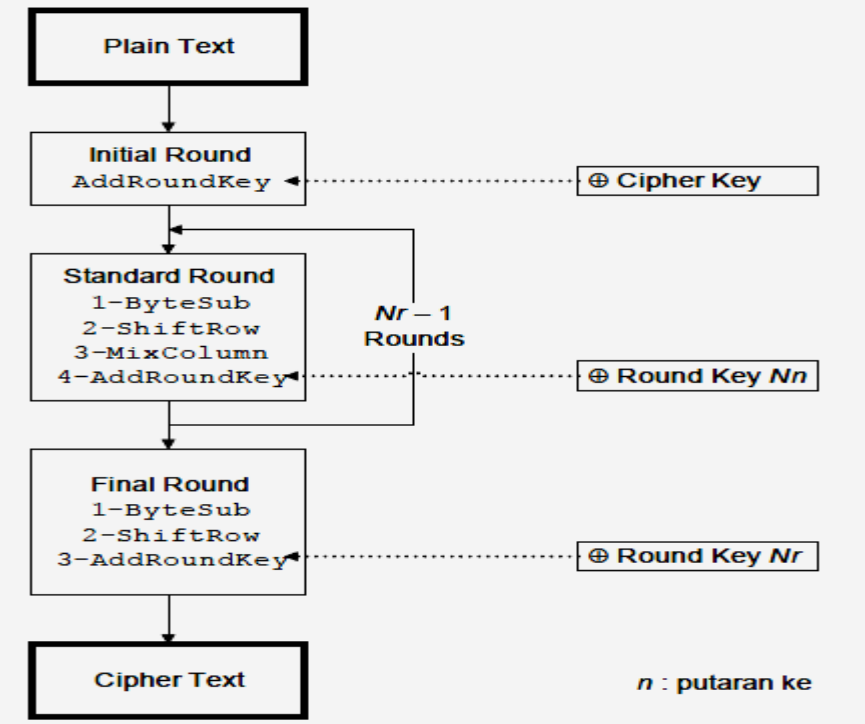
→ Plainteks : array yang berukuran 16 byte, yang berisi data masukan

→ Cipherteks : array yang berukuran 16 byte, yang berisi hasil enkripsi.

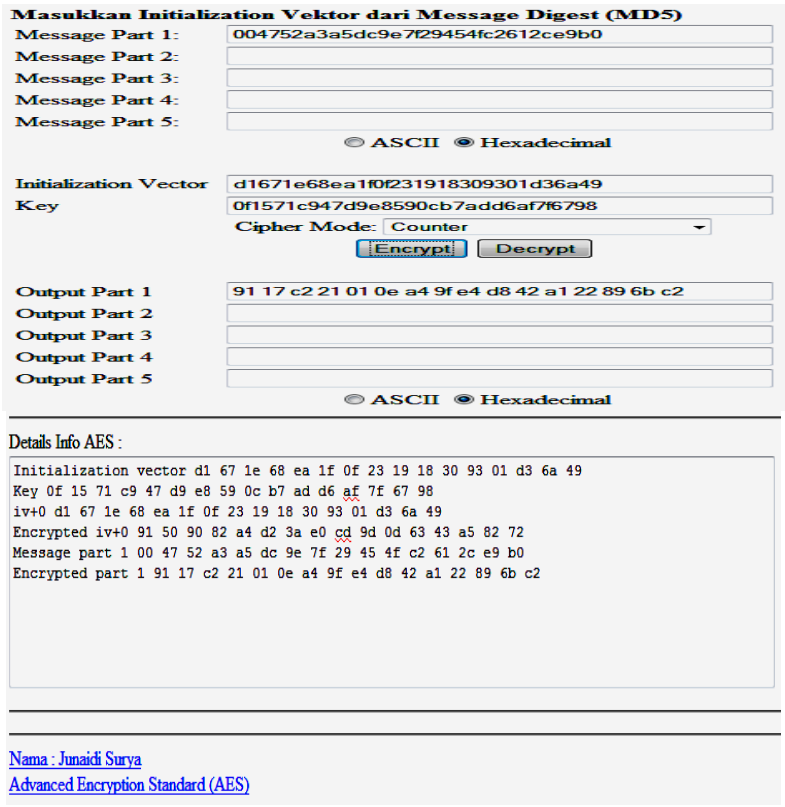
→ Key : array yang berukuran 16 byte, yang berisi kunci ciphering (disebut juga cipher key).

Dengan 16 byte, maka baik blok data dan kunci yang berukuran 128-bit dapat disimpan di dalam ketiga array tersebut (128 = 16 x 8). Selama kalkulasi plainteks menjadi cipherteks, status sekarang dari data mahasiswa disimpan di dalam array of byte dua dimensi, state, yang berukuran NROWS x NCOLS.

Elemen array state diacu sebagai S[r,c], dengan $0 \leq r < 4$ dan $0 \leq c < Nc$ (Nc adalah panjang blok dibagi 32). Pada AES, Nc = 128/32 = 4.



Gambar 14 : Diagram Proses Enkripsi AES



Gambar 15 : Proses Enkripsi AES

Mode Operasi Cipher Blok

Pada cipher blok, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang sama. Enkripsi dilakukan terhadap blok bit plainteks menggunakan bit-bit kunci (yang ukurannya sama dengan blok plainteks). Algoritma enkripsi menghasilkan blok cipherteks yang berukuran sama dengan blok plainteks. Dekripsi masukkan string data mahasiswa dilakukan dengan cara yang serupa seperti enkripsi. Misalkan blok plainteks (*P*) yang berukuran *m* bit dinyatakan sebagai vektor

$$P = (p1, p2, ..., pm)$$

yang dalam hal ini *pi* adalah bit 0 atau bit 1 untuk $i = 1, 2, \ldots, m$, dan blok cipherteks (*C*) adalah

$$C = (c1, c2, ..., cm)$$

yang dalam hal ini *ci* adalah bit 0 atau bit 1 untuk *i* = 1, 2, …, m. Bila plainteks dibagi menjadi *n* buah blok, barisan blok-blok plainteks dinyatakan sebagai

$$(P1, P2, \ldots, Pn)$$

Untuk setiap blok plainteks *Pi*, bit-bit penyusunnya dapat dinyatakan sebagai vektor

$$Pi = (pi1, pi2, ..., pim)$$

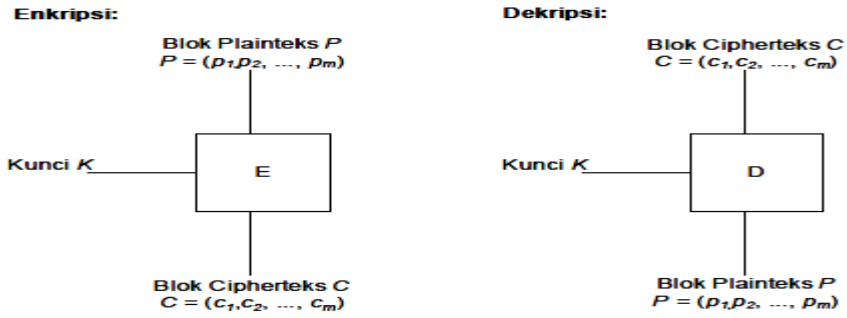
Enkripsi dengan kunci *K* dinyatakan dengan persamaan

$$Ek(P) = C,$$

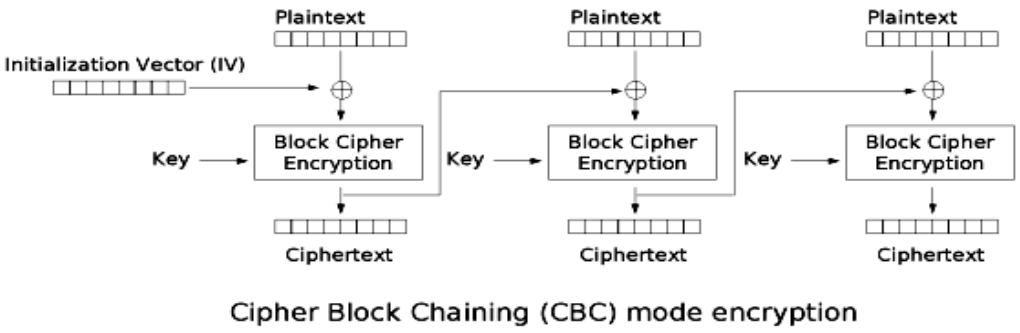
sedangkan dekripsi dengan kunci *K* dinyatakan dengan persamaan

$$Dk(C) = P$$

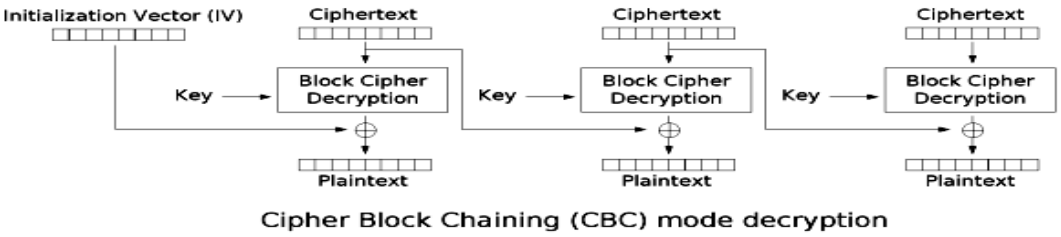
Skema enkripsi dan dekripsi dengan *cipher* blok dapat dilihat pada gambar 20 berikut ini.



Gambar 16 : Skema Enkripsi dan Dekripsi dengan *Cipher* Blok



Gambar 17 : Enkripsi pada Mode CBC



Gambar 18 : Dekarya ilmiah pada Mode CBC

Jika blok pertama memiliki indeks 1, maka rumus matematis untuk enkripsi pada mode CBC adalah :

$$C_i = E_K(P_i \oplus C_{i-1}), C_0 = IV$$

sedangkan rumus matematis untuk dekripsi pada mode CBC adalah :

$$P_i = D_K(C_i) \oplus C_{i-1}, C_0 = IV$$

Keterangan :

Ci : Cipherteks pada blok i data masukkan string

Pi : Plainteks pada blok i data awal string

Ek(...) : Fungsi enkripsi yang digunakan untuk menghasilkan digital

Dk(...) : Fungsi dekripsi yang digunakan untuk proses autentikasi

IV : Initialization Vector

Hingga saat ini, CBC merupakan mode operasi enkripsi cipher blok yang paling sering digunakan. Kelemahan utamanya adalah bahwa proses enkripsi pada CBC dilakukan secara sekuensial (sehingga tidak dapat dilakukan paralelisasi), dan bahwa harus dilakukan padding pada pesan sehingga berukuran sama dengan ukuran blok cipher. Salah satu cara untuk mengatasinya adalah dengan menggunakan metode ciphertext stealing.

V. KESIMPULAN

1. Digital Signature bukanlah tanda tangan secara tradisional di atas kertas, yang di scan secara digital, melainkan sebuah kode-kode digital yang diperoleh dari fungsi hashing menggunakan algoritma tertentu yang kita tanamkan pada KTP.
2. Message Digest 5 (MD5) adalah sebuah fungsi hash satu arah yang mengubah masukan dengan panjang variabel menjadi keluaran dengan panjang yang tetap yaitu 128 bit dengan empat (4) buah blok yang masing-masing terdiri dari 32 bit sehingga menjadi 128 bit yang disebut nilai hash, sehingga hash menggunakan MD5 cukup aman dan banyak digunakan dalam *certificate authority.*
3. Aplikasi Digital Signature pada KTP didasarkan pada pola sidik jari, yang pada kenyataan bahwa perubahan sedikitpun pola sidik jari maka sidik jarinya menjadi rusak dan perubahan 1 bit pada pesan akan mengubah secara rata-rata setengah dari bit-bit *message digest.* Dengan kata lain fungsi hash sangat peka terhadap perubahan sekecil apapun pada data masukan yang menyebabkan autentikasinya menjadi tidak valid.
4. Software aplikasi digunakan, bila proses EDP pada sistem data kependudukan sudah berjalan dengan baik.

Daftar Pustaka

[1] Aghus Sofwan, Agung Budi P, Toni Susanto,(2006),Aplikasi Kriptogra¯ Dengan Algoritma Message Digest 5 (MD5),*Transmisi, Vol. 11, No. 1, Juni 2006*

[2] Budi Raharjo, (2005),*Keamanan Sistem Informasi Berbasis Internet*, PT Insan Indonesia Bandung dan PT INDOCISC Jakarta

[3] Buccafurry, Francesco (2008),*Digital Signature Trust Vulnerability A New Attack*, Jurnal ISSA.

[4] Dony Ariyus (2008),*Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi*, Andi Yogyakarta.

[5] http://www.cs.eku.edu/faculty/styer/460/Encrypt

[6] http://en.wikipedia.org

[7] Schneier Bruce, (1996), *Applied Cryptography Second Edition: Protocols, Algo-rthms, and Source Code in C (cloth)*, John Wiley Sons, Inc.