BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis

Penelitian bertujuan untuk merancang sebuah sistem yang dapat melakukan Perancangan Aplikasi Keamanan Data Dengan Metode *End Of File* (EOF) dan Algoritma *Message-Digest algortihm 5 (*MD5). Ketika seseorang yang hendak mengirim file kepada orang lain, tidak ingin isi file tersebut diketahui oleh orang lain, yang mana isi file tersebut bersifat sangat rahasia atau pribadi, yang hanya boleh diketahui antara pihak pengirim dan pihak penerima pesan. Oleh karena itu, biasanya pengirim tersebut mengirim file secara tersembunyi agar tidak ada pihak lain yang mengetahui. Salah satu cara untuk mengatasi situasi di atas adalah mengembangkan suatu aplikasi yang mampu menyamarkan file tersebut pada suatu media.

Agar perangkat lunak keamanan data ini dapat berjalan dengan baik, ada beberapa persyaratan yang harus dipenuhi. Adapun persyaratan kebutuhan sistem yang dirancang ini adalah sebagai berikut :

1. Perangkat keras (*Hardware*)

   Intel ® Core TM 13-321 7U CPU@ 1.80 GHz, Hardisk 500GB dan RAM 2GB.

2. Perangkat Lunak (*Software*)

   Bahasa pemrograman yang digunakan adalah Visual Studio 2010 dan berjalan pada sistem operasi *windows*.

III.2. Strategi Pemecahan Masalah

Adapun langkah-langkah yang penulis lakukan dalam menyelesaikan masalah Perancangan Aplikasi Keamanan Data Dengan Metode End Of File (EOF) dan Algoritma MD5 ini terdiri dari beberapa tahapan sebagai berikut :

1. Mengumpulkan Teori dan Contoh-Contoh Kasus

   Dalam tahap ini penulis mengumpulkan teori-teori yang berhubungan dengan masalah teknik *steganografi*, *kriptografi,* audio, teknik EOF dan algoritma MD5. Teori-teori ini penulis kumpulkan dari beberapa sumber seperti buku-buku di perpustakaan, artikel-artikel di internet serta referensi dari tugas akhir mahasiswa lain yang berhubungan dengan masalah yang dihadapi.

2. Merancang Program

   Langkah pertama dalam perancangan program ini adalah merancang proses kerja sistem. Proses kerja sistem penulis rancang menggunakan sebuah bagan alir (*flowchart*) yang menjelaskan secara rinci proses-proses yang akan dilakukan program dalam penyisipan sebuah teks pada audio.

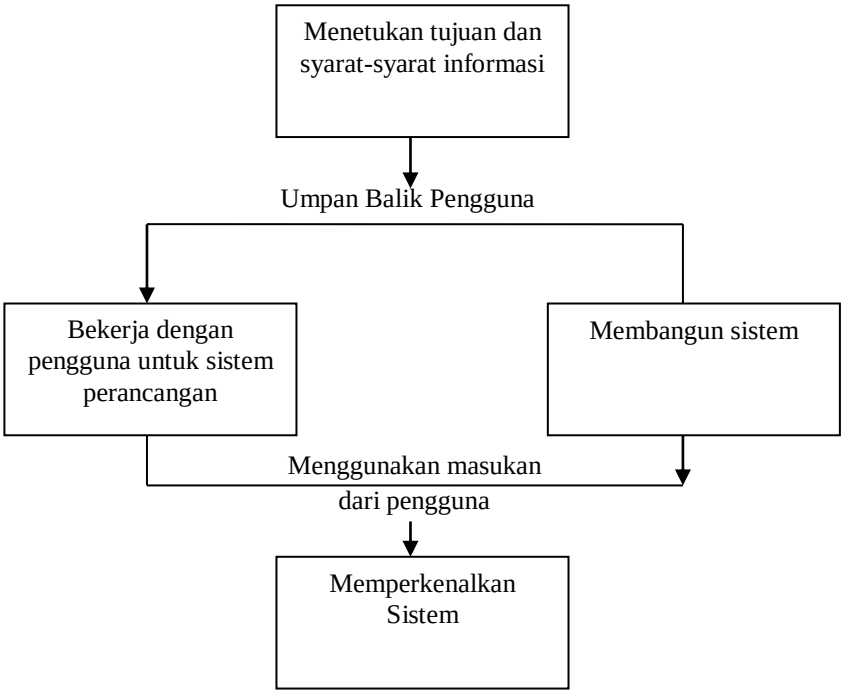
   Langkah berikutnya adalah merancang bentuk tampilan program. Bentuk tampilan program yang penulis rancang adalah sbuah *form* dengan tombol-tombol yang dapat diguanakan user untuk berinteraksi dengan program yang dirancang. Dalam langkah ini penulis juga merancang algoritma pemrograman yang akan penulis gunakan dalam implementasi rancangan program dalam bahasa pemrograman yang digunakan.

3. Mengimplementasikan Rancangan Program

Bahasa pemrograman yang penulis pilih dalam implementasi rancangan program adalah Microsoft VB.net 2010. Bahasa pemrograman ini penulis pilih karena lebih familiar dibandingkan bahasa pemrograman lain dan sering penulis gunakan pada saat perkuliahan.

Pada tahapan ini, penulis mengimplementasikan rancangan tampilan program serta melakukan *coding* sesuai dengan bahasa pemrograman yang digunakan. Tahapan implementasi program yang penulis lakukan adalah membuat tampilan *form*, membuat *module-module* yang dibutuhkan serta membuat *coding* terhadap tombol-tombol dan menu-menu pada *form*. Melakukan pengujian program.

Pada tahapan akhir ini, penulis melakukan serangkaian pengujian terhadap program yang dihasilkan. Pengujian-pengujian ini dilakukan untuk mencari kesalahan-kesalahan (*error*) pada program dan melakukan perbaikan-perbaikan yang dibutuhkan. Adapun skema metode penyelesaian masalah yang penulis lakukan dapat dilihat pada Gambar III.1.



Gambar III.1 Skema Metode Penyelesaian Masalah

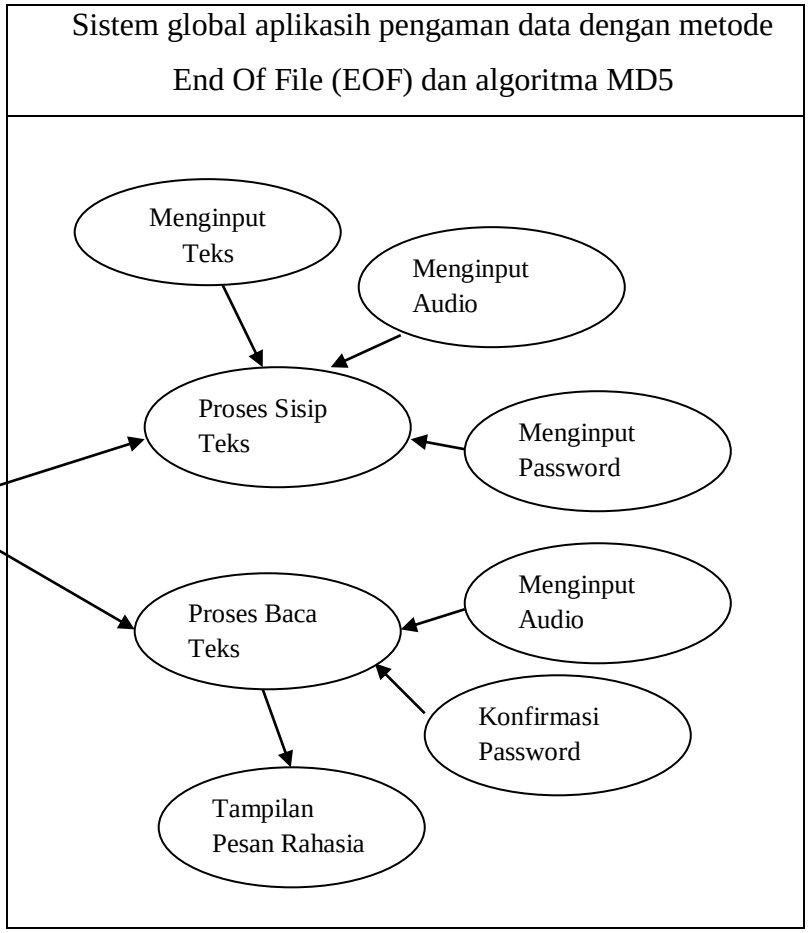
III.3. Struktur Data yang Digunakan

Struktur data yang digunakan penulis dalam perancangan perangkat lunak adalah *Unified Modeling Language* (UML). *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan dan membangun sistem perangkat lunak. UML yang digunakan meliputi perancangan *Diagram Use Case*, *Activity Diagram* dan *Squence Diagram.*

III.3.1. *Diagram Use Case*

Diagram use case digunakan untuk memberikan gambaran kebutuhan perangkat lunak secara visual. *Use case* diagram yang memiliki *use case* proses penyisipan dan *use case* proses pembacaan. Proses sisip memiliki 3 *include*

didalamnya, sedangkan proses baca memiliki 2 *include* dan 1 *extend*. Pengirim merupakan pengguna yang melakukan penyisipan teks kedalam audio, sedangkan penerima adalah pengguna yang melakukan pembacaan teks pada audio. *Use case* memilih teks digunakan oleh pengirim untuk memilih teks yang akan disisipkan kedalam audio, kemudian *use case* memilih audio digunakan oleh pengirim untuk memilih berkas audio yang akan digunakan sebagai media penyisipan. *Use case* memilih berkas audio oleh penerima digunakan untuk memilih berkas audio yang akan dibaca untuk mendapatkan teks yang disisipkan. Kita bisa lihat pada Gambar III.2.

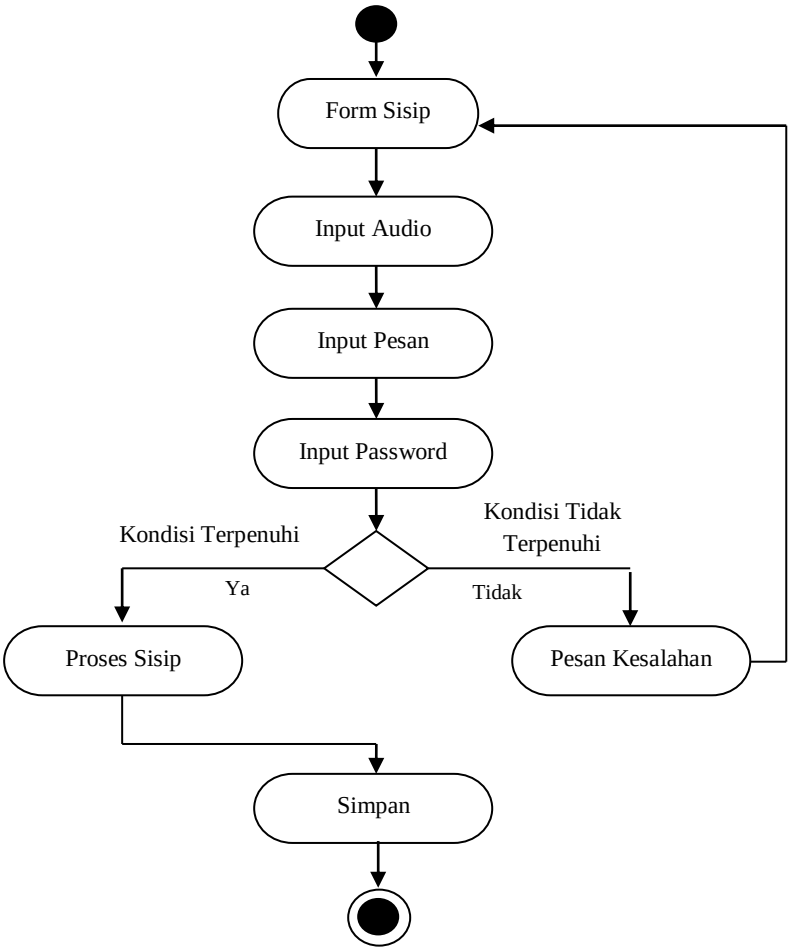


Gambar III.2 *Diagram Use Case* Keamanan Data

III.3.2. *Activity Diagram*

Activity diagram menggambar aliran fungsionalitas dalam suatu sistem. Dapat digunakan dalam analisa kebutuhan untuk menggambarkan aliran kejadian melalui suatu *use case* atau menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, keputusan yang mungkin terjadi dan bagaimana seluruh alir berakhir. *Activity diagram* dari program Perancangan Aplikasi Keamanan Data Dengan Metode *End Of File* (EOF) dan Algoritma MD5 digambarkan seperti gambar di bawah ini :

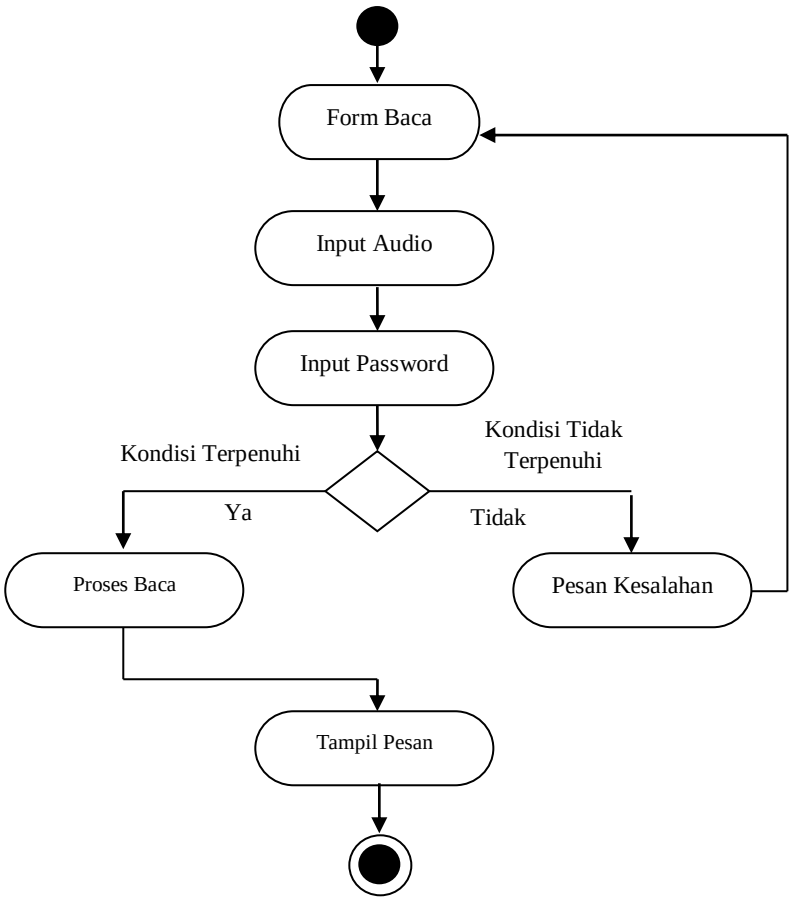
1. Proses sisip Teks



Gambar III.3 *Activity Diagram* Proses Sisip Teks

Pada gambar III.3 menggambarkan proses penyisipan teks pada sistem Keamanan Data Menggunakan Metode EOF dan Algoritma MD5 yang akan dirancang.

2. Proses Baca Teks



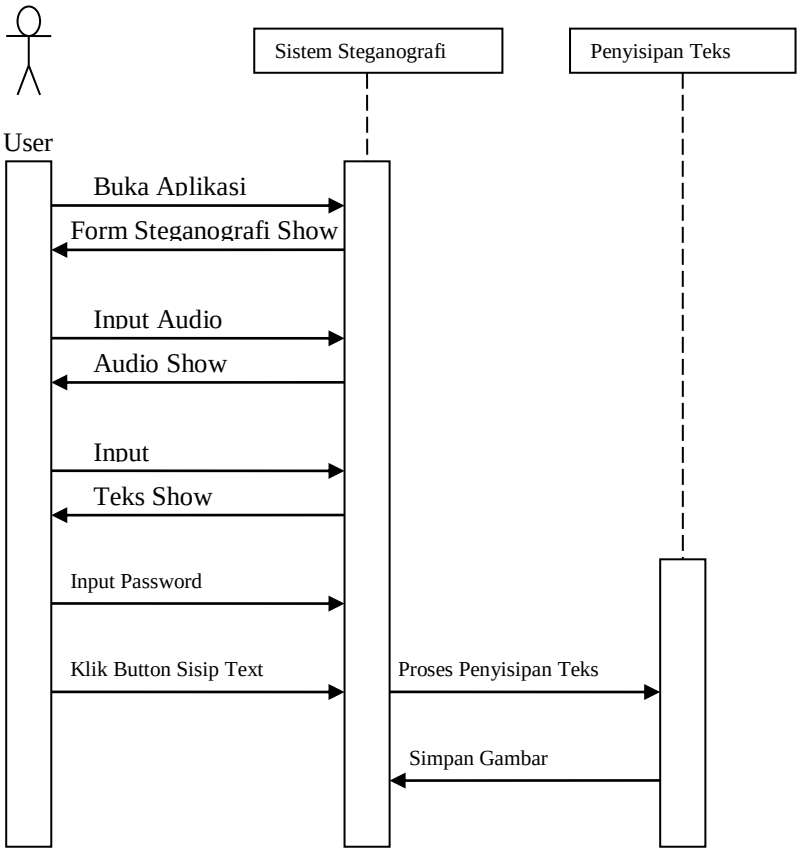
Gambar III.4 *Activity Diagram* Proses Baca Teks

Pada gambar III.4 menggambarkan proses baca teks pada sistem Keamanan Data Menggunakan Metode EOF dan Algoritma MD5 dan melihat hasil akhir dari proses tersebut yaitu menampilkan pesan yang disembunyikan pada audio.

III.3.3. *Sequence Diagram*

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya diletakkan di paling kiri dari diagram.Pada diagram ini. Berikut gambar *sequence* diagram enkripsi dan dekripsi di bawah ini :

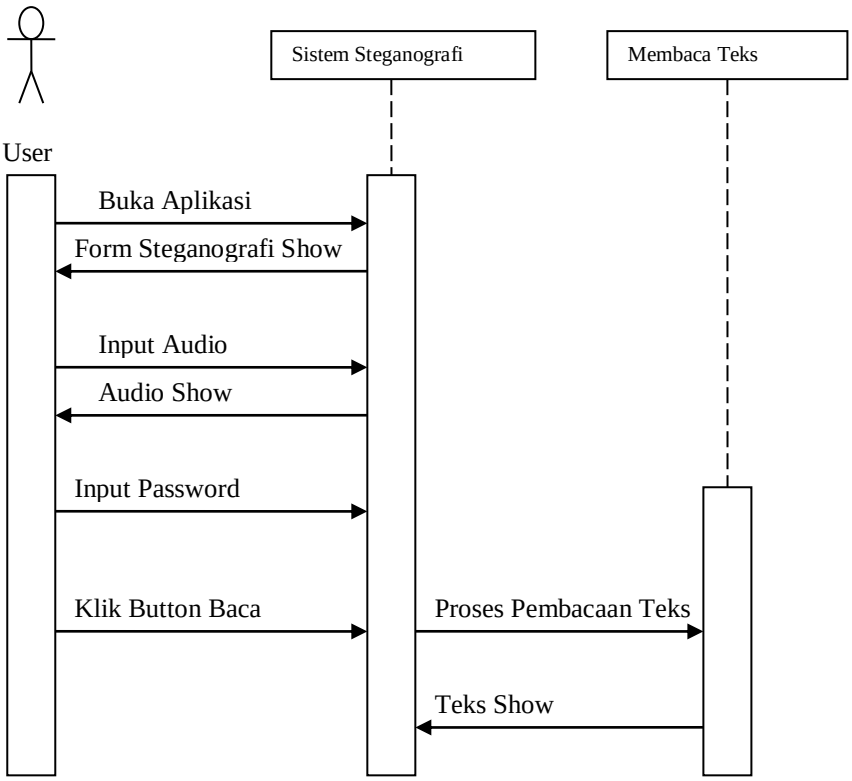
1. Proses Penyisipan Teks



Gambar III.5 *Sequence Diagram* Penyisipan Teks

Pada gambar III.5 menggambarkan perancangan sistem *sequence diagram* pada sistem penyisipan teks, sehingga dapat dilihat proses kerja dari sistem tersebut. User membuka aplikasi, maka *form* menu utama muncul, kemudian Menginput audio, menginput teks, menginputkan password dan klik *Button* Sisip. Kemudian proses penyisipan teks dilakukan, setelah itu penyimpanan audio yang sudah disisip teks.

2. Proses Pembacaan Teks



Gambar III.6 *Sequence Diagram* Pembacaan Teks

Pada gambar III.6 menggambarkan perancangan sistem squence diagram pada sistem pembacaan teks, sehingga dapat dilihat proses kerja dari sistem tersebut. User membuka aplikasi, maka *form* menu utama muncul, kemudian

Menginput audio, menginputkan password dan klik *Button* Baca. kemudian proses baca teks dilakukan, setelah berhasil teks akan tampil.

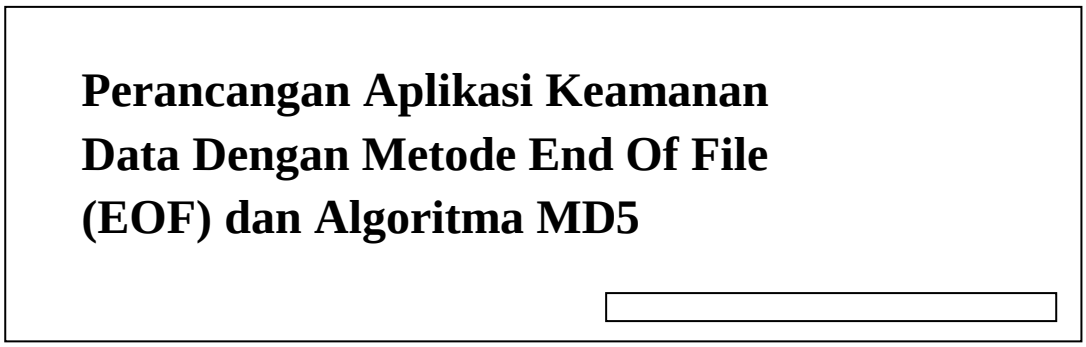
III.4. Perancangan

Adapun tahapan perancangan yang penulis lakukan adalah perancangan antarmuka perangkat lunak, *flowchart* atau algoritma serta algoritma dari program.

III.4.1. Rancangan Layar

Adapun rancangan tampilan Perancangan Aplikasi Keamanan Data Dengan Metode End Of File (EOF) dan Algoritma MD5 adalah sebagai berikut :

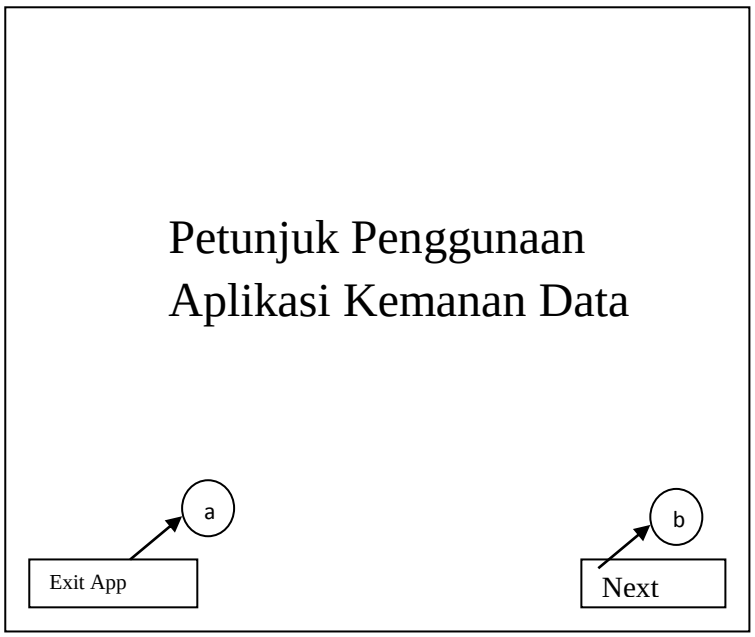
1. Rancangan Form Awal



Gambar III.7 Form Tampilan Awal

2. Rancangan Form Petunjuk

Form petunjuk merupakan *form* tampilan yang dirancang sebagai petunjuk cara penggunaan aplikasi.



Gambar III.8 Form Petunjuk Keamanan Data

Keterangan Gambar III.8:

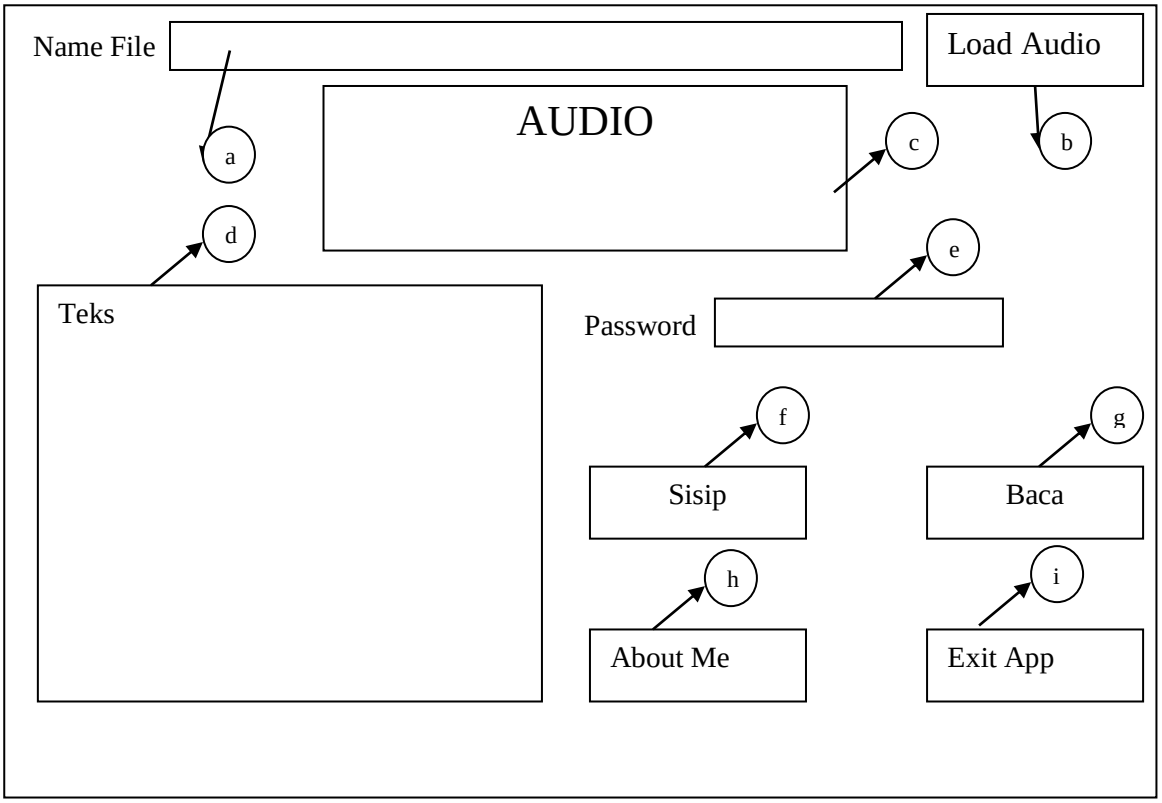
Dari gambar diatas terdapat beberapa menu yaitu:

a. Tombol exit brfungsi untuk keluar dari aplikasi.

b. Tombol next berfungsi untuk menuju ke form keamanan data.

3. Rancangan *Form* Keamanan Data

Form Keamanan Data merupakan tampilan yang dirancang sebagai interface agar user dapat berinteraksi dengan sistem. Dalam melakukan interaksi dengan *user*, *form* menggunakan tombol-tombol yang dapat dipilih oleh user. Adapun bentuk rancangan *form* sisip dan baca seperti pada Gambar III.9.



Gambar III.9 *Form* Keamanan Data

Keterangan Gambar III.9:

Dari gambar diatas terdapat beberapa menu yaitu:

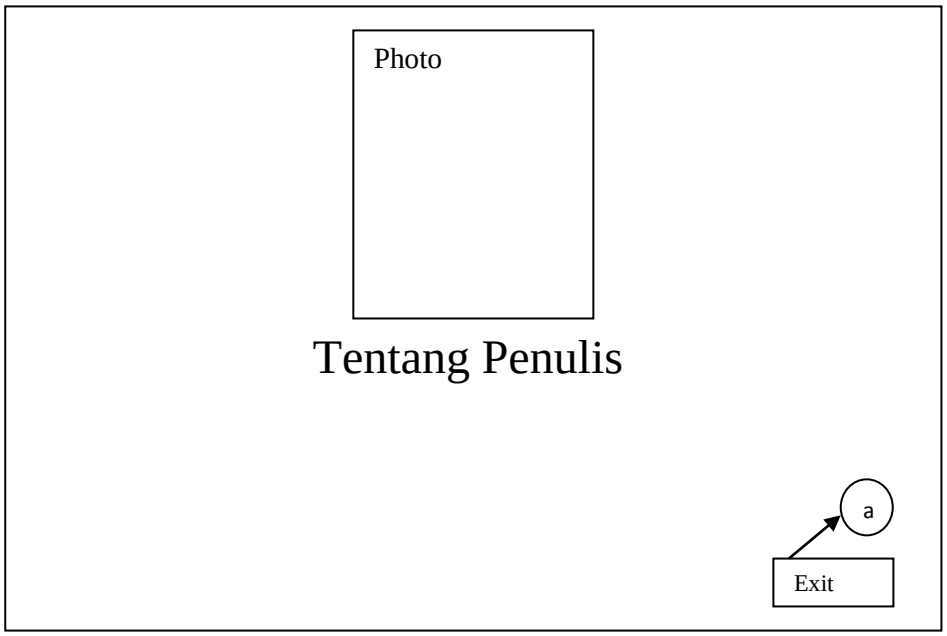
a. Nama File berfungsi menampilkan nama file dari audio.
b. *Load Audio* berfungsi untuk mengambil audio yang akan disisip teks, juga mengambil audio yang sudah disisip teks untuk dibaca.
c. Kotak audio berfungsi menampilkan audio yang dibuka dari *Load Audio*.
d. Kotak teks berfungsi untuk teks yang akan disimpan atau menampilkan teks yang sudah dibaca di audio.
e. Password berfungsi untuk menambah keamanan yang akan disisip diaudio.
f. Sisip berfungsi melakukan proses penyisipan teks ke dalam audio.
g. Baca berfungsi untuk melakukan pembacaan teks di dalam audio.

h. Tombol *About Me* berfungsi untuk mengetahui tentang penulis.

i. Tombol exit brfungsi untuk keluar dari aplikasi.

a. Rancangan *Form About*

Form about merupakan *form* tampilan yang dirancang sebagai data pribadi penulis.



Gambar III.10 *Form About*

Keterangan Gambar III.10:

Dari gambar diatas terdapat beberapa menu yaitu:

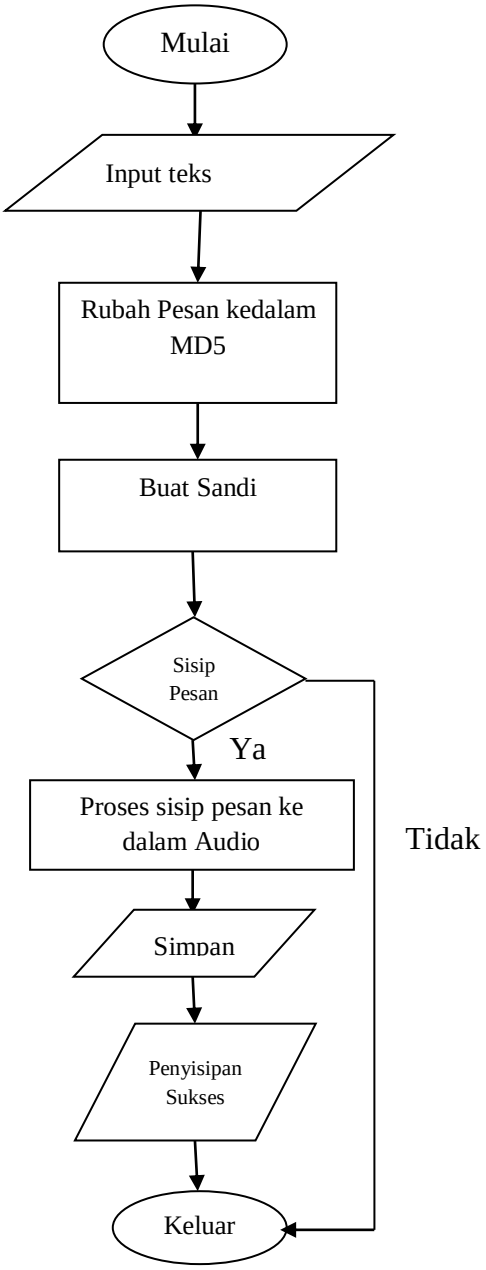
a. Tombol exit brfungsi untuk keluar dari aplikasi.

III.4.2. *Flowchart*

Bagan alir (flowchart) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

1. Flowchart Aplikasi Sisip Teks

Adapun bentuk perancangan *flowchart* aplikasi proses kerja sisip teks yang penulis rancang seperti ditunjukan pada gambar III.11.



Gambar III.11 *Flowchart* Aplikasi Sisip Teks

Keterangan Gambar III.11 :

a. Mulai.

b. *User* mebuka *form steganografi*.

c. *User input* audio sebagai tempat penampung teks yang akan disisip.

d. *User* memasukan teks yang akan disisip.

e. *User* memasukan *password*.

f. Klik tombol sisip teks untuk melakukan proses penyisipan teks ke dalam audio.

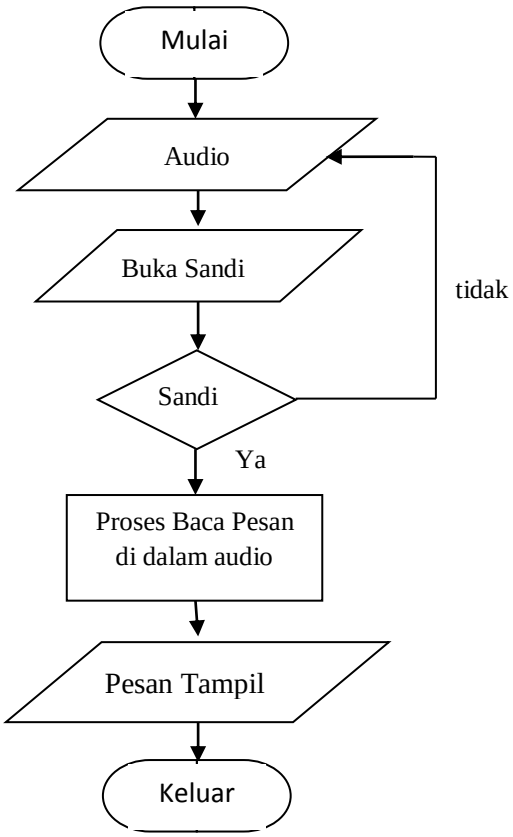
g. Jika salah akan muncul pesan kesalahan, dan jika benar akan langsung keproses.

h. User dapat menyimpan hasil penyisisipan teks ke lokasi penyimpanan.

i. Jika selesai maka aplikasi akan ditutup.

2. *Flowchart* Aplikasi Baca Teks

Adapun bentuk perancangan *flowchart* aplikasi proses kerja baca teks yang penulis rancang seperti ditunjukan pada gambar III.12.



Gambar III.12 *Flowchart* Aplikasi Baca Teks

Keterangan Gambar III.12 :

a. Mulai.

b. *User* mebuka *form steganografi*.

c. *User input* audio yang telah disisip teks dan memasukan *password*.

d. Klik tombol baca teks untuk menampilkan teks rahasia.

e. Jika salah akan muncul pesan kesalahan, dan jika benar akan langsung keproses.

f. Jika selesai maka aplikasi akan ditutup.

III.4.3. Perhitungan Manual Algoritma MD5 dan Metode EOF

Sebagai contoh, dilakukan perhitungan string teks “Ilmu Komputer”. Adapun langkah perhitungan hash menggunakan algoritma MD5 adalah sebagai berikut:

1. Penambahan bit tambahan String teks “Ilmu Komputer” dengan panjang 13 karakter diubah ke dalam kode ASCII. Hasil pengubahan karakter dapat dilihat pada tabel III.1.

Tabel III.1 Hasil pengubahan karakter

Karakter	Kode ASCII
I	73
l	108
m	109
u	117
<spasi>	32
K	75
o	111
m	109
p	112
u	117
t	116
e	101
r	114

String harus memenuhi pada blok 512 bit. Sehingga akan ditambahkan 448- 13x8= 344 bit tambahan yang mana diawali dengan 1 dan pada bit 343 dengan 0.

2. Penambahan panjang string keseluruhan Total dari bit keseluruhan yang didapat dari sebelumnya, yaitu 448-344 = 104 yang dalam bit dinyatakan 0110 1000. Dengan demikian bit yang akan ditambahkan kemudian diurutkan dari terkecil menjadi 104,0,0,0,0,0,0,0. Bit ini menjadikan string memiliki panjang

sama dengan perkalian 512 bit. Hasil penambahan panjang string keseluruhan dapat dilihat pada tabel III.2.

Tabel III.2 Hasil panjang string keseluruhan

Posisi ke-	Nilai
0	73
1	108
2	109
3	117
4	32
5	75
6	111
7	109
8	112
9	117
10	116
11	101
12	114
13	128
14	0
15	0
16	0
...	...
54	0
55	0
56	104
57	0
58	0
59	0
60	0
61	0
62	0
63	0

Mengubah string menjadi 16 blok String dengan panjang 64 bit kemudian diubah menjadi 16 blok bit. Hasil dari pengubahan 64 bit menjadi 16 blok bit dapat dilihat pada tabel III.3.

Tabel III.3 Hasil pengubahan 64 bit menjadi 16 blok

Urutan ke-	Nilai
0	1970105417
1	1836010272
2	1702131056
3	32882
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	104
15	0

3. Inisialisai buffer MD MD5 membutuhkan 4 buah penyangga (buffer) yang masing-masing penjangnya 32 bit. Total panjang penyangga adala 4 x 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 nilainilai (dalam notasi HEX). Inisialisasi buffer MD dapat dilihat pada tabel III.4.

Tabel III.4 Inisialisasi buffer MD

32-bit nilai register (Hexa)	Setelah konversi kedalam pengurutan terendah (Hexa)	Setelah konversi kedalam pengurutan terendah (Decimal)
A: 01 23 45 67	0x67452301	A: 1732584193
B: 89 AB CD EF	0xEFCDAB89	B: 4023233417
C: FE DC BA 98	0x98BADCFE	C: 2562383102
D: 76 54 32 10	0x10325476	D: 271733878

4. Memproses string 16 blok menjadi string 32 bit

Untuk menghasilkan string 32 bit, dibutuhkan 4 perulangan menggunakan fungsi dengan rumusan: F(X,Y,Z) = XY ∨ not(X) Z G(X,Y,Z) = XZ ∨ Y not(Z) H(X,Y,Z) = X xor Y xor Z I(X,Y,Z) = Y xor (X ∨ not(Z)) Fungsi G, H, dan I mirip dengan fungsi F, dimana melakukan manipulasi bit secara paralel untuk menghasilkan perhitungan dari bit X, Y, dan Z. dikarenakan X , Y, dan Z bersifat independen dan tidak bias, maka setiap bit G (X, Y, Z), H (X, Y, Z), dan I (X, Y, Z) akan bersifat independen dan tidak bias. Dengan catatan, fungsi H merupakan fungsi manipulasi bit "xor" atau "parity" dari input. Langkah ini menggunakan elemen sebanyak 64 dalam tabel T [1 ... 64] yang dibangun dari fungsi sinus. T [i] menunjukan elemen ke-i dari tabel yang sama dengan bagian integer dari 4294967296 kali abs (sin (i)) dalam fungsi radian. Nilai T[i] dapat dilihat pada tabel III.5

Tabel III.5 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] = 6D9D6122	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] = 2441453	T[38] = 4BDECFA9	T[54] = 8F0CCC92
T[7] = A8304613	T[23] = D8A1E681	T[39] = F6BB4B60	T[55] = FFEFF47D
T[8] = FD469501	T[24] = E7D3FBC8	T[40] = 289B7EC6	T[56] = 85845DD1
T[9] = 698098D8	T[25] = 21E1CDE6	T[41] = EAA127FA	T[57] = 6FA87E4F
T[10] = 8B44F7AF	T[26] = C33707D6	T[42] = D4EF3085	T[58] = FE2CE6E0
T[11] = FFFF5BB1	T[27] = F4D50D87	T[43] = 4881D05	T[59] = A3014314
T[12] = 895CD7BE	T[28] = 455A14ED	T[44] = D9D4D039	T[60] = 4E0811A1
T[13] = 6B901122	T[29] = A9E3E905	T[45] = E6DB99E5	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

Perulangan pertama menggunakan operasi: a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s) a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s) a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s) a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s)

Putaran 1:

16 kali operasi dasar dengan g(b, c, d)= F(b, c, d) diberikan pada Tabel III.6.

Tabel III.6. Rincian operasi pada fungsi F(b, c, d)

No.	[*abcd*	*k*	*s*	*i*]
1	[ABCD	0	7	1]
2	[DABC	1	12	2]
3	[CDAB	2	17	3]
4	[BCDA	3	22	4]
5	[ABCD	4	7	5]
6	[DABC	5	12	6]
7	[CDAB	6	17	7]
8	[BCDA	7	22	8]
9	[ABCD	8	7	9]
10	[DABC	9	12	10]
11	[CDAB	10	17	11]
12	[BCDA	11	22	12]
13	[ABCD	12	7	13]
14	[DABC	13	12	14]
15	[CDAB	14	17	15]
16	[BCDA	15	22	16]

Putaran 2:

16 kali operasi dasar dengan g(b, c, d)= G(b, c, d) diberikan pada Tabel III.7.

Tabel III.7. Rincian operasi pada fungsi G(b, c, d)

No.	[*abcd*	*k*	*s*	*i*]
1	[ABCD	1	5	17]
2	[DABC	6	9	18]
3	[CDAB	11	14	19]
4	[BCDA	0	20	20]
5	[ABCD	5	5	21]
6	[DABC	10	9	22]
7	[CDAB	15	14	23]
8	[BCDA	4	20	24]
9	[ABCD	9	5	25]
10	[DABC	14	9	26]
11	[CDAB	3	14	27]
12	[BCDA	8	20	28]
13	[ABCD	13	5	29]
14	[DABC	2	9	30]
15	[CDAB	7	14	31]
16	[BCDA	12	20	32]

Putaran 3:

16 kali operasi dasar dengan g(b, c, d) = H (b, c, d) diberikan pada Tabel III.8.

Tabel III.8. Rincian operasi pada fungsi H (b, c, d)

No.	[*abcd*	*k*	*s*	*i*]
1	[ABCD	5	4	33]
2	[DABC	8	11	34]
3	[CDAB	11	16	35]
4	[BCDA	14	23	36]
5	[ABCD	1	4	37]
6	[DABC	4	11	38]
7	[CDAB	7	16	39]
8	[BCDA	10	23	40]
9	[ABCD	13	4	41]
10	[DABC	0	11	42]
11	[CDAB	3	16	43]
12	[BCDA	6	23	44]
13	[ABCD	9	4	45]
14	[DABC	12	11	46]
15	[CDAB	15	16	47]
16	[BCDA	2	23	48]

Putaran 4:

16 kali operasi dasar dengan g(b, c, d) = I(b, c, d) diberikan pada Tabel III.9.

Tabel III.9. Rincian pada fungsi I(b, c, d)

No.	[*abcd*	*k*	*s*	*i*]
1	[ABCD	0	6	49]
2	[DABC	7	10	50]
3	[CDAB	14	15	51]
4	[BCDA	5	21	52]
5	[ABCD	12	6	53]
6	[DABC	3	10	54]
7	[CDAB	10	15	55]
8	[BCDA	1	21	56]
9	[ABCD	8	6	57]
10	[DABC	15	10	58]
11	[CDAB	6	15	59]
12	[BCDA	13	21	60]
13	[ABCD	4	6	61]
14	[DABC	11	10	62]
15	[CDAB	2	15	63]
16	[BCDA	9	21	64]

Masing-masing perulangan diikuti dengan 16 operasi. Adapun hasil perulangan dapat dilihat pada tabel III.10

Tabel III.10 Hasil perhitungan 4 perulangan

Fungsi	Iterasi	Perubahan Register	Nilai
FF	1	a	1540754351
	2	d	-1086169388
	3	c	-1662035182
	4	b	110334631
	5	a	-1857000786
	6	d	618376166
	7	c	-596362555
	8	b	-2114550423
	9	a	688761912
	10	d	2092291217
	11	c	-1265442809
	12	b	212147281
	13	a	-1545336363
	14	d	-154434875
	15	c	2020576933
	16	b	-1975380017
GG	17	a	-986497183
	18	d	601759827
	19	c	1748277731
	20	b	-1148624887
	21	a	-1662350359
	22	d	891792236
	23	c	-1069889286
	24	b	202314186
	25	a	1770161914
	26	d	906874754
	27	c	-1380510167
	28	b	196093360
	29	a	-289350973
	30	d	-755379888
	31	c	-1878044587
	32	b	-926817903

HH	33	a	1474898728
	34	d	-218163596
	35	c	-2126282248
	36	b	-700907421
	37	a	-804747245
	38	d	1334163013
	39	c	1528002552
	40	b	-1681898629
	41	a	459857267
	42	d	-1660116531
	43	c	96530292
	44	b	718464494
	45	a	-1600621016
	46	d	911972035
	47	c	-1479706569
	48	b	-141526692
II	49	a	-1584268940
	50	d	1241899922
	51	c	663520065
	52	b	-447625936
	53	a	1048453670
	54	d	1177466359
	55	c	-788891107
	56	b	-630686348
	57	a	617739050
	58	d	-1989162990
	59	c	-2065611098
	60	b	1144071368
	61	**a**	**-508960117**
	62	**d**	**-938521949**
	63	**c**	**-1407851924**
	64	**b**	**1932174465**

Hasil pada 4 bit terakhir, yaitu a,d,c,b kemudian dilakukan perhitungan dengan rumusan:

A = A + a

B = B + b

C = C + c

D = D + d

Dengan hasil akhir, a = 1223624076, b = 1660440586, c = 1154531178, d = -666788071. Hasil akhir kemudian dikonversi kedalam bentuk hexadesimal kemudian digabungan menghasilkan string MD5, yaitu: 8c05ef480a50f8626abfd044199f41d8. Hash inilah yang dijadikan acuan dalam mencari file duplikat.

Kemudian hasil dari enkripsi algoritma MD5 akan disisip ke dalam metode EOF dengan teks enkripsi 8c05ef480a50f8626abfd044199f41d8.

Misalkan matriks tingkat derajat audio sebagai berikut :

196 10 97 182 101 40

67 200 100 50 90 50

25 150 45 200 75 28

176 56 77 100 25 200

101 34 250 40 100 60

44 66 99 125 190 200

Kode biner pesan disisipkan di akhir audio, sehingga audio menjadi :

196 10 97 182 101 40

67 200 100 50 90 50

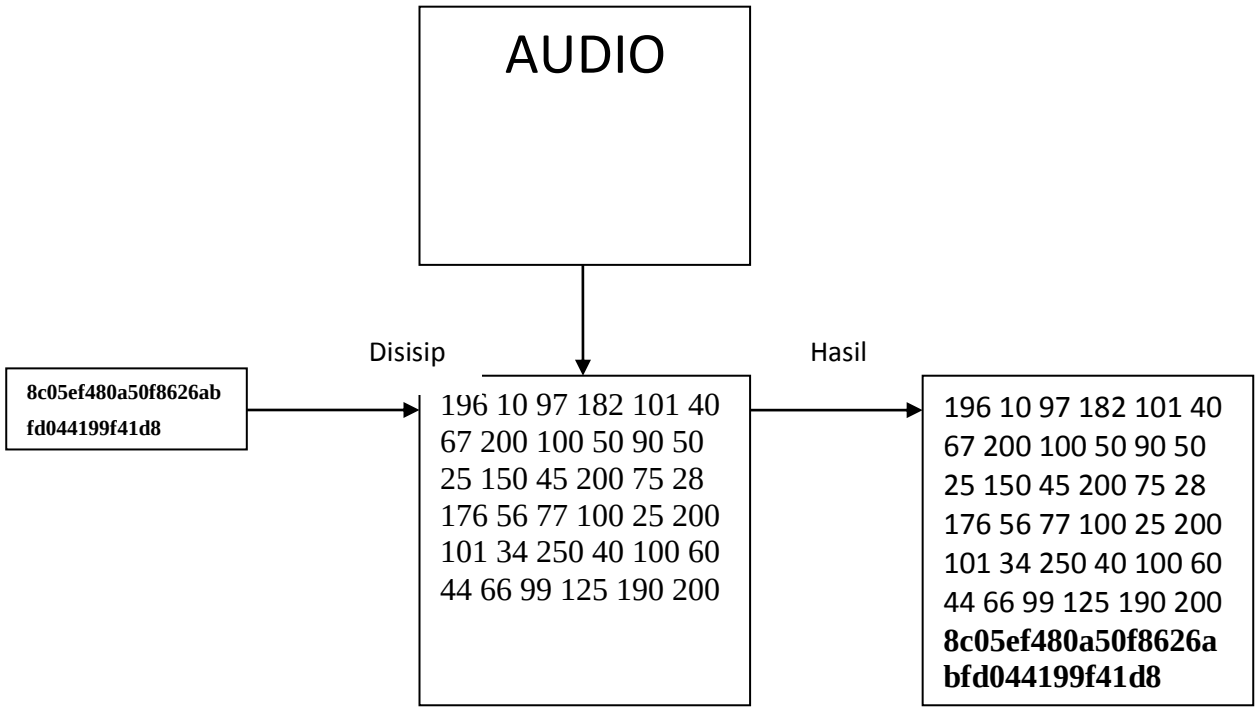
25 150 45 200 75 28

176 56 77 100 25 200

101 34 250 40 100 60

44 66 99 125 190 200

8c05ef480a50f8626abfd044199f41d8



Gambar III.13 Implementasi Algoritma MD5 dan Metode End Of File (EOF)

III.4.4. Algoritma Dari Program

Algoritma dari program merupakan algoritma yang digunakan untuk menjelaskan alur program secara umum. Adapun bentuk algoritma program yang penulis gunakan dalam merancang Keamanan Data Menggunakan Metode EOF dan Algoritma MD5 ini adalah sebagai berikut :

Start

Tampilkan form utama

Pilih tab proses

If tab proses sisip diklik then

Input pesan

Load audio

Input kata sandi dengan algoritma MD5

Sisip pesan dengan metode EOF

If proses ada kesalahan then

Tampil pesan kesalahan

Else Proses Lanjut

End If

Tombol simpan pesan diklik

Elseif tab proses ekstrak diklik then

Load audio

Masukan kata sandi dengan algoritma MD5

Baca pesan dengan metode EOF

If proses ada kesalahan then

Tampil pesan kesalahan

Else Proses Lanjut

End If

Pesan rahasia tampil

Else if tombol about diklik then

Tampilkan form about

Else tombol keluar diklik then

Tutup aplikasi

Endif

End