Implementasi Algoritma AES *(Advanced Encryption Standard)* 256 Bit Dan Kompresi Menggunakan Algoritma *Huffman* Pada Aplikasi *Voice Recorder*

Selvia Rahmawati[1], Ichsan Taufik[2], Gitarja Sandi[3]
[1,2,3]Teknik Informatika UIN Sunan Gunung Djati Bandung
JL. A.H. Nasution 105 Bandung
viarivhi04@gmail.com[1], ichsan@uinsgd.ac.id[2], sandi@uinsgd.ac.id[3]

Abstrak *-- Keamanan data dengan cara enkripsi tidak hanya dapat dilakukan pada file berbentuk teks dan gambar saja. Tetapi, dapat dilakukan pada file suara yaitu dengan melakukan voice recorder pada aplikasi enkripsi yang dibuat. Agar voice dapat dilakukan enkripsi, maka voice tersebut harus diubah menjadi data digital yang akan dikenali oleh komputer. Dengan melakukan enkripsi hasil file biasanya akan semakin besar dibandingkan file sebelum di enkripsi, hal tersebut karena ada penambahan chiper pada setiap data yang dienkripsi. Hasil file yang besar tersebut dapat berpengaruh terhadap media penyimpanan (storage). Maka pada penelitian ini akan mengimplementasikan algoritma AES (Advanced Encryption Standard) 256 bit untuk mengenkripsi dan mendekripsi file voice recorder dan Algoritma Huffman Coding sebagai kompresi file untuk memperkecil hasil file enkripsi agar memori yang dipakai tidak terlalu besar. Hasil yang didapat adalah perbedaan dari ukuran file enkripsi dan kompresi dibandingkan dengan menggunakan enkripsi saja sebesar 40%. Sedangkan waktu proses enkripsi lebih lama dibandingkan dengan dekripsi, rata-rata waktu dari 20 sampel pengujian pada proses enkripsi membutuhkan 18 detik, pada proses dekripsi 5 detik.*

Kata kunci *: Voice, AES 256 bit, Huffman Coding, Android.*

1. Pendahuluan

Masalah keamanan dan kerahasiaan data merupakan salah satu aspek penting yang harus dijaga agar data tersebut tidak tersebarluaskan kepada pihak yang tidak bertanggungjawab. Bukan hanya data yang berbentuk file doc saja yang dapat diamankan, tetapi file berbentuk audio, video, dan image dapat diamankan. *Voice* dalam arti kata bahasa Indonesia adalah suara, yaitu fenomena fisik yang dihasilkan oleh getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu, suara berhubungan erat dengan rasa mendengar [1].

Voice yang masih berbentuk sinyal analog masuk pada *microphone*, didalamnya akan diubah dari sinyal analog menjadi tegangan elektrik dengan bantuan *transducer*, kemudian didalam komputer dibantu dengan *soundcard* untuk perubahan dari tegangan elektrik menjadi data-data digital. Perubahan ini disebut sebagai proses *encoding*, yaitu perubahan sinyal kedalam bentuk yang di optimasi untuk keperluan transmisi data atau penyandian data menjadi data-data digital[2].

Seperti yang pernah dilakukan pada penelitian sebelumnya,yaitu tentang Implementasi Algoritma Blowfish untuk Keamanan Data Suara, dimana penelitian tersebut membahas mengenai pertukaran data suara melalui internet *(client server)*. *File* suara tersebut dienkripsi lalu dikirimkan kepada penerima, walaupun ada pihak yang dapat mengambil file suara tersebut, file suara tidak dapat diputar karena file yang telah terenkripsi. Namun pada penelitian ini masih terdapat kekurangan, diantaranya mengenai pengaplikasian yang masih di desktop, ukuran file

enkripsi yang dihasilkan masih besar sehingga pada saat file dikirim akan menghabiskan waktu yang lama dan penggunaan memori yang dipakai besar, dan algoritma yang digunakan masih dapat ditingkatkan keamanannya [3].

Untuk menjawab kekurangan tersebut, pada penelitian ini dilakukan pengembangan enkripsi *voice* dengan menggunakan algoritma AES (*Advanced Encryption Standard*) 256 bit. Untuk membedakan dengan penelitian sebelumnya, ditambahkan dengan proses kompresi, yang bertujuan untuk mengecilkan hasil file enkripsi atau pada chipernya. Algoritma yang digunakan adalah algoritma Huffman. Yaitu salah satu algoritma kompresi yang apabila telah melakukan dekompresi tidak akan kehilangan data asli, dimana algoritma huffman ini cukup sederhana untuk diterapkan dan dikombinasikan dengan algoritma kriptografi. Huffman merupakan kompresi jenis *Lossless*, teknik kompresi tanpa menimbulkan hilangnya data[4]. Dengan mengurangnya jumlah bit pada data maka akan mempercepat untuk proses penyimpanan ataupun pengiriman data karena file yang dihasilkan lebih kecil.

2. Metodologi Riset

2.1 Algoritma AES (Advanced Encryption Standard)

AES *(Advanced Encryption Standards)* dipublikasikan oleh NIST *(National Institute Of Standard and Technology)* pada tahun 2001 yang digunakan untuk menggantikan algoritma DES yang semakin lama semakin mudah untuk dibobol. AES diperoleh dari hasil kompetisi yang diadakan NIST tahun 1997. Pada tahap pertama 15 peserta dari 21 peserta lolos ke tahap berikutnya berdasarkan penilaian tingkat keamanan, harga, algoritma dan karakteristik implementasi. Sepuluh dari 15 peserta tersebut gugur pada tahap berikutnya karena diangap kurang aman dan efektif.

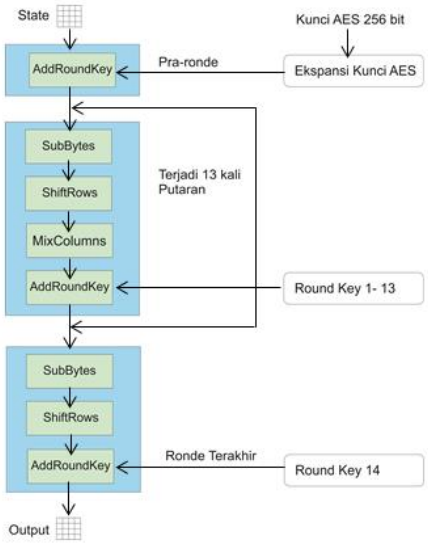
Pada agustus 1999 dipilih lima kandidat dari tahap seleksi akhir, yaitu *MARS* (IBM, Amerika Serikat), *RSA (RSA corp*., Amerika Serikat), *Rijndael* (Belgia), *Serpent* (Israel, Norwegia, Inggris), dan *Twofish* (Counterpane., Amerika Serikat). Pada tahap ini NIST memberikan penilaian pada general *security*, implementasi *software* dan *hardware*, ruang lingkup, implementasi atas serangan, enkripsi dan dekripsi, kemampuan kunci, dan kemampuan lain seperti fleksibilitas dan kepotensial untuk tingkat instruksi pararel. Akhirnya, 2 Oktober 2000 terpilih algoritma *Rijndael* yang dibuat oleh Dr. Vincent Rijment dan Dr. Joan Daemen sebagai pemenang [4].

2.1.1 Struktur Enkripsi Algoritma AES

Proses didalam AES merupakan transformasi terhadap *state*. Sebuah teks asli dalam blok (128 *bit*) terlebih dahulu diorganisir sebagai *states.* Enkripsi AES adalah transformasi terhadap *state* secara berulang dalam beberapa ronde. *States* yang menjadi keluaran ronde k menjadi masukan untuk ronde ke-k +1.

Pada awalnya teks asli direorganisasi sebagai sebuah *state*, kemudian sebelum ronde 1 dimulai blok teks asli dicampur dengan kunci ronde ke-0 (transformasi ini disebut *AddRoundKey*). Setelah itu, ronde ke-1 sampai ronde ke-(Nr-1) dengan Nr adalah jumlah ronde menggunakan 4 jenis transformasi, yaitu *SubBytes, ShiftRows,MixColoumns* dan *AddRoundKey*. Pada ronde terakhir, yaitu ronde ke-Nr dilakukan transformasi serupa dengan ronde lain namun tanpa transformasi *MixColoumns*. Secara garis besar enkripsi AES seperti pada Gambar 1.

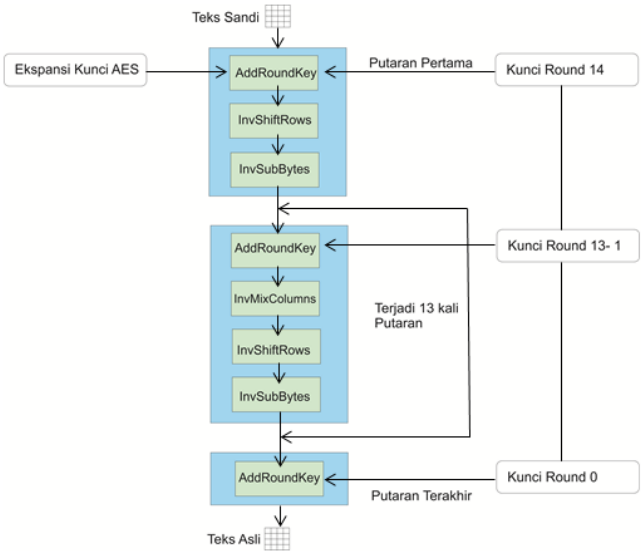
ISBN: 978-602-512-810-3



Gambar 1. Struktur Enkripsi AES

2.1.2 Struktur Dekripsi AES

Secara ringkas algoritma dekripsi merupakan kebalikan algoritma enkripsi AES. Algoritma dekripsi AES menggunakan transformasi *invers* semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap transformasi dasar AES memiliki transformasi invers, yaitu : *InvSubBytes, InvShiftRows, dan InvMixColoumns. AddRoundKey* merupakan transformasi yang bersifat *self-invers* dengan syarat menggunakan kunci yang sama. Algoritma dekripsi AES dapat dilihat pada Gambar 2.



Gambar 2. Struktur Dekripsi AES

2.2 Algoritma Huffman

Huffman coding menggunakan struktur pohon dalam pemrosesannya. Pohon adalah graf tak berarah yang tidak mengandung sirkuit. Didalam struktur pohon dikenal terminologi parent (orang tua) dan child (anak). *Parent* yaitu sebuah simpul yang memiliki lintasan ke simpul lain dengan tingkatan (level) dibawahnya. *Child* yaitu sebuah simpul yang memiliki lintasan ke simpul lain dengan tingkatan (level) diatasnya [2].

ISBN: 978-602-512-810-3

Biasanya sebuah karakter dikodekan dalam kode ASCII (8 bit) atau kode Unicode (16 bit) yang telah memiliki panjang tetap (fixed-length). Berbeda dengan cara penkodean ASCII atau Unicode, pengkodean dengan Huffman Coding menggunakan panjang bit yang bervariasi dalam mengkodekan sebuah karakter. Karakter yang memiliki frekuensi kemunculan lebih besar memiliki panjang bit yang lebih pendek, sebaliknya karakter yang memiliki frekuensi kemunculan yang lebih kecil memiliki panjang bit yang lebih panjang.

Hal ini menyebabkan kode untuk sebuah karakter dengan menggunakan cara pengkodean Huffman tidak tetap seperti dalam ASCII code atau Unicode. Huffman menggunakan kode awalan (prefix code) yang direpresentasikan dalam struktur pohon biner. Cara penyusunan kedalam pohon biner adalah dengan memberikan label '0' untuk cabang kiri dan label '1' untuk cabang kanan. Hal ini dilakukan secara berulang hingga akhirnya terbentuk rangkaian bit yang merupakan kode awalan (prefix code) [2].

3. Analisis dan Hasil

3.1 Analisis Masalah

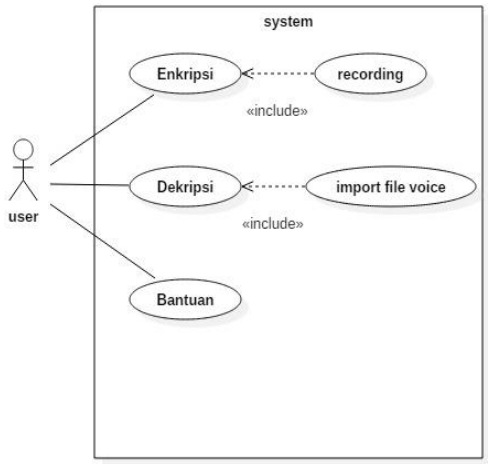
Penelitian ini mengembangankan sistem yang pernah dilakukan dan terdapat perbedaan dengan penelitian sebelumnya yang masih terdapat kekurangan. Penelitian yang berjudul Implementasi Algoritma Blowfish pada Data Suara, penelitian tersebut menjelaskan mengenai pengamanan data pada suara yang dilakukan secara *realtime* dengan menggunakan algoritma blowfish. Hasil dari penelitian tersebut adalah pengamanan data yang berhasil terenkripsi, namun masih terdapat waktu *delay* pada proses pengiriman file dan juga ukuran file yang berukuran besar sehingga dapat menghabiskan memori pada perangkat yang digunakan dan mengganggu komunikasi yang dilakukan secara rahasia tersebut. Dan pengaplikasian yang diterapkan pada desktop. Begitu pun algoritma yang digunakan masih ada kekurangan, karena masih ada algoritma yang lebih aman.

Maka pada penelitian ini dilakukan pengamanan suara melalui *voice recorder* dengan menggunakan algoritma AES 256 bit. Perbedaan AES dan blowfish yaitu terdapat pada chiper blok, blowfish yang mempunyai 64 bit panjang kunci 448 bit sedangkan AES 256 bit panjang kunci 2841 bit. Itu berarti bahwa enkripsi yang dilakukan lebih rumit dan berarti lebih aman karena mempunyai banyak *key*. Dan penelitian ini menambahkan proses kompresi yang bertujuan untuk mengecilkan ukuran file enkripsi, karena diaplikasikan di *mobile* android maka harus mendapatkan ukuran file yang kecil namun tidak akan menghilangkan data asli.

3.2 Perancangan Sistem

a. Usecase Diagram

Use case pada aplikasi ini adalah *user* dapat mengakses secara langsung aplikasi untuk digunakan. Berikut pada Gambar 3 adalah diagram *Use case* dari Aplikasi *Voice Recorder:*

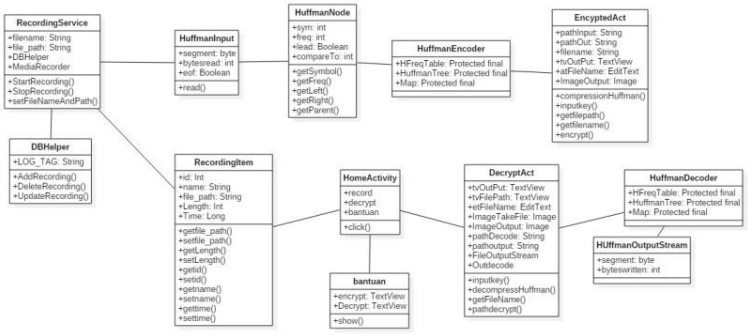


Gambar 3. *Usecase* Diagram Aplikasi *Voice Recorder*

ISBN: 978-602-512-810-3

b. Class Diagram

Class Diagram yang terdapat pada aplikasi *Voice Recorder*, berikut pada Gambar 4*:*

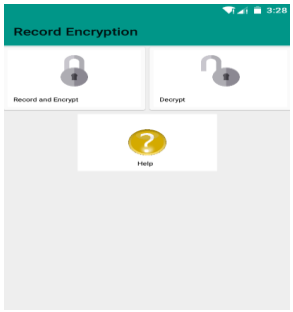


Gambar 4. *Class* Diagram Aplikasi *Voice Recorder*

4. Hasil

a. Menu Utama

Berikut merupakan tampilan antarmuka menu utama aplikasi *Enkripsi Voice Recorder* pada Gambar 5 :

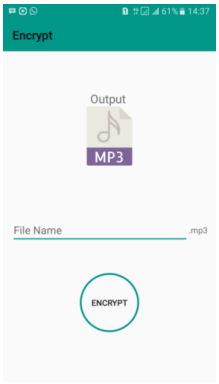


Gambar 5. Tampilan Menu Utama Aplikasi

Gambar 5 merupakan tampilan menu utama yang berisi 3 *button* pilihan, *Record and Encrypt*, *Decrypt*, dan *Help*. *Record and Encrypt* merupakan *form* untuk melakukan *recording* dan enkripsi pada *file record* yang telah dilakukan. *Decrypt* merupakan form untuk melakukan dekripsi pada *file voice*. Dan *Help* merupakan form untuk menjelaskan pemakaian aplikasi enkripsi dan dekripsi *voice recorder*.

b. Halaman Enkripsi

Berikut merupakan tampilan untuk proses enkripsi pada Gambar 6 :



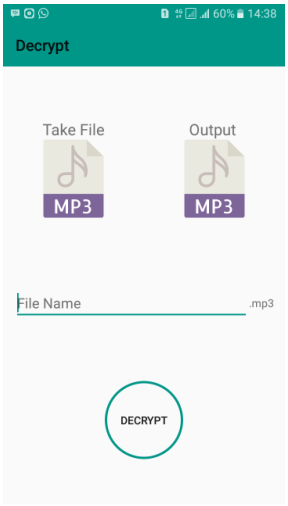
Gambar 6. Tampilan Proses Enkripsi

ISBN: 978-602-512-810-3

Gambar 6 merupakan tampilan untuk melakukan enkripsi pada file. Terdapat *image output* sebagai keterangan tempat file enkripsi disimpan, *file name* untuk nama file enkripsi.

c. Halaman Dekripsi

Berikut merupakan tampilan untuk proses dekripsi pada Gambar 7 :



Gambar 7. Tampilan Proses Dekripsi

Gambar 7 merupakan tampilan untuk proses dekripsi, terdapat dua image yaitu take file untuk memasukkan file enkripsi dan Output untuk tempat penyimpanan file dekripsi, File Name untuk memberi nama file dekripsi.

d. Hasil Pengujian

Tahap pengujian hasil kompresi dan enkripsi pada *voice recorder* menggunakan Algoritma *Huffman Coding* dan Algoritma AES (*Advanced Encryption Standard*) 256 bit. Dapat dilihat pada Tabel 1 berikut ini :

Tabel 1. Hasil Pengujian Kompresi dan Enkripsi

No	Nama File	Key	Waktu (s)	Ukuran (Kb)	
				Asli	Enkripsi
1.	Myrecord1	Selvia0412	11	83,63	82,83
2.	Myrecord1	Fadil04	12	83,63	82,83
3.	Myrecord1	Enkripsi	11	83,63	82,83
4.	Myrecord2	Selvia0412	26	186	186
5.	Myrecord2	Fadil04	25	186	186
6.	Myrecord2	enkripsi	26	186	186
7.	Myrecord3	Selvia0412	14	100	99,75
8.	Myrecord3	Fadil04	13	100	99,75
9.	Myrecord3	enkripsi	14	100	99,75
10.	Myrecord4	Selvia0412	27	188	187

ISBN: 978-602-512-810-3

Tabel 2. Hasil Pengujian Kompresi dan Enkripsi (Lanjutan)

No	Nama File	Key	Waktu (s)	Ukuran (Kb)	
				Asli	Enkripsi
11.	Myrecord4	Fadil04	27	188	187
12.	Myrecord4	enkripsi	27	188	187
13.	Myrecord5	Selvia0412	27	198	197
14.	Myrecord5	Fadil04	26	198	197
15.	Myrecord5	enkripsi	27	198	197
16.	Myrecord6	Selvia0412	13	86,89	86,17
17.	Myrecord6	Fadil04	11	86,89	86,17
18.	Myrecord6	enkripsi	13	86,89	86,17
19.	Myrecord7	Selvia0412	11	79,96	79,12
20.	Myrecord7	Fadil04	12	79,96	79,12

Hasil Pengujian Dekompresi dan Dekripsi

Tahap pengujian hasil dekompresi dan dekripsi pada *voice recorder* menggunakan Algoritma *Huffman Coding* dan Algoritma AES (*Advanced Encryption Standard*) 256 bit. Dapat dilihat pada Tabel 2 berikut ini :

Tabel 3. Hasil Pengujian Dekompresi dan Dekripsi

No	Nama File	Key	Waktu (s)	Ukuran (kb)	
				Enkripsi	Dekripsi
1.	Myrecord1	Selvia0412	3	82,83	83,63
2.	Myrecord1	Fadil04	3	82,83	83,63
3.	Myrecord1	Enkripsi	3	82,83	83,63
4.	Myrecord2	Selvia0412	8	186	186
5.	Myrecord2	Fadil04	8	186	186
6.	Myrecord2	Enkripsi	8	186	186
7.	Myrecord3	Selvia0412	4	99,75	100
8.	Myrecord3	Fadil04	4	99,75	100
9.	Myrecord3	Enkripsi	4	99,75	100
10.	Myrecord4	Selvia0412	7	187	188
11.	Myrecord4	Fadil04	7	187	188
12.	Myrecord4	Enkripsi	7	187	188
13.	Myrecord5	Selvia0412	7	197	198
14.	Myrecord5	Fadil04	7	197	198
15.	Myrecord5	Enkripsi	7	197	198
16.	Myrecord6	Selvia0412	3	86,17	86,89
17.	Myrecord6	Fadil04	3	86,17	86,89
18.	Myrecord6	Enkripsi	3	86,17	86,89
19.	Myrecord7	Selvia0412	3	79,12	79,96
20.	Myrecord7	Fadil04	3	79,12	79,96

ISBN: 978-602-512-810-3

5. Kesimpulan Dan Saran

1. Dengan menggunakan Algoritma AES 256 bit dan Algoritma Huffman menghasilkan *file voice* yang tidak dapat diputar saat telah dilakukan enkripsi, dan menghasilkan file enkripsi lebih kecil dibandingkan dengan file asli.
2. Waktu yang dibutuhkan pada saat enkripsi lebih banyak dibandingkan saat melakukan dekripsi. Rata-rata dari 20 sampel pengujian enkripsi membutuhkan waktu 18 detik, sedangkan dekripsi hanya 5 detik.
3. Dengan diterapkannya teknik kompresi dengan *Algoritma Huffman*, menghasilkan file yang tidak menghabiskan memori yang dipakai. Hasil file enkripsi menjadi lebih kecil dibandingkan dengan file dekripsi atau dengan file asli. Presentase dari file yang terenkripsi dan terkompresi dibandingkan dengan file yang terenkripsi saja sebesar 40%.

Saran

1. Pengembangan aplikasi selanjutnya dapat menggunakan algoritma kriptografi asimetris dengan menggabungkan teknik kompresi lainnya.
2. Pengembangan berikutnya dapat dilakukan dengan mengamankan file data melalui *voice note* dalam aplikasi *chatting* yang berjalan secara *realtime.*
3. *File voice record* sebaiknya tidak tersimpan, tetapi sudah langsung dapat terenkripsi. Jadi, pada saat merekam suara hasil dari rekaman tersebut sudah terenkripsi.

Daftar Pustaka

[1] "Iman Nurzaman Pengertian Suara.". <URL: http://iman-nurzaman.blogspot.co.id/2013/02/pengertian-suara.html>" .

[2] K. Kunci, "Penyandian (Encoding) dan Penguraian Sandi (Decoding) Menggunakan Huffman Coding."

[3] Sutardi, "Implementasi Algoritma Blowfish Untuk Keamanan Data Suara," *J. Ilm. Tek. Mesin*, vol. 6, no. 2, pp. 51–58, 2015.

[4] R. Primartha, "Penerapan Enkripsi dan Dekripsi File menggunakan Algoritma Advanced Encryption Standard (AES)," *J. Res. Comput. Sci. Appl.*, vol. 2, no. 1, pp. 13–18, 2013.

[5] "Kompresi Lossless _ TomatCoklat." . <URL:https://tomatcoklat.wordpress.com/2012/10/11/kompresi-lossless/>.

[6] N. Charibaldi and B. Yuwono, "Aplikasi Enkripsi Pengiriman File Suara Menggunakan Algoritma Blowfish," vol. 2011, no. semnasIF, pp. 201–207, 2011.

[7] S. Tayde, A. Prof, and S. Siledar, "File Encryption , Decryption Using AES Algorithm in Android Phone," vol. 5, no. 5, pp. 550–554, 2015.

[8] R. Warsita, R. A. Setiawan, P. Studi, T. Informatika, and A. Huffman, "Rancang Bangun Aplikasi Kompresi Audio Berbasis Android Menggunakan Algoritma Huffman," no. 14, pp. 1–11.

[9] E. Ophie, "Optimasi Enkripsi Teks Menggunakan AES dengan Algoritma Kompresi Huffman," *Sekol. Tinggi Elektron. dan Inform. Inst. Teknol. Bandung*, 2015.

[10] A. F. Marisman and A. Hidayati, "Pembangungan Aplikasi Pembanding Kriptografi dengan Caesar Cipher dan Advance Ecryption Standard(AES) untuk File Teks," *J. Penelit. Komun. dan Opini Publik*, vol. 19, no. 3, pp. 213–222, 2015.

[11] A. Tiwa, A. S. M. Lumenta, A. M. Rumagit, and A. P. R. Wowor, "Studi Analasis Pengiriman Suara Menggunakan Algoritma Serpent," 2013.

[12] R. Primartha, "Penerapan Enkripsi dan Dekripsi File menggunakan Algoritma Advanced Encryption Standard (AES)," *J. Res. Comput. Sci. Appl.*, vol. 2, no. 1, pp. 13–18, 2013.

[13] "Sadikin,Rifki." . 2012. Kriptografi untuk Keamanan Jaringan. Yogyakarta: Andi." .

[14] L. Firmansah, "Kompresi Data Audio Lossless format FLAC Menjadi Audio Lossy Format MP3 dengan Algoritma Huffman Shift Coding," 2015.

[15] E. M. Harahap, D. Rachmawati, S. Si, M. Kom, and M. Kom, "Impelementasi Kompresi Teks Menggunakan Metode Huffman untuk Menghemat Karakter pada Short Message Service."

ISBN: 978-602-512-810-3

[16] W. Handbook, *Introduction to*. 2005.
[17] S. Multimedia, "Kompresi Audio / Video," *Audio*, 2006.
[18] S. Barber, "The Rational Unified Process : For Dummies What You Should Walk Away With …," *Technology*, pp. 1–25, 2006.
[19] Haviluddin, "Memahami Penggunaan UML (Unified Modelling Language)," *Memahami Pengguna. UML (Unified Model. Lang.*, vol. 6, no. 1, pp. 1–15, 2011.
[20] S. U. Guide, "StarUML 5.0 User Guide."
[21] U. S. Utara, "BAB 2 LANDASAN TEORI 2.1 Teori Umum Android® adalah sebuah kumpulan perangkat lunak untuk perangkat," pp. 7–110, 2013.
[22] Murtiwiyati and G. Lauren, "Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar berbasis Android," *J. Ilm.*, vol. 12, p. 2,3, 2013.
[23] A. Singh, S. Sharma, and S. Singh, "Android Application Development using Android Studio and PHP Framework," *Int. J. Comput. Appl. Recent Trends Futur. Prospect. Eng. Manag. Technol.*, pp. 975–8887, 2016.
[24] A. Rouf, "Pengujian Perangkat Lunak Dengan Menggunakan Metode White Box dan Black Box," *J. Teknol. Inf. HIMSYA-Tech*, vol. 8, no. 1, pp. 1–7, 2012.
[25] A. E. Utami, O. D. Nurhayati, and K. T. Martono, "Aplikasi Penerjemah Bahasa Inggris – Indonesia dengan Optical Character Recognition Berbasis Android," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, pp. 167–177, 2016.

ISBN: 978-602-512-810-3