Implementasi ECDSA pada Audio Digital

Muhammad Nassirudin
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
13511044@std.stei.itb.ac.id

***Abstrak*—Pertukaran informasi digital semakin mudah dilakukan seiring semakin berkembang dan mudah diaksesnya internet dari *gadget* pribadi. Hal tersebut juga berlaku pada berkas audio digital. Saat ini, sudah banyak tersedia aplikasi web, *mobile*, dan desktop yang menyediakan layanan berbagi berkas audio digital. Lebih dari itu, aplikasi-aplikasi tersebut juga menyediakan layanan merekam dan menyimpan berkas audio dari pengguna menggunakan dalam sistem *cloud*. Oleh karena itu, semakin mudah seseorang untuk mendapatkan suara rekaman orang lain. Rekaman suara seseorang merupakan hak cipta orang tersebut dan sudah selayaknya tidak boleh ada orang lain yang mengubah-ubahnya, apalagi memanfaatkannya tanpa bertanggung jawab. Berdasarkan latar belakang inilah, dalam makalah ini diajukan sebuah sistem keamanan menggunakan tanda tangan digital yang tidak mengubah isi dan ukuran berkas audio secara signifikan. Tanda tangan digital tersebut ditujukan untuk dapat melakukan verifikasi kepemilikan audio digital serta mengetahui apakah berkas audio telah mengalami modifikasi atau belum. Algoritma yang digunakan adalah ECDSA yang berbasiskan pada *elliptic curve cryptography* (ECC) serta fungsi *hash* SHA-1. Hasil pengujian memperlihatkan bahwa ECDSA dapat diimplementasikan dengan baik pada berkas audio digital tanpa adanya perubahan pada isi dan ukuran berkas audio.**

Kata kunci—audio digital; tanda tangan digital; ECDSA; ECC; SHA-1

I. PENDAHULUAN

Pertukaran informasi digital semakin mudah dilakukan karena internet semakin mudah diakses. Interaksi masyarakat pun semakin sering terjadi di dunia maya baik melalui media-media sosial maupun aplikasi-aplikasi sosial berbasis internet. Hal tersebut juga berlaku pada pertukaran berkas audio digital. Saat ini, sudah banyak situs dan aplikasi yang menawarkan jasa berbagi audio digital yang berbasis *cloud computing*, yaitu berkas audio digital tidak diproses di mesin pengguna, tetapi melalui internet. Contoh aplikasi yang menawarkan jasa berbagi audio digital adalah SoundCloud dan aplikasi *mobile* Sing!.

Sudah banyak orang yang telah menikmati jasa berbagi audio digital: mulai dari merekam suara sendiri hingga menikmati rekaman suara pengguna lain. Sayangnya, kemudahan akses berkas audio digital tersebut belum disertai keamanan akan keabsahan berkas audio tersebut seperti validasi pemilik audio tersebut dan apakah berkas audio telah dimodifikasi atau belum. Beberapa penyedia jasa berbagi audio digital telah menerapkan kebijakan yang kompleks untuk menjaga berkas-berkas audio penggunanya dari penggunaan yang tidak bertanggung jawab.

Akan tetapi, suatu berkas audio digital masih mungkin tersebar di internet dan dimodifikasi oleh orang yang tidak bertanggung jawab. Isi berkas audio dapat diubah sehingga melanggar hak cipta dari pemilik aslinya. Selain itu, berkas audio juga dapat disangkal oleh pemiliknya karena tidak ada cara untuk melakukan validasi siapa pemilik aslinya.

Oleh karena itu, pada makalah ini ditawarkan sebuah sistem keamanan audio digital yang terdapat di dalam berkas audio itu sendiri. Dengan demikian, tidak peduli apakah berkas audio diunduh atau tersebar di internet, keaslian dan kepemilikan berkas audio masih dapat diverifikasi. Cara yang ditawarkan adalah dengan mengimplementasikan tanda tangan digital dengan ECDSA dalam berkas audio digital.

II. DASAR TEORI

A. *Kriptografi Asimetri*

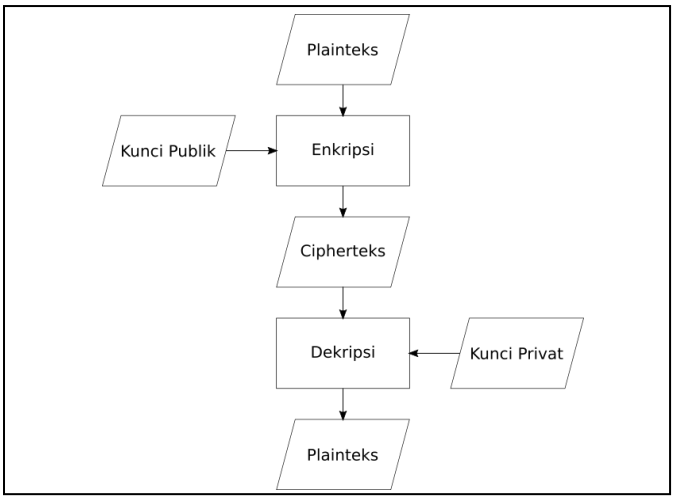
Hingga tahun 1970, kriptografi masih menggunakan kunci yang sama untuk melakukan enkripsi dan dekripsi pesan. Oleh karena itu, kunci yang digunakan bersifat sangat rahasia dan harus dipertukarkan melalui saluran yang sangat aman. Sayangnya, saluran yang aman pada saat itu sangatlah mahal dan lambat. Berdasarkan masalah tersebut, Diffie-Hellman mengajukan suatu cara baru dalam mempertukarkan kunci kriptografi, yaitu dengan menggunakan 2 buah kunci: (1) kunci privat dan (2) kunci publik. Skema pertukaran kunci secara umum dapat dilihat pada Gbr. 1.

Terdapat 2 buah keuntungan menggunakan kriptografi kunci publik, yaitu (1) tidak diperlukan pengiriman kunci rahasia dan (2) tidak perlu mengingat banyak kunci rahasia. Adapun kunci publik tetap perlu saling dipertukarkan. Hanya saja, kunci publik tidak bersifat rahasia sehingga dapat dipertukarkan melalui saluran yang tidak aman. Salah satu algoritma yang menggunakan prinsip kunci publik adalah *Elliptic Curve Cryptography* (ECC).

B. *Elliptic Curve Cryptography*

Pada algoritma ECC, setiap pengguna memegang sebuah kunci privat dan kunci publik. Kunci privat digunakan untuk melakukan dekripsi pesan atau membuat tanda tangan digital, sedangkan kunci publik digunakan untuk melakukan enkripsi

pesan atau verifikasi tanda tangan digital. Keamanan algoritma ECC bergantung pada *Elliptic Curve Discrete Logarithm Problem* (ECDLP) sebagai berikut.



Gbr. 1. Skema pertukaran kunci pada kriptografi asimetri

Sebelum dipaparkan mengenai ECDLP, terlebih dahulu dijelaskan berikut ini beberapa operasi pada *elliptic curve* yang menjadi inti dari ECDLP itu sendiri. *Elliptic curve* merupakan sebuah kurva yang diberikan oleh persamaan (1) dengan $4a^3 + 27b^2 \neq 0$. Di dalam *elliptic curve*, didefinisikan sebuah titik identitas, yaitu titik $O(x,\infty)$.

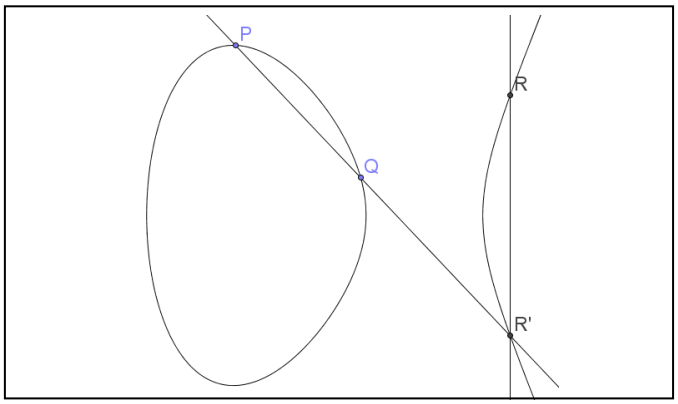
$$y^2 = x^3 + ax + b \quad (1)$$

Terdapat 2 buah operasi dasar pada *elliptic curve*, yaitu (1) penjumlahan titik dengan titik dan (2) perkalian titik dengan bilangan skalar. Misalkan $P(x_p,y_p)$ dan $Q(x_q,y_q)$ dua buah titik pada kurva, hasil penjumlahan $P + Q$ adalah $R(x_r,y_r)$ dengan rumus diberikan pada (2), (3), dan (4). Ilustrasi penjumlahan 2 titik diberikan pada Gbr. 2.

$$x_r = \lambda^2 - x_p - x_q \quad (2)$$

$$y_r = \lambda(x_p - x_r) - y_q \quad (3)$$

$$\lambda = (y_p - y_q)/(x_p - x_q) \quad (4)$$



Gbr. 2. Penjumlahan 2 titik pada *elliptic curve*

Adapun untuk penggandaan titik, yaitu penambahan titik dengan dirinya sendiri seperti dalam $2P = P + P$, persamaan gradien λ yang digunakan diberikan pada (5). Selain itu, didefinisikan juga beberapa kasus khusus yang melibatkan titik identitas seperti yang diberikan oleh (6) dan (7).

$$\lambda = (3x_p^2 + a)/(2y_p) \quad (5)$$

$$P + O = O + P = P \quad (6)$$

$$P + (-P) = O \quad (7)$$

Operasi berikutnya adalah perkalian sebuah titik dengan sebuah bilangan skalar, *k*. Didefinisikan *kP* adalah perkalian titik dengan bilangan skalar. Caranya adalah dengan menjumlahkan titik *P* sebanyak *k* kali. Salah satu sifat istimewa dari *elliptic curve* adalah hasil dari kedua operasi tersebut (penjumlahan dan perkalian titik) merupakan sebuah titik lagi pada *elliptic curve* yang sama.

ECDLP menyatakan bahwa pada persamaan $kP = Q$, mudah dicari *Q* jika diketahui nilai *k* dan *P*. Sebaliknya, sangat sulit menghitung nilai *k* jika hanya diketahui nilai *P* dan *Q*, terutama untuk nilai *k* yang sangat besar. Implementasi perhitungan semua rumus *ellitpic curve* dilakukan dalam modulus *p* (dengan *p* bilangan prima) yang disebut sebagai *Galois Field*. Hal tersebut dilakukan karena algoritma kriptografi hanya melibatkan bilangan-bilangan bulat. Sebagai akibatnya, pembagian dalam rumus diubah menjadi fungsi invers dalam operasi modulus *p*.

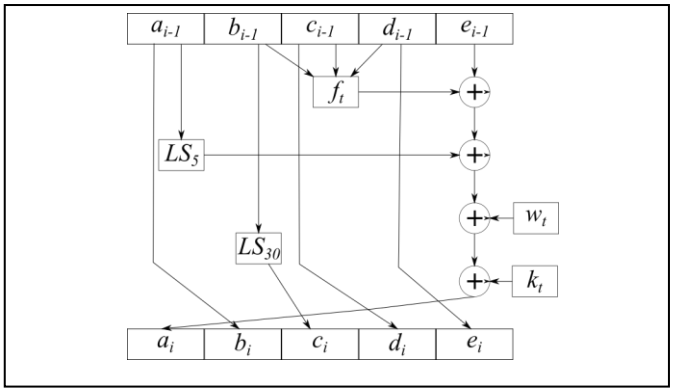
Dalam algoritma ECC, terlebih dahulu disepakati beberapa parameter antara pengirim dengan penerima. Parameter-parameter tersebut adalah (1) parameter persamaan kurva (*a* dan *b*), (2) bilangan prima *p*, dan (3) sebuah titik basis pada kurva *B*. Mengikuti skema pertukaran kunci yang diajukan oleh Diffie-Hellamn, kunci privat dalam ECC adalah sebuah bilangan bulat *k* yang berada pada interval $[0, p - 1]$ sedangkan kunci publik adalah perkalian titik antara kunci privat dengan titik basis yang telah disepakati, *kB*.

C. SHA-1

SHA merupakan singkatan dari *Secure Hash Algorithm*. SHA merupakan fungsi *hash* satu arah yang dikeluarkan oleh NIST. SHA menerima masukan pesan dengan ukuran maksimum 2^{64} bit dan memberikan keluaran berupa *message digest* dengan ukuran 160 bit. Salah satu variasi SHA adalah SHA-1.

Pembuatan message digest pada SHA-1 melalui 4 buah tahap utama: (1) penambahan *padding bit*, (2) penambahan panjang pesan, (3) inisialisasi *buffer*, dan (4) pengolahan pesan dalam ukuran blok 512 bit. Penambahan *padding bit* dimaksudkan untuk membuat ukuran pesan menjadi 448 bit (dalam modulus 512). Penambahan *padding bit* dimulai dengan menambah bit 1 diikuti barisan bit 0. Panjang pesan kemudian ditambahkan di akhir dan dinyatakan dalam ukuran 64 bit. Oleh karena itu, ukuran pesan pasti menjadi kelipatan 512.

Selanjutnya, pesan dibagi ke dalam blok-blok berukuran 512 bit. Setiap blok kemudian diproses dalam proses yang disebut H_{SHA}. Dalam H_{SHA}, setiap blok pesan akan diproses sebanyak 80 putaran. Untuk putaran yang pertama, diinisialisasi 5 buah nilai *buffer* masing-masing berukuran 32 bit. Skema pengolahan blok pesan diberikan pada Gbr. 3.



Gbr. 3. Skema proses H_{SHA}

Dalam putaran ke-*t*, dihitung nilai a_i, b_i, c_i, d_i, dan e_i menggunakan rumus yang diberikan pada (8) sampai dengan (12). Pada putaran pertama, kelima nilai tersebut di-*assign* dengan nilai buffer yang telah diinisialisasi.

$$a_i = k_t + w_t + LS_5(a_{i-1}) + f_t(b_{i-1}, c_{i-1}, d_{i-1}) + e_{i-1} \quad (8)$$

$$b_i = a_{i-1} \quad (9)$$

$$c_i = LS_{30}(b_{i-1}) \quad (10)$$

$$d_i = c_{i-1} \quad (11)$$

$$e_i = d_{i-1} \quad (12)$$

$$w_t = LS_1(w_{t-16} \oplus w_{t-14} \oplus w_{t-8} \oplus w_{t-3}) \quad (13)$$

Fungsi $LS_n(x)$ menyatakan pergeseran bit-bit x ke kiri (siklis) sejauh n bit. Sementara itu, k_t adalah sebuah konstanta pada putaran ke-t yang diberikan pada Tabel I. Nilai w_t berukuran 32 bit dan nilai w_1 sampai dengan w_{16} merupakan pecahan dari blok pesan yang sedang diproses. Untuk putaran ke-17 sampai dengan ke-80, digunakan rumus yang diberikan pada (13). Adapun fungsi $f_t(x,y,z)$ diberikan pada Tabel II.

TABEL I. KONSTANTA NILAI K

Putaran	Nilai *k* (heksadesimal)
$1 \le t \le 20$	5A827999
$21 \le t \le 40$	6ED9EBA1
$41 \le t \le 60$	8F1BBCDC
$61 \le t \le 80$	CA62C1D6

TABEL II. FUNGSI LOGIKA F

Putaran	$f_t(x,y,z)$
$1 \le t \le 20$	$(x \wedge y) \vee (\sim x \wedge z)$
$21 \le t \le 40$	$x \oplus y \oplus z$
$41 \le t \le 60$	$(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$
$61 \le t \le 80$	$x \oplus y \oplus z$

D. Elliptic Curve Digital Signature Algorithm

Elliptic Curve Digital Signature Algorithm (ECDSA) merupakan salah satu variasi algoritma tanda tangan digital yang menggunakan *elliptic curve cryptography*. Terdapat 2 algoritma utama dalam ECDSA, yaitu (1) pembentukan tanda tangan dan (2) verifikasi tanda tangan.

Sebelum dipaparkan algoritma keduanya, terlebih dahulu disepakati beberapa variabel sebagai berikut: m adalah pesan yang ingin ditandatangani, B adalah titik basis kurva, d adalah kunci privat, $Q = dB$ adalah kunci publik, dan n adalah bilangan bulat yang menjadi order dari B (yaitu, $nB = O$). Algoritma pembentukan tanda tangan dilakukan sebagai berikut:

1) dihitung e = SHA1(m);
2) dipilih bilangan acak k dalam interval [1, n – 1];
3) dihitung perkalian titik $kB = (x_1, y_1)$;
4) dihitung $r = x_1 \bmod n$ dan $s = k^{-1}(e + rd) \bmod n$;
5) jika $r = 0$ atau $s = 0$, diulang dari langkah 2;
6) tanda tangan adalah pasangan (r, s).

Sementara itu, algoritma verifikasi tanda tangan digital dilakukan dalam 2 proses: (1) pengecekan kunci publik Q apakah titik valid pada kurva dan (2) verifikasi tanda tangan digital itu sendiri. Berikut ini adalah langkah-langkah pengecekan kunci publik:

1) dicek apakah Q adalah titik identitas;
2) dicek apakah Q terletak pada kurva;
3) dicek apakah hasil perkalian titik $nQ = O$.

Setelah itu, baru dilakukan verifikasi tanda tangan digital dengan tahap sebagai berikut:

1) dicek apakah r dan s keduanya terletak pada interval [1, n – 1];
2) dihitung e = SHA1(m);
3) dihitung $w = s^{-1} \bmod n$;
4) dihitung $u_1 = ew \bmod n$ dan $u_2 = rw \bmod n$;
5) dihitung penjumlahan titik $u_1B + u_2Q = (x_2, y_2)$;
6) jika $r \equiv x_2 \pmod n$, tanda tangan valid; tidak jika sebaliknya.

III. IMPLEMENTASI

Pertukaran kunci publik-privat diimplementasikan menggunakan skema yang diajukan oleh Diffie-Hellman. Parameter-parameter *elliptic curve* dan inisialisasi nilai *buffer* untuk SHA-1 diimplementasikan menggunakan standard P-192 yang dikeluarkan oleh NIST. Nilai parameter tersebut diberikan pada Tabel III.

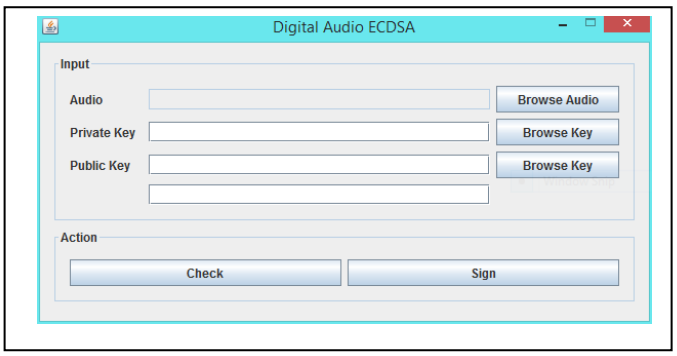
TABEL III. PARAMETER ECC DAN SHA-1

Parameter	Nilai
Bilangan prima, p	6277101735386680763\ 83578942320766641608\ 3908700390324961279
Order, n	62771017353866807638\ 35789423176059013 76\ 7194773182842284081
Parameter ECC, a	-3
Parameter ECC, b	64210519e59c80e7\ 0fa7e9ab72243049\

Parameter	Nilai
(heksadesimal)	feb8deecc146b9b1
Aksis titik basis, B_x (heksadesimal)	188da80eb03090f6\ 7cbf20eb43a1880\ 0f4ff0afd82ff1012
Ordinat titik basis, B_y (heksadesimal)	07192b95ffc8da78\ 631011ed6b24cdd\ 573f977a11e794811
a_0 (heksadesimal)	3045ae6f
b_0 (heksadesimal)	c8422f64
c_0 (heksadesimal)	ed579528
d_0 (heksadesimal)	d38120ea
e_0 (heksadesimal)	e12196d5

Tanda tangan digital dibubuhkan di akhir berkas audio digital dengan maksud agar tidak merusak metadata berkas audio. Tanda yang dipakai sebagai tag awal tanda tangan digital adalah karakter '/'. Format audio digital yang dipakai dalam pengujian adalah format-format yang sering dipakai di dunia maya. Terdapat 5 buah format audio digital yang akan diuji, yaitu (1) midi (.mid), (2) MPEG-1 atau MPEG-2 Audio Layer III atau lebih dikenal dengan MP3 (.mp3), (3) Ogg (.ogg), (4) Waveform Audio File Format atau WAVE atau WAV (.wav), dan (5) Windows Media Audio (.wma).

Implementasi dilakukan di atas bahasa pemrograman Java dan menggunakan Java Swing untuk membangun tampilan grafik antarmukanya. Tampilan antarmuka yang telah selesai dibangun dapat dilihat pada Gbr. 4.



Gbr. 4. Tampilan antarmuka program

IV. Pengujian dan Analisis

Terdapat 2 buah tujuan pengujian yang akan dilakukan. Tujuan yang pertama adalah memastikan apakah sebuah berkas audio digital sudah dimodifikasi atau masih asli. Tujuan yang kedua adalah nirpenyangkalan, yaitu memastikan kepemilikan sebuah berkas audio digital. Berdasarkan kedua tujuan tersebut, dibuat sebuah skenario pengujian sebagai berikut:

1) pembubuhan tanda tangan digital pada berkas audio digital;
2) verifikasi berkas audio digital yang belum dimodifikasi;
3) verifikasi berkas audio digital yang sudah dimodifikasi; dan
4) pengecekan manual apakah terdapat perubahan ketika berkas audio digital dimainkan.

Berkas audio digital yang digunakan berdurasi sekitar 5 menit 30 detik. Berkas audio tersebut kemudia dikonversi ke dalam 5 buah format audio digital sebagaimana yang telah disebutkan dalam Subbab III. Terdapat 3 buah tolak ukur yang diperhatikan dalam pengujian: (1) waktu pembentukan/verifikasi tanda tangan digital dan (2) kemampuan program untuk memeriksa valid tidaknya berkas audio digital.

Hasil pengujian pembuatan tanda tangan digital diberikan pada Tabel IV, sedangkan hasil verifikasi berkas asli dan modifikasi dapat dilihat pada Tabel V. Untuk pengujian verifikasi, audio dimodifikasi dengan mengubah nilai *pitch* audio menggunakan perangkat lunak Audacity®. Namun, berkas audio yang telah dimodifikasi akan kehilangan tanda tangan digital sehingga secara otomatis program akan memberikan pesan eror. Meskipun dengan suatu cara tanda tangan digital tetap ada, program akan memberikan pesan bahwa berkas audio sudah tidak valid lagi.

TABEL IV. Hasil Pengujian Pembuatan Tanda Tangan Digital

Format Audio	Ukuran (MB)	Durasi (ms)	Hasil Dapat Dimainkan?
.mid	0,06	879	Ya
.mp3	5,15	24.260	Ya
.ogg	5,96	26.500	Ya
.wav	56,80	246.261	Ya
.wma	11,10	50.951	Ya

TABEL V. Hasil Pengujian Verifikasi Tanda Tangan Digital

Format Audio	Ukuran (MB)	Durasi (ms)	Valid?
.mid	0,06	1.628	Ya
.mp3	5,15	33.832	Ya
.ogg	5,96	38.005	Ya
.wav	56,80	321.548	Ya
.wma	11,10	79.356	Ya
.mid (modifikasi)	0,06	1.808	Tidak
.mp3 (modifikasi)	5,15	38.224	Tidak
.ogg (modifikasi)	5,96	36.956	Tidak
.wav (modifikasi)	56,80	290.803	Tidak
.wma (modifikasi)	11,10	90.223	Tidak

Berdasarkan hasil yang didapatkan, dapat disimpulkan bahwa ECDSA terimplementasikan dengan baik pada berkas audio digital. Karena panjang tanda tangan digital tidak lebih dari 100 karakter, perubahan ukuran berkas audio tidak menjadi masalah. Selain itu, berkas audio juga tetap dapat dimainkan seperti aslinya.

Hasil yang paling penting adalah kedua tujuan pengujian berhasil dicapai. Yang pertama adalah program mampu mendeteksi berkas audio yang sudah tidak asli lagi, sementara yang kedua adalah program mampu melakukan verifikasi pemilik asli berkas audio (nirpenyangkalan).

Akan tetapi, waktu yang diperlukan program dalam prosesnya masih relatif lama. Lamanya proses tidak hanya terjadi ketika pembuatan tanda tangan digital, tetapi juga ketika melakukan verifikasi. Satu-satunya subproses yang ada di dalam kedua proses tersebut adalah pembentukan *message digest* menggunakan algoritma SHA-1. Oleh karena itu, perlu dilakukan optimasi pada implementasi algoritma SHA-1 atau mencari alternatif algoritma yang lain.

V. KESIMPULAN

Implementasi ECDSA sebagai tanda tangan digital untuk berkas audio digital merupakan pilihan yang baik. Hal tesebut diukur dari segi keamanan, yaitu berhasilnya program melakukan verifikasi pemilik asli dan berkas audio yang dimodifikasi. Selain itu, implementasi yang dilakukan terbukti tidak mengubah berkas audio secara signifikan, baik dari ukuran maupun isinya.

Namun, penggunaan SHA-1 memakan waktu yang cukup lama. Oleh karena itu, pekerjaan berikutnya adalah melakukan optimasi algoritma pembentukan *message digest* untuk mempercepat kinerja program.

REFERENSI

[1] Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *Information Theory, IEEE Transactions on*, *22*(6), 644-654.

[2] K. Aji Nugraha Santosa, "Implementasi Algoritma RSA dan Three-Pass Protocol pada Sistem Pertukaran Pesan Rahasia," unpublished.

[3] P. Gabrielle Wicesawati, "Penggunaan ECC pada Timestamping," unpublished.

[4] http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf. Diakses pada 9 Mei 2015, 05:15WIB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2015

Muhammad Nassirudin
NIM.13511044