ANALISA PERBANDINGAN PERFORMANSI RSA (*RIVEST SHAMIR ADLEMAN*) DAN ECC (*ELLIPTIC CURVE*) PADA PROTOKOL *SECURE SOCKET LAYER* (SSL)

Niko Adianson, Yupianti, Adhadi Kurniawan

Program Studi Sistem Komputer Fakultas Ilmu Komputer Universitas Dehasen Bengkulu
Jl. Meranti Raya No. 32 Kota Bengkulu 38228 Telp. (0736) 22027, 26957 Fax. (0736) 341139

ABSTRACT

Algoritma RSA (Rivest Shamir Adleman) ditemukan tahun 1978 Len Adleman, Ron Rivest dan Adi Shamir dengan ketentuan $C = M^e \bmod n$ sedangkan Algoritma ECC ditemukan tahun 1985 oleh Victor Miller dengan ketentuan $P + O = O + P = P$ untuk setiap $P \in \sum (Fp)$. Dimana dalam algoritma RSA pemaktoran nilai kunci menjadi tingkat kerumitan dan pemecahannya dari pada algoritma ini sedangkan algoritma ECC tingkat kerumitan menentukan titik-titik kurvanya, dimana semakin banyak titik-titik pada kurvanya maka semakin sulit menentukan titik temunya. Algoritma ECC ini banyak diterapkan pada tanda tangan digital. Dari hasil pengujian dilakukan melakukan analisa perbandingan kerumitan dari pada masing-masing algoritma tersebut dan juga tingkat kesulitan dalam pemecahannya. Dan didapatkan hasil algoritma RSA dalam penerapannya membutuhkan sepesifikasi hardware yang tinggi dari pada ECC karena nilai pembentukan kunci RSA lebih besar.

Kata Kunci **:** *RSA, ECC dan SSL*

ABSTRACT

RSA (Rivest Shamir Adleman) was found in 1978 Len Adleman, Ron Rivest, Adi Shamir and with the provisions of the $C = M^e \bmod n$ while the ECC algorithm was found in 1985 by Victor Miller with the provisions of $P + O = O + P = P$ untuk setiap $P \in \sum (Fp)$. Where in the RSA algorithm pemaktoran key value into the complexity and the solution of the algorithm while the ECC algorithm determines the level of complexity of the points the curve, where the more points on the curve, the more difficult to determine the meeting point. ECC algorithms are widely applied to digital signatures. From the results of tests carried out comparative analysis on the complexity of each algorithm and also the level of difficulty in solving them. And the results obtained in the RSA algorithm implementation requires high hardware specs of the ECC for RSA key value greater formation.

Keywords: RSA, ECC dan SSL

I. PENDAHULUAN

Kemajuan ilmu pengetahuan dan teknologi yang sangat cepat telah memberi pengaruh yang baik serta manfaat yang besar bagi manusia dalam berbagai bidang kehidupan. Keunggulan tersebut tidak lepas dari hasil penelitian dan percobaan oleh para ilmuwan dan engineer, yang selalu mencari terobosan dan temuan baru untuk menciptakan sesuatu yang baru bermanfaat dan berguna bagi kehidupan manusia.

Salah satu bidang yang berkembang dengan pesat akibat perkembangan ilmu pengetahuan dan teknologi adalah bidang penyampaian media informasi baik secara lisan mau tulisan. Kini cara berkomunikasi, bertransaksi yang kita telah mengalami banyak perubahan, dari yang semula hanya bisa berkomunikasi atau bertransaksi saat bertemu muka saja, kini manusia sudah dapat berkomunikasi menggunakan teknologi jaringan (*network technology*) yang menghubungkan dua atau lebih komputer Teknologi jaringan tersebut memberikan dampak positif dan dampak negatif.

Dampak positifnya adalah cepat dan mudah kita melakukan komunikasi, transaksi dan lain sebagainya karena sudah tidak terbatas oleh ruang dan waktu lagi. Sedangkan untuk dampak negatifnya dapat dilihat dari semakin meningkatnya tingkat kejahatan di internet, seperti *hacking, cracking, carding, phising, spamming*, dan *defacing*.

Dengan latar belakang inilah maka kemudian disiasati bagaimana cara menjaga kerahasian dan mendeteksi keaslian dari informasi yang dikirim atau diterimanya. Untuk menyiasati hal ini,munculah sebuah ilmu yang disebut dengan kriptografi.

Dari berbagai kelompok kriptografi, terdapat beberapa algoritma kriptografi yang dikenal seperti blowfish, twofish, MARS, IDEA, 3DES, AES, RSA, Diffie-Hellman, MD5, SHA-256, SHA-512, ECC dan lain-lain. Namun diantara kriptografi yang telah disebutkan, terdapat 2 kriptografi yang paling terkenal dan paling sering dibahas saat ini, yaitu RSA (*Rivest Shamir Adleman*) dan ECC (*Elliptic Curve Cryptgraphy*).

Kedua algoritma ini memiliki tingkat keamanan yang cukup tinggi, apalagi ditambah digunakan pada pada protocol SSL (*Secure Sockets Layer*). Yang mana Protokol SSL ini sendiri sudah memberikan

fasilitas enkripsi data, autentikasi server, integritas pesan dan juga pilihan untuk *autentikasi client* seperti pada e-commerce, jaringan telekomunikasi dan lain sebagainya.

II. TINJAUAN PUSTAKA

A) Pengertian Analisa

Menurut Supriyono (2003:89) analisa adalah penelusuran kesempatan atau tantangan atau sumber. Analisa juga melibatkan pemecahan suatu keseluruhan kedalam bagian-bagian untuk mengetahui sifat, fungsi dan saling berhubungan antar bagian tersebut. Analisa sangat diperlukan atau penting karena sifat dari lingkungan sangat dinamis dan berubah dengan cepat.

Analisa adalah menguraikan suatu pokok menjadi beberapa bagian dan menelaah bagian itu sendiri serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan.

Analisa merupakan suatu proses kerja dari rentetan tahapan pekerjaan sebelum riset didokumentasikan melalui tahapan penulisan laporan (Umar, 2002:42). Berdasarkan pengertian analisa di atas, maka dapat disimpulkan bahwa analisa adalah menyelidiki, menguraikan, menelaah suatu permasalahan untuk mengetahui pemahaman yang tepat.

B) Pengertian Performansi atau kinerja

Menurut Mangkunegara (2005 : 67), mengemukakan bahwa "kinerja (prestasi kerja) adalah hasil kerja secara kuantitas dan kualitas yang dicapai oleh seorang pegawai dalam melaksanakan tugasnya sesuai dengan tanggung jawab yang diberikan kepadanya".

Dan kinerja juga merupakan suatu kondisi yang harus diketahui dan di konfirmasikan kepada pihak tertentu untuk mengetahui tingkat pencapian hasil suatu instansi yang dihubungkan dengan visi yang digunakan suatu organisasi atau perusahaan serta mengetahui dampak positif dan negatif dari suatu kegiatan operasional.

Menurut Sutalaksana (1979:42) Perancangan sistem kerja menghasilkan beberapa alternatif sehingga harus dipilih alternatif terbaik. Pemilihan alternatif rancangan sistem kerja ini harus berlandaskan 4 kriteria utama, yaitu: kriteria waktu, kriteria fisik, kriteria psikis,dan kriteria sosiologis. Berdasarkan ke-4 kriteria tersebut suatu sistem kerja dipandang terbaik jika memberikan waktu penyelesaian pekerjaan dengan wajar dan normal serta menggunakan tenaga fisik paling ringan, sehingga memberi dampak psikis dan sosiologis paling rendah.

Berdasarkan pengertian oleh para ahli diatas, maka penulis dapat menarik sebuah kesimpulan bahwa kinerja merupakan kualitas dan kuantitas dari suatu hasil kerja (output) individu maupun kelompok dalam suatu aktifitas tertentu, yang di akibatkan oleh kemampuan dia sendiri atau kemampuan yang di peroleh dari proses penambahan beberapa fitur baik itu sebagian maupun secara keseluruhan sehingga mendapatkan hasil yang sesuai dengan keinginan.

C) Pengertian Kriptografi

Menurut Kromodimoeljo (2010:297) Dalam kriptografi klasik (simetris), jika seseorang mengetahui cara mengenkripsi naskah asli menjadi naskah acak, maka orang tersebut juga mengetahui cara mendekripsi naskah acak yang dihasilkan. Demikian juga jika seseorang mengetahui cara mendekripsi naskah acak, maka orang tersebut juga mengetahui cara mengenkripsi naskah asli untuk menghasilkan naskah acak.

Sekitar pertengahan tahun 1970an, muncul konsep baru dalam kriptografi yaitu kriptografi public key (asimetris). Seseorang yang mengetahui cara mengenkripsi naskah asli belum tentu mengetahui juga cara mendekripsi naskah acak yang dihasilkan. Demikian juga seseorang yang mengetahui cara mendekripsi naskah acak belum tentu mengetahui juga cara mengenkripsi naskah asli untuk menghasilkan naskah acak tersebut. *Enkripsi* dan *dekripsi* dalam *kriptografi public key* menggunakan sepasang kunci yaitu kunci publik (*public key*) dan kunci privat (*private key*). Naskah yang telah dienkripsi menggunakan kunci privat hanya dapat didekripsi menggunakan kunci publik dan naskah yang dapat didekripsi menggunakan kunci publik dapat dipastikan telah dienkripsi menggunakan kunci privat. Sebaliknya, naskah yang telah dienkripsi menggunakan kunci publik hanya dapat didekripsi menggunakan kunci privat. Mekanisme ini memungkinkan berbagai aplikasi, dua yang terpenting diantaranya adalah distribusi kunci sesi dan tanda tangan digital (*digital signature*).

D) Algoritma secara umum

Kata algoritma diambil dari nama ilmuwan muslim dari Al-Khuwārizmi (780-846M), sebagaimana tercantum pada terjemahan karyanya dalam bahasa latin dari abad ke-12 "Algorithmi de numero Indorum". Pada awalnya kata algorisma adalah istilah yang merujuk kepada aturanaturan aritmetis untuk menyelesaikan persoalan dengan menggunakan bilangan numerik arab (sebenarnya dari India, seperti tertulis pada judul di atas). Pada

abad ke-18, istilah ini berkembang menjadi algoritma, yang mencakup semua prosedur atau urutan langkah yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan. Pemecahan sebuah masalah pada hakekatnya adalah menemukan langkah-langkah tertentu yang jika dijalankan efeknya akan memecahkan masalah tersebut (Wahid, 2004:1).

Algoritma adalah urutan langkah-langkah yang dinyatakan dengan jelas dan tidak rancu untuk memecahkan suatu masalah (jika ada pemecahannya) dalam rentang waktu tertentu. Artinya, setiap langkah harus dapat dikerjakan dan mempunyai efek tertentu (Wahid, 2004:2). Arti yang modern untuk algoritma agak mirip kepada kata resep, proses, metode, teknik, prosedur, rutin, kecuali bahwa kata "algoritma" itu sendiri memiliki arti sesuatu yang sedikit berbeda.

Ada 5 komponen utama dalam algoritma yaitu finiteness, definiteness, input, output dan *effectiveness* :

1) Finiteness.

Sebuah algoritma harus selalu berakhir setelah sejumlah langkah berhingga.

2) Definiteness.

Setiap langkah dari sebuah algoritma harus didefinisikan secara tepat, tindakan yang di muat harus dengan teliti dan sudah jelas ditentukan untuk setiap keadaan.

3) Input.

Sebuah algoritma memiliki nol atau lebih masukan, sebagai contoh, banyaknya masukan diberikan di awal sebelum algoritma mulai.

4) Output.

Sebuah algoritma memiliki satu atau lebih keluaran, sebagai contoh, banyaknya keluaran memiliki sebuah hubungan yang ditentukan terhadap masukan.

5) Effectiveness.

Pada umumnya sebuah algoritma juga diharapkan untuk efektif.

E) Pengertian Algoritma RSA (Rivest Shamir Adleman)

Menurut Sentot Kromodimoeljo (2010:298), pada tahun 1978, Len Adleman, Ron Rivest dan Adi Shamir mempublikasikan sistem RSA (lihat [adl78]). Semula sistem ini dipatenkan di Amerika Serikat dan seharusnya masa paten habis tahun 2003, akan tetapi RSA Security melepaskan hak paten setelah 20 September 2000. Sebetulnya sistem serupa telah dilaporkan oleh Clifford Cocks tahun 1973 meskipun informasi mengenai ini baru dipublikasi tahun 1997 karena merupakan hasil riset yang diklasifikasikan sangat rahasia oleh pemerintah Britania Raya (Clifford Cocks bekerja untuk GCHQ, suatu badan di Britania Raya yang fungsinya serupa dengan fungsi NSA di Amerika Serikat), jadi validitas paten patut dipertanyakan karena adanya *prior art*.

Kita jelaskan secara garis besar bagaimana cara kerja RSA. Setiap pengguna memilih, menggunakan *random number generator*, dua bilangan prima yang sangat besar p dan q (masing-masing lebih dari 200 digit)

RSA (Rivest Shamir Adleman) adalah sebuah algoritma pada *enkripsi publik key*. RSA merupakan salah satu metode enkripsi yang paling banyak digunakan. Metode mengambil dua bilangan secara acak yang akan dijadikan kunci sehingga didapat dua kunci yaitu kunci publik dan kunci private. Algortima RSA dijabarkan pada tahun 1977 oleh tiga orang : Ron Rivest, Adi Shamirdan Len Adlemandari Massachusetts Institute of Technology. Huruf RSA itu sendiri berasal dari inisial nama mereka (Rivest—Shamir—Adleman).

Algoritma tersebut dipatenkan oleh Massachusetts Institute of Technology pada tahun 1983di Amerika Serikatsebagai U.S. Patent 4405829. Paten tersebut berlaku hingga 21 September 2000. Semenjak Algoritma RSA dipublikasikan sebagai aplikasi paten, regulasi di sebagian besar negara-negara lain tidak memungkinkan penggunaan paten. Hal ini menyebabkan hasil temuan Clifford Cocks di kenal secara umum, paten di Amerika Serikat tidak dapat mematenkannya (Wikipedia Indonesia).

Proses enkripsi dan dekripsi untuk *plaintext* blok M dan ciphertext blok C dapat digambarkan sebagai berikut :

$$C = M^e \bmod n$$
$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Kedua belah pihak (pengirim dan penerima) harus mengetahui nilai dari n. Pengirim mengetahui nilai e dan hanya menerima yang tahu nilai d . Jadi, dapat disimpulkan bahwa kunci publik dari algoritma ini adalah KU = {e,n} dan kunci pribadinya adalah KR = {d,n}. Untuk penentuan kunci ini juga tidaklah bebas, harus melalui rumus tertentu. Secara lengkap, langkah-langkahyang dilakukan di dalam algoritma RSA adalah sebagai berikut :

1) Pilih 2 bilangan p dan q
2) Cari n = pq
3) Hitung (n) = (p-1)(q-1), sebagai euler's totient function
4) Pilih kunci publik e secara acak sehingga e adalah bilangan prima dari GCD (e, Φ(n)) =1, yang

didapat dengan menggunakan algoritma euclidean.

5) Hitung kunci pribadi dengan kombinasi linier dari e dari Φ(n) berdasarkan algoritma euclidean yang diperoleh dari langkah ke empat sehingga mendapatkan bentuk akhir d = d+Φ(n)
6) Untuk mengenkripsi plaintext M , $0 \leq M \leq n-1$, dilakukan perhitungan $C = M^e \pmod n$

Untuk mengetahui dengan lebih jelas prinsip kerja algoritma RSA ini, kita Penentuan kunci publik dan kunci pribadi :

1) Pilih 2 bilangan prima p = 29 dan q = 31
2) Cari n = pq = 29 x 31 = 899
3) Hitung Φ(n) = (p-1)(q-1) = (29-1)(31-1) = 840
4) GCD (e, Φ(n)) = 1
   840 = (13)64+8
   13 = (8)1+5
   8 = (5)1+3
   5 = (3)1+2
   3 = (2)1+1
   2 = (1)2+0
   GDC(13,840) =1
   Jadi e = 13 telah memenuhi syarat, karena GCD (e, Φ(n))=1
5) Kombinasi linier dari 13 dan 840 adalah :
   1 = λ Φ (n) –e.d
   1 = 3 –2(1)
   1 = 3 – (5-(3)) = (3) 2 –(5)1
   1 = (8 –(5)1) 2 –(5) 1 = (8) 2 –(5) 3
   1 = (8) 2 –(13 –(8)1)3 = (8)5 –(13)3
   1 = (840 –(13)64) 5 –(13) 3 = (840) 5 –(13) 323

6) Plaintext M = 123, $0 \leq m \leq n-1$
   enkripsi plaintext M :
   $C = M^e \pmod n = 123^{13} \bmod 899$
   = 402

7) dekripsi ciphertext C :
   $M = C^d \pmod n = 402^{517} \pmod{899}$
   = 123

Dari hasil yang dperoleh terlihat bahwa pesan yang telah dienkripsi tersebut dapat didekripsi kembali menjadi pesan yang asli.

Dalam memecahkan algoritma RSA ini, seorang *cryptanalyst* dapat melakukan pendekatan sebagai berikut :

1) *Brute force* : mencoba semua kemungkinan kunci pribadi
2) Mencoba mencari faktor p dan q, sehingga dapat dihitung (n). Dengan mengetahui (n), maka dapat ditentukan faktor d.
3) Menentukan (n) secara langsung tanpa menentukan p dan q. Hal ini juga dapat menetukan hasil perhitungan dari faktor d.
4) Menentukan d secara langsung, tanpa menetukan (n).

F) Pengertian Algoritma ECC (Elliptic Curve Cryptgraphy)

Menurut Sentot Kromodimoeljo (2010:315), Sistem *kriptografi public key* yang berbasis pada sukarnya mengkomputasi logaritma diskrit seperti Diffie-Hellman, DSA dan ElGamal bekerja menggunakan suatu multiplicative group GF(q). Suatu *elliptic curve over a finite field* juga memberikan Abelian group yang dapat digunakan untuk mekanisme kriptografi yang serupa dengan sistem berbasis logaritma diskrit. Lebih menarik lagi, *elliptic curve over a finite field* memberikan lebih banyak eksibilitas dibandingkan *finite field* yang terbatas pada GF(p) dan $GF(p^n)$.

Algoritma ECC memiliki kepanjangan dari *Elliptic Curve Cryptpgraphy* ini ditemukan pada tahun 1985 oleh Victor Miller dan Neil Koblitz sebagai mekanisme alternatif untuk implementasi kriptografi asimetris. Algoritma ECC dibuat berdasarkan logaritma diskrit yang lebih menantang pengerjaannya pada kunci yang memiliki panjang yang sama.

Pada Februari 2005 *National Security Agency* (NSA) mempresentasikan strategi dan rekomendasi untuk mengamankan jalur komunikais pemerintahan Amerika dan jaringan komunikasi yang tidak diketahui.Protocol yang dipakai adalah Suite B, yang terdiri dari Elliptic Curve Diffie-Hellman (ECDH), Elliptic Curve MenezesQu-Vanstone (ECMQV) untuk pertukaran dan persetujuan kunci ; Elliptic Curve Digital Signature Algorithm (ECDSA) untuk digital signatures; the Advanced Encryption Standard (AES) untuk *symmetric encryption* and *the Secure Hashing Algorithm* (SHA).

Secara umum, algoritma ECC memiliki kemanan yang kuat, efisien, skalabilitasnya di atas algoritma kriptografi lainnya.

Untuk membentuk elliptic curve cryptoystem (ECC) diperlukan aturan penjumlahan dua titik pada kurva elips E(Fp) yang menghasilkan titik ke tiga pada kurva elips. Aturan ini dapat dijelaskan secara geometris sebagai berikut:

1) P + O = O + P = P untuk setiap P € ∑ (Fp)
2) Jika P = (x,y) € ∑ (Fp) maka (x,y) + (x,-y) = O (titik (x,-y) dinyatakan dengan –P dan disebut negative P. tentunya –P merupakan sebuah titik dalam kurva).

3) Diberikan P = (x1,y1) € ∑ (Fp) dan Q = (x2,y2) € ∑ (Fp), dimana P= -Q. maka P + Q = (x3,y3) diperoleh dengan mengambil garis L yang melewati titik P dan Q atau garis singgung L untuk P=Q.

G) Pengertian SSL (Socket Secure Layer)

Menurut Sentot Kromodimoeljo (2010:335), SSL adalah singkatan dari *Secure Socket Layer*, suatu defacto standard yang dibuat oleh Netscape, perusahaan pembuat web browser terpopuler tahun 1995. Ada dua versi SSL yang digunakan secara umum, yaitu SSL2 dan SSL3. Tahun 1999, standard SSL diambil alih oleh *Internet Engineering Task Force* (IETF) dan namanya diubah menjadi TLS, yang merupakan singkatan dari *Transport Layer Security* (SSL3 menjadi TLS versi 1). Perubahan nama ini mungkin agar nama menjadi lebih netral karena soket adalah istilah Unix. Versi terbaru dari TLS adalah versi 1.2. Karena nama SSL sudah sangat melekat, meskipun nama sudah berganti, disini kita menyebutnya sebagai SSL/TLS.

H) Standart SSL/TLS

Menurut Sentot Kromodimoeljo (2010:335), Dalam sesi menggunakan SSL/TLS, proses yang disebut client berkomunikasi dengan proses yang disebut server. Secara garis besar, sesi antara client dan server diamankan dengan, pertama melakukan handshake, lalu mengenkripsi komunikasi antara client dan server selama sesi berlangsung. Tujuan dari handshake adalah:

1) Server authentication (optional).
2) Menentukan parameter enkripsi.
3) Client authentication (optional).

Bagian *handshake* mungkin merupakan yang terpenting dalam sesi SSL/TLS. Yang jelas handshake merupakan bagian paling rumit dari segi protokol. Secara garis besar, protokol *handshake* adalah sebagai berikut :

1) *Client* mengirim ke *server*: nomor versi SSL/TLS yang digunakan *client*, parameter enkripsi, data yang dibuat secara acak, dan informasi lain yang dibutuhkan oleh *server* untuk berkomunikasi dengan *client*. Jika dibutuhkan, *client* juga meminta *server certificate*.
2) *Server* mengirim ke *client*: nomor versi SSL/TLS yang digunakan *server*, parameter enkripsi, data yang dibuat secara acak, dan informasi lain yang dibutuhkan oleh *client* untuk berkomunikasi dengan *server*. Jika diminta dalam langkah 1, *server* juga mengirim *server certificate*. Jika dibutuhkan, *server* juga meminta client untuk mengirim *client certificate*.
3) Jika *client* meminta *server certificate* dalam langkah 1, *client* melakukan *server authentication*, menggunakan *server certificate* dan informasi lain yang didapat. Jika *authentication* sukses, *server certificate* tidak diminta, atau pengguna mengizinkan, *client* meneruskan ke langkah 4. Jika tidak, sesi dihentikan.
4) Menggunakan data yang telah didapat, *client* membuat suatu *premaster secret* untuk sesi. Tergantung jenis enkripsi yang digunakan, ini dapat dilakukan dengan partisipasi server. Premaster secret dienkripsi menggunakan kunci publik *server* (diambil dari *server certificate*), lalu dikirim ke *server*.
5) Jika *server* meminta *client certificate* pada langkah 2, *client* menandatangan secara digital data yang unik untuk sesi yang diketahui oleh *client* dan *server*. Data berikut *digital signature* dan *client certificate* dikirim oleh *client* ke *server*.
6) Jika *server* meminta *client certificate* pada langkah 2, *server* melakukan *client authentication*. Jika *authentication* diminta dan gagal, maka sesi dihentikan.
7) *Client* dan *server* membuat *master secret* menggunakan *premaster secret*. *Master secret* digunakan oleh *client* dan *server* untuk membuat kunci sesi yang merupakan kunci enkripsi simetris.
8) *Client* memberi tahu *server* bahwa kunci sesi akan digunakan untuk mengenkripsi komunikasi lebih lanjut. *Client* kemudian mengirim pesan yang dienkripsi ke *server* yang mengatakan bahwa ia selesai dengan *handshake*.
9) *Server* memberi tahu client bahwa kunci sesi akan digunakan untuk mengenkripsi komunikasi lebih lanjut. *Server* kemudian mengirim pesan yang dienkripsi ke *client* yang mengatakan bahwa ia selesai dengan *handshake*.
10) *Handshake* selesai.

Server authentication dilakukan dengan memeriksa *server certificate*. Dalam *server certificate* terdapat informasi antara lain :

a) kunci publik server,
b) masa berlaku certificate,
c) domain name untuk server, dan
d) domain name untuk pembuat certificate (biasanya certificate dibuat oleh suatu certificate authority)

I) Penggunaan SSL/TLS

Masih dalam buku Sentot Kromodimoeljo (2010:339), Seperti dikatakan sebelumnya, penggunaan terbesar SSL/TLS adalah untuk *secure* web browsing. Semua web browser yang populer mendukung *secure* web browsing menggunakan SSL/TLS. Biasanya pengguna web browser tidak perlu mengetahui SSL/TLS. Saat web browser akan menampilkan web page dengan prefix https (jadi bukan http), maka web browser secara otomatis akan memulai sesi SSL/TLS. Dalam melakukan *handshake* SSL/TLS, web browser akan melakukan *server authentication* dengan memeriksa *certificate* untuk web server. Web browser biasanya sudah memiliki daftar *certificate authority* yang dapat dipercaya, dan ada web browser yang memperbolehkan pengguna untuk menambah pembuat sertifikat yang dipercaya kedalam daftar. Jika ada masalah dalam *authentication* maka web browser biasanya memberi tahu pengguna dan menanyakan pengguna apakah sesi diteruskan atau tidak. Biasanya ada juga opsi untuk menambah sertifikat yang bermasalah ke daftar pengecualian dimana sertifikat yang ada dalam daftar tidak perlu diperiksa. Jika *handshake* SSL/TLS berhasil maka web page dapat ditampilkan setelah terlebih dahulu didownload dengan proteksi sesi SSL/TLS.

Karena web browser dan web server mendukung SSL/TLS, maka semua sistem informasi *client-server* yang berbasis web dapat menggunakan fasilitas SSL/TLS dengan mudah. Buku ini sangat merekomendasikan pendekatan *client-server* yang berbasis web untuk suatu sistem informasi. Penggunaan sistem informasi akan sangat fleksibel karena pengguna dapat berada dimana saja asalkan ada koneksi TCP/IP dengan server, contohnya:

1) di komputer yang sama dengan server
2) di komputer lain yang terhubung dengen server melalui local area network, atau
3) di lokasi lain, bahkan di negara yang berbeda waktu 12 jam dengan server, asalkan ada koneksi internet ke server.

III. METODOLOGI PENELITIAN

A) Perangkat Keras dan Perangkat Lunak

Adapun instrumen perangkat keras yang digunakan dalam penelitian ini adalah Personal Komputer dengan spesifikasi sebagai berikut :

1) Processor Intel Pentium Core 2 Duo
2) Memory 4 GB
3) HDD 320 GB
4) Monitor 18”
5) Keyboard + Mouse

Perangkat lunak yang penulis gunakan dalam penelitian ini adalah :

1) Sistem Operasi Windows XP SP 3
2) Browser Mozilla dan chrome

B) Metode Pengumpulan Data

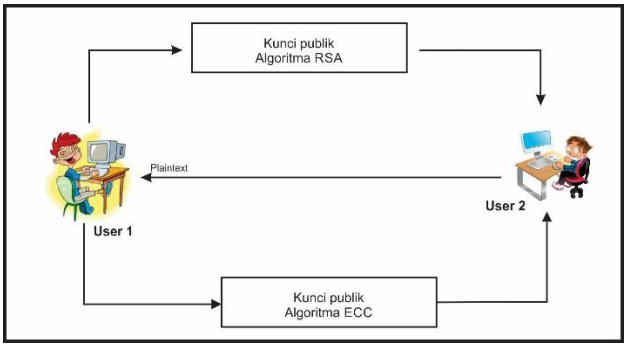
Adapun metode penelitian yang digunakan penulis adalah :

1) Metode eksperimen terhadap keamanan suatu situs yaitu dengan cara mengamati tingkat keamanannya.
2) Metode wawancara yaitu melakukan proses tanya jawab yang dilakukan secara langsung pada beberapa orang pakar mau non pakar IT yang penulis ketahui.
3) Studi Pustaka adalah suatu metode pengumpulan data yang diambil dari perpustakaan atau instansi yang berupa karya ilmiah, jurnal, buku-buku serta dari *internet* yang berhubungan dengan penulisan ini. Tujuan dari studi pustaka ini adalah untuk mendalami dan memperoleh keterangan yang lengkap terhadap obyek yang diteliti.

C) Metode Perancangan Sistem

1) Blok Diagram Global

Blok diagram global dari sistem ini adalah mengetahui tingkat keamanan dengan menerapkan metode argoritma RSA dan algoritma ECC pada protocol SSL.



Gambar 1. Blok Diagram Global

Keterangan

1) User 1 ingin berkomunikasi dengan user 2
2) User 1 mengirim kunci (password) public ke user 2 untuk keamanan berkomunikasi.

Pada proses inilah dilakukan proses algoritma RSA dan algoritma ECC sesuai dengan kesepakatan antara user 1 dengan user 2. Agar komunikasi mereka aman dari proses penyadapan atau dengan kata lain aman dari pencurian pihak-pihak yang tidak bertanggung jawab.

2) Prinsip Kerja Sistem

Prinsip kerja sistem disini bertujuan untuk memfokuskan kerja sistem yang akan di gunakan dari rancangan blok diagram yang akan dibuat dan di implementasikan sesuai dengan rancangan blok diagram dengan pokok kerja sistem. Sistem pengujian yang akan di lakukan melalui proses persiapan alat alat yang di butuhkan, pemasangan infrastruktur jaringan, pengujian konektifitas jaringan. Sampai pada akhir mendapat kesimpulan berupa tingkat keamanan dari RSA dan ECC serta waktu yang dibutuhkan untuk melakukan proses enkripsi.

3) Rencana Kerja

Perancangan rencana kerja tidak terlepas dari blok diagram yang merupakan suatu pernyataan gambar yang ringkas, dari gabungan sebab dan akibat antara masukkan dan keluaran dari suatu sistem.

Gambar 2. Rencana Kerja

Keterangan :

a) Indentifikasi Masalah.

Indentifikasi masalah di sini untuk melakukan indentifikasi masalah masalah yang ada dan sebagai bahan acuan pada tujuan penelitian yang diangkat yaitu tingkat perbandingan algoritma RSA dan ECC.

b) Analisa Kebutuhan.

Sebelum melakukan penelitian lebih lanjut penulis menganalisa kebutuhan-kebutuhan yang penulis perlukan baik itu dari buku-buku, artikel dan lain sebagainya.

c) Pengumpulan Data.

Aktivitas pada tahap ini yaitu mengumpulkan data dengan teknik wawancara, observasi serta dokumentasi baik itu dari internet, buku-buku, Tanya jawab lansung, dan dari eksperimen penulis sendiri, dimana pengumpulan data ini akan menjadi landasan pedoman dalam membantu kegiatan analisa perbandingan performansi RSA dan ECC pada protokol SSL.

d) Analisa Data Untuk Menentukan perbandingan performansi RSA dan ECC pada protokol SSL.

Setelah data didapat dilanjutkan ke tahap selanjutnya menganalisa data tentang perbandingan algoritma RSA dan ECC.

e) Pengujian Menggunakan Komputer Notbook.

Pada tahap kerangka kerja ini penulis melakukan pengujian dengan penerapan metode yang digunakan untuk melihat hasil sebuah sistem yang telah diimplementasikan apakah sudah sesuai dengan kriteria-kriteria yang telah ditentukan dalam melakukan perbandingan performansi RSA dan ECC pada protokol SSL untuk mengetahui tingkat keamanan.

f) Kesimpulan.

Pada akhir pembahasan dilakukan pengambilan kesimpulan sehingga didapatkan informasi tentang hasil perbandingan performansi RSA dan ECC pada protokol SSL.

4) Rencana Pengujian

Pengujian sistem dilakukan menggunakan metode *black box*, yaitu dengan menguji kemampuan sistem berdasarkan spesifikasi yang telah ditentukan. Dimana penulis mengambil atau menjalan file multimedia dari sumber yang sama, maka penulis akan mendapatkan hasil seperti dapat dilihat pada Table 1.

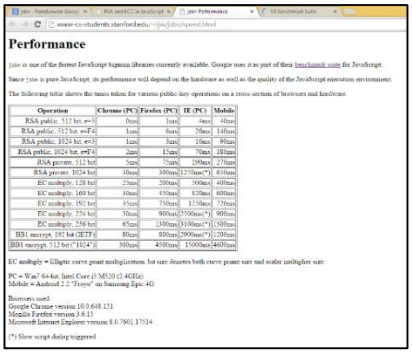
Tabel 1. Rencana Pengujian

Instrumen	RSA	ECC
Tingkat kerumitan algoritma		
Besar ukuran kunci publik		
Kecepatan		

IV. PEMBAHASAN

A) Hasil

Dari serangkaian penelitian yang penulis lakukan dalam membanding algoritma RSA dan ECC, yaitu mulai dari pengumpulan bahan, data, analisa data sampai pada hasil. Adapun hasil pengujian dapat dilihat pada gambar



Gambar 3. Tampilan Hasil Pengujian

Dari gambar diatas dapat dilihat waktu yang dibutuhkan untuk Algoritma RSA lebih Cepat dibandingkan dengan ECC.

B) Algoritma RSA

1) Mekanisme dasar kerja RSA

Tingkat keamanan algoritma penyandian RSA sangat bergantung pada ukuran kunci sandi tersebut (dalam bit), karena makin besar ukuran kunci, maka makin besar juga kemungkinan kombinasi kunci yang bisa dijebol dengan metode mengencek kombinasi satu persatu kunci atau lebih dikenal dengan istilah brute force attack. Jika dibuat suatu sandi RSA dengan panjang 256 bit, maka metode brute force attack akan menjadi tidak ekonomis dan sia-sia dimana para hacker pun tidak mau/sanggup untuk menjebol sandi tersebut.

2) Kekuatan Algoritma RSA

Jika nilai N berjumlah kecil, maka nilai factor p dan q akan mudah diterka oleh parahacker. Maka untuk membuat nilai N sukar untuk dijebol oleh para hacker kita perlu nilai factor p dan q yang besar. Misalkan, dibandingkan kita memilih nilai 5 dan 11, lebih baik kita pilih bilangan prima yang besar, seperti 673 dan 24971, yang akan menghasilkan nilaid = 16805483 dan nilai e = 16779840.

Tetapi jika dihitung dengan suatu perangkat lunak ataupun suatu program yang kita buat yang dapat menghitung faktor-faktor dari suatu nilai. Angka-angka di atas dapat dengan mudahnya didapatkan faktor-faktornya. Sehingga hal ini menyimpulkan bahwa kita membutuhkan nilai p dan q yang sangat besar.

C) Algoritma ECC

1) Mekanisme Dasar Algoritma ECC

Dalam suatu sistem kriptografi diperlukan suatu langkah-langkah yang digunakan guna mempermudah pada saat proses implementasi. Proses penjumlahan titik merupakan inti dari kriptografi kunci publik berdasarkan kurva eliptis. Oleh karena itu, perlu dirumuskan langkah-langkah untuk melakukan proses penjumlahan titik.

2) Contoh Penerapan ECC

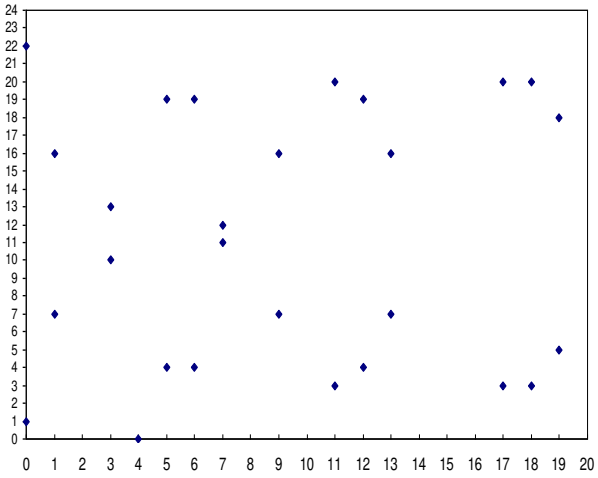
Sebuah kurva elips E pada Z_p didefinisikan dalam persamaan :

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

dimana $a,b \in Z_p$ dan $4a^3 + 27b^2 \neq 0 \pmod p$, dan sebuah titik $\boldsymbol{O}$ yang disebut dengan titik *infinity*. Himpunan $E(Z_p)$ adalah semua titik (x,y), untuk x,y $\in Z_p$, yang memenuhi persamaan (1) pada titik $\boldsymbol{O}$.

Untuk menjelaskan uraian di atas, berikut ini diberikan contoh pencarian himpunan $E(Z_p)$. Diberikan persamaan kurva elips E: $y^2 = x^3 + x + 1$ dengan $p = 23$, yaitu grup $\boldsymbol{Z}_{23}$ (pada persamaan (1) $a = b = 1$). Maka untuk nilai $4a^3 + 27b^2 = 4 + 27 \neq 0$, sehingga E ada dalam kurva elips. Titik-titik dalam $E(Z_{23})$ adalah :

(0,1)	(6,4)	(12,19)
(0,22)	(6,19)	(13,7)
(1,7)	(7,11)	(13,16)
(1,16)	(7,12)	(17,3)
(3,10)	(9,7)	(17,20)
(3,13)	(9,16)	(18,3)
(4,0)	(11,3)	(18,20)
(5,4)	(11,20)	(19,5)
(5,19)	(12,4)	(19,18)



Gambar 4. Sebaran Titik-titik pada Kurva Elips $E(Z_{23})$

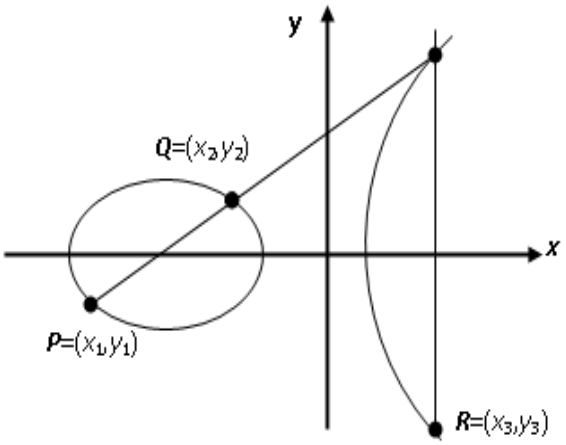
Aturan Penjumlahan Dua Titik Kurva :

$$\lambda = \begin{cases} \left(\dfrac{y_2 - y_1}{x_2 - x_1}\right) \\ \left(\dfrac{3x_1^2 + a}{2y_1}\right) \end{cases}$$

mod p, untuk $P \neq Q$
mod p, untuk $P = Q$
maka diperoleh (x_3,y_3) sebagai berikut :

$$x_3 = (\lambda^2 - x_1 - x_2) \bmod p$$

$$y_3 = (\lambda(x_1 - x_3) - y_1) \mod p$$



Gambar 5**.** Penjumlahan Dua Titik Berbeda

Contoh $P \neq Q$, untuk P=(3,10) dan Q=(9,7), maka $P+Q=(x_3,y_3)$ diperoleh melalui :

$$\lambda = \frac{7-10}{9-3} = 11 \in Z_{23}$$

$x_3 = 11^2 - 3 - 9 = 109 = 17 \pmod{23}$
$y_3 = 11(3 - 17) - 10 = (-164) = 20 \pmod{23}$
Jadi $P+Q$ = (17,20)

3) Penerapan Kunci RSA dan ECC

Untuk melakukan perbandingan algoritma RSA dan ECC. Penulis menggunakan jsbn library, Jsbn adalah library Big Integer untuk bahasa pemrograman javascript, Untuk pengujian kali ini menggunakan aplikasi berbasis web. Pengujian penulis lakukan pada web dengan alamat http://www-cs-students.stanford.edu/~tjw/jsbn/.

Library Jsbn adalah library BigNum pada javascript yang tercepat sampai saat ini. Karena jsbn merupakan pure javascript, kinerjanya tergantung dengan perangkat keras yang digunakan.

D) Hasil Pengujian

Dari serangkaian analisa dan pengujian yang penulis lakukan penulis dapatkan hasil seperti table berikut ini :

Tabel 4.9 Hasil Pengujian

Instrumen	RSA	ECC
Tingkat kerumitan algoritma	Pemaktoran bilangan (kunci)	Penentuan titik kurva
Besar ukuran kunci publik	Lebih Besar	Lebih Kecil

Dan hasil dari jsbn library javascript adalah :

RSA public, 512 bit, e=3	0ms	1ms	4ms	40ms
RSA public, 512 bit, e=F4	1ms	6ms	20ms	140ms
RSA public, 1024 bit, e=3	1ms	3ms	10ms	90ms
RSA public, 1024 bit, e=F4	2ms	15ms	70ms	180ms
RSA private, 512 bit	5ms	75ms	190ms	270ms
RSA private, 1024 bit	30ms	300ms	1250ms(*)	650ms
EC multiply, 128 bit	25ms	200ms	500ms	400ms
EC multiply, 160 bit	30ms	450ms	820ms	600ms
EC multiply, 192 bit	35ms	750ms	1250ms	720ms
EC multiply, 224 bit	50ms	900ms	2500ms(*)	900ms
EC multiply, 256 bit	65ms	1300ms	3100ms(*)	1300ms

V. PENUTUP

A) Kesimpulan

Dari pembahasan skripsi tentang perbandingan Perbandingan Performansi RSA dan ECC dan setelah dilakukan penelitian dan analisa terhadap kedua algoritma tersebut. Maka penulis dapat disimpulkan:

1) Algoritma RSA kekuatannya ditentukan oleh pemaktoran nilai-nilai yang ditentukan. Dimana semakin besar nilai pemaktornya maka semakin lama proses pemecahannya.
2) Algoritma ECC sebaiknya digunakan dalam pengiriman atau persandian data-data digital seperti tanda tangan dan lain sebagainya.
3) Dalam penerapan algoritma RSA lebih membutuhkan spesifikasi hardware yang lebih baik dari pada ECC. Karena jika nilai pemaktorannya besar maka akan membutuhkan kinerja dari pada hardware yang bagus

B) Saran

Dari pengumpulan bahan dan sampai pada hasil pengujian, maka penulis dapat menyarankan:

1) Dalam pengunaan algoritma sebaiknya menentukan dulu angarannya. Karena penerapan algoritma ECC lebih membutuhkan biaya yang lebih tinggi dari pada RSA.
2) Dan dalam penggunaannya hendaknya disesuaikan dengan keperluan kita.

DAFTAR PUSTAKA

Anwar Prabu Mangkunegara. 2005. *Sumber Daya Manusia perusahaan*. Remaja Rosdakarya: Bandung

Anggawisastra, R., Sutalaksana, I. Z, dan Tjakraatmadja, J. H, (1979). *Teknik Tata Cara Kerja*. Departemen Teknik Industri ITB: Bandung

Sentot Kromodimoeljo, 2010, *Teori dan Aplikasi Kriptografi*, SPK IT Consulting: Jakarta

Sofana, Iwan. 2008. *Membangun Jaringan Komputer*. Informatika. Bandung.

Syafrizal, Melwin. 2005. *Pengantar Jaringan Komputer*. ANDI, Yogyakarta.

Tanenbaum, A. S. 2003. *Computer Networks*, Fourth Edition. Prentice-Hall International, Inc.

Wahid Sulaiman. 2004. *Analisis Regresi Menggunakan SPSS*. Yogyakarta: Andi.