PERANCANGAN ALGORITMA KRIPTOGRAFI KUNCI PUBLIK MENGGUNAKAN ARSITEKTUR JARINGAN SARAF TIRUAN

Ibrahim Arief – NIM : 13503038

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13038@students.if.itb.ac.id

Abstrak

Makalah ini membahas mengenai perancangan sistem kriptografi kunci publik baru yang berbasiskan jaringan saraf tiruan untuk proses enkripsinya. Kunci privat pada sistem kriptografi kunci publik ini digunakan sebagai bobot-bobot hubungan antara node-node yang berbeda lapisan dari jaringan saraf tiruan untuk proses enkripsi. Jaringan saraf tiruan yang digunakan hanya terdiri dari neuron-neuron linear untuk memudahkan dekripsi cipherteks yang dihasilkan. Sifat linear jaringan saraf tiruan tersebut memungkinkan proses dekripsi untuk menggunakan metode eliminasi Gauss-Jordan. Metode tersebut biasanya digunakan untuk mencari solusi tertentu dari sebuah sistem persamaan linear. Dengan menganalisa struktur jaringan saraf tiruan, dapat dihasilkan sebuah sistem persamaan linear yang ekivalen dengan jaringan saraf tiruan tersebut. Koefisien dari sistem persamaan linear yang ekivalen tersebut menjadi kunci publik dari sistem kriptografi kunci publik ini. Koefisien tersebut dihasilkan dari perkalian dan penjumlahan enam bobot berbeda pada jaringan saraf tiruan, sehingga keamanan kunci privat tetap terjaga meskipun kunci publik diketahui. Jaringan saraf tiruan yang digunakan terdiri dari tiga node pada lapisan masukan, tiga node pada lapisan tersembunyi, dan tiga node pada lapisan keluaran. Penggunaan node pada lapisan tersembunyi jaringan saraf tiruan dengan jumlah yang lebih sedikit daripada jumlah node pada lapisan masukan ataupun keluaran akan menghasilkan solusi metode eliminasi Gauss-Jordan yang tidak sesuai dengan plainteks, sedangkan penggunaan jumlah node pada lapisan tersembunyi yang lebih banyak daripada jumlah node pada lapisan masukan ataupun lapisan keluaran akan menambah kompleksitas algoritma tanpa ada manfaat yang berarti.

***Kata kunci**:* enkripsi, dekripsi, *gauss-jordan*, jaringan saraf tiruan, algoritma kunci publik.

1. Pendahuluan

Sebuah pesan yang dikirimkan dari pengirim ke penerima melalui suatu media transmisi pesan memiliki resiko untuk disadap oleh pihak yang tidak berkepentingan.

Untuk menjaga kerahasiaan pesan, terdapat dua macam pendekatan yang umum digunakan, yaitu dengan menggunakan jalur komunikasi yang terjamin keamanannya, atau dengan menyandikan pesan dalam bentuk yang hanya dapat dibaca oleh pihak yang berkepentingan.

Pendekatan yang banyak digunakan adalah pendekatan kedua, hal ini dikarenakan menjaga keamanan jalur komunikasi membutuhkan biaya yang cukup besar jika dibandingkan dengan menjaga kerahasiaan pesan dengan penyandian.

Untuk mencapai tujuan menjaga kerahasiaan dari pesan, algoritma penyandian menggunakan kunci rahasia dalam proses enkripsi dan dekripsi dari pesan. Kunci rahasia tersebut dioperasikan dengan pesan dan menghasilkan pesan yang telah dirahasiakan. Pesan yang telah dirahasiakan tersebut dapat dikirimkan melalui media transmisi yang tidak terjamin keamanannya karena tidak dapat dibaca oleh pihak yang tidak berkepentingan selama pihak tersebut tidak memiliki kunci rahasia untuk membaca pesan yang telah menjalani proses enkripsi.

Terdapat dua jenis sistem kriptografi, yaitu sistem kriptografi kunci simetri dan sistem kriptografi kunci publik. Perbedaan antara kedua sistem kriptografi tersebut adalah pemanfaatan kunci yang digunakan dalam proses enkripsi dan dekripsi. Pada kriptografi kunci simetri, kunci yang digunakan untuk proses pengenkripsian adalah sama dengan kunci yang digunakan untuk proses pendekripsian. Hal ini menyebabkan kerahasiaan kunci yang digunakan dalam

kriptografi kunci simetri menjadi sangat penting untuk dijaga. Sedangkan pada kriptografi kunci publik, kunci yang digunakan untuk proses pendekripsian berbeda dengan kunci yang digunakan untuk proses pengenkripsian. Hal ini menyebabkan kunci pengenkripsi tidak harus dijaga kerahasiaannya dan dapat dipublikasikan dengan bebas.

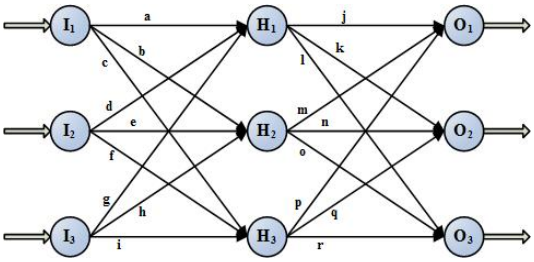
Makalah ini akan membahas mengenai perancangan sebuah sistem kriptografi kunci publik baru yang menggunakan arsitektur jaringan saraf tiruan untuk metode enkripsinya. Sistem kriptografi ini akan memiliki pasangan kunci privat dan kunci publik yang masing-masing dapat digunakan untuk proses enkripsi dan dekripsi. Untuk proses enkripsi, digunakan jaringan saraf tiruan dengan kunci privat sebagai bobot-bobot sinaps antara neuron-neuron jaringan saraf tiruan tersebut. Kunci publik dibangkitkan dari kunci privat dan memenuhi syarat keamanan dimana tidak mungkin mendeduksi kunci privat dari kunci publik tersebut. Keamanan kunci publik menggunakan karakteristik sulitnya memfaktorkan dua buah bilangan.

Untuk keperluan dekripsi, maka digunakan algoritma Gauss-Jordan. Algoritma ini umumnya digunakan untuk menyelesaikan sistem persamaan linear. Untuk sistem kriptografi kunci publik ini, algoritma ini cocok untuk digunakan sebagai algoritma dekripsi dengan menggunakan kunci publik. Hal ini dikarenakan terdapat kesamaan antara perhitungan keluaran jaringan saraf tiruan yang digunakan dengan sistem persamaan linear.

2. Jaringan Saraf Tiruan

2.1 Definisi Jaringan Saraf Tiruan

Jaringan saraf tiruan merupakan suatu paradigma pemrosesan informasi yang meniru cara kerja sistem saraf biologis. Jaringan saraf tiruan terdiri dari sekumpulan unit pemrosesan yang dikenal sebagai unit neuron. Kumpulan unit neuron tersebut saling terhubung dan saling berkerja sama dalam mencari solusi dari sebuah permasalahan.



Gambar 1. Arsitektur jaringan saraf tiruan

Permasalahan yang cocok untuk ditangani oleh jaringan saraf tiruan terbagi menjadi tiga kelompok, yaitu:

1. Permasalahan klasifikasi
   Permasalahan penentuan kelas yang cocok untuk sebuah masukan dengan pola tertentu.

2. Prediksi pola
   Permasalahan pembuatan suatu pola secara lengkap dari sebagian masukan pola dan memprediksi kecenderungan yang mungkin muncul dari data masukan tersebut.

3. Kompresi data
   Permasalahan pengurangan jumlah bit dari suatu blok data untuk disimpan atau dikirim dalam batas-batas kesalahan yang diperkenankan. Sebuah pola tertentu dapat dikenali dari data masukan dan dijadikan basis penmbangkitan data keluaran dengan ukuran lebih kecil. Namun penggunaan jaringan saraf tiruan untuk permasalahan ini memiliki resiko karena pembacaan pola keluaran untuk menghasilkan kembali data masukan rentan akan kesalahan pengklasifikasian data.

2.2 Neuron

Neuron merupakan unit pemroses pada jaringan saraf tiruan. Pada jaringan saraf tiruan, neuron memiliki empat komponen utama, yaitu:

1. Koneksi masukan
   Sumber masukan neuron yang menerima masukan dari neuron-neuron lainnya atau dari luar jaringan. Setiap masukan memiliki bobot yang bersesuaian dengan setiap koneksi antar neuron. Umumnya masukan pada setiap neuron bernilai kontinu dengan rentang nilai antara [0, 1] atau [-1, -1].

2. Fungsi penjumlah
   Fungsi ini menjumlahkan masukan-masukan yang diterima berdasarkan bobot dari masukan tersebut. Masukan yang diterima dikalikan dengan bobotnya lalu hasil seluruh perkalian tersebut dijumlahkan. Fungsi penjumlah dapat didefinisikan melalui persamaan berikut:

$$net = \sum_{i=0}^{n} w_i x_i \qquad (2.1)$$

   yang dalam hal ini net adalah hasil keluaran dari fungsi penjumlahan, w_i menyatakan bobot koneksi masukan ke-i, dan x_i menyatakan masukan dari bobot tersebut.

3. Fungsi aktivasi
   Fungsi aktivasi adalah fungsi yang menentukan keluaran sebuah neuron dari hasil penjumlahan yang didapat melalui persamaan (2.1). Fungsi aktivasi ini dilambangkan dengan notasi σ. Terdapat beberapa jenis fungsi aktivasi, yaitu:
   a. Fungsi linier
      Nilai keluaran neuron sama dengan hasil penjumlahan yang didapat melalui persamaan (2.1), yaitu:

$$\sigma(net) = net \qquad (2.2)$$

      yang dalam hal ini net menyatakan nilai hasil penjumlahan yang didapat melalui persamaan (2.1).

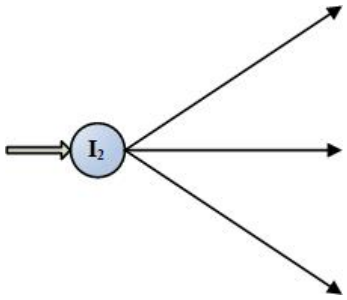
   b. Fungsi ambang (threshold)
      Nilai keluaran neuron dikeluarkan secara diskrit jika nilai hasil penjumlahan dari persamaan (2.1) melebihi nilai ambang tertentu. Dalam penggunaan fungsi ambang, biasanya batasan nilai tersebut adalah nol karena nilai batasan telah ikut diperhitungkan dari adanya bobot bias yang dimiliki unit neuron. Fungsi ambang dapat didefinisikan sebagai berikut:

$$\sigma(net) = \begin{cases} 1 \text{ untuk } net > 0 \\ 0 \text{ untuk } net \leq 0 \end{cases} \qquad (2.3)$$

   c. Fungsi sigmoid
      Nilai keluaran dipetakan dari rentang (-∞,+∞) menjadi bilangan riil dengan rentang antara [0, 1]. Fungsi ini dipilih agar pembelajaran yang menggunakan turunan dari fungsi aktivasi dapat menggunakan fungsi yang kontinu. Fungsi ini dapat didefinisikan sebagai berikut:

$$\sigma(net) = \frac{1}{1 + e^{-net}} \qquad (2.4)$$

4. Koneksi keluaran
   Koneksi keluaran mengirimkan keluaran neuron ke neuron-neuron lainnya atau sebagai keluaran dari jaringan saraf tiruan.



Gambar 2. Sebuah neuron dengan tiga sinaps keluaran

2.3 Arsitektur Jaringan Saraf Tiruan

Jaringan saraf tiruan dapat terdiri dari beberapa lapisan neuron yang semua neuron dalam satu lapisan saling terhubung dengan lapisan tetangganya. Lapisan neuron terbagi menjadi tiga jenis menurut lokasinya dalam jaringan saraf tiruan, yaitu:

1. Lapisan masukan
2. Lapisan tersembunyi
3. Lapisan keluaran

Pada jaringan saraf tiruan hanya terdapat satu lapisan lapisan masukan dan satu lapisan keluaran. Sedangkan untuk lapisan tersembunyi jumlahnya bervariasi sesuai dengan

permasalahan yang dihadapi jaringan saraf tiruan. Namun umumnya jumlah lapisan tersembunyi adalah antara nol sampai dengan tiga lapisan. Hal ini dikarenakan jaringan saraf tiruan dengan tiga lapisan tersembunyi sudah mencukupi untuk menyelesaikan berbagai permasalahan yang mungkin dihadapi.

Jumlah unit neuron pada lapisan masukan bersesuaian dengan jumlah data masukan diskrit dari permasalahan yang dihadapi. Sedangkan jumlah unit neuron pada lapisan keluaran bersesuaian dengan jumlah yang dibutuhkan untuk memodelkan solusi dari permasalahan. Untuk lapisan tersembunyi, jumlah yang diperlukan sangat bervariasi dan biasanya dibutuhkan analisa *heuristik* untuk menentukan jumlah unit yang optimal untuk permasalahan yang dihadapi jaringan saraf tiruan.

2.4 Pembelajaran Jaringan Saraf Tiruan

Jaringan saraf tiruan cocok untuk digunakan dalam permasalahan pengolahan data yang memiliki pola. Dalam menghadapi permasalahan tersebut, jaringan saraf tiruan memiliki kemampuan pembelajaran untuk mengenali dan memprediksi pola data masukan.

Pembelajaran dalam jaringan saraf tiruan dilakukan dengan mengubah bobot-bobot yang dimiliki setiap koneksi antar unit pemroses. Algoritma yang umum digunakan dalam pembelajaran jaringan saraf tiruan adalah algoritma propagasi balik.

Algoritma propagasi balik dapat digunakan untuk menentukan bobot dalam koneksi antar neuron-neuron dalam jaringan saraf tiruan. Algoritma ini secara garis besar memiliki empat tahapan:

1. Pengkalkulasian nilai keluaran dari data masukan.
2. Perbandingkan nilai keluaran yang didapat dengan nilai keluaran yang diharapkan untuk menentukan tingkat kesalahan.
3. Propagasi balik tingkat kesalahan yang didapat dari lapisan neuron keluaran menuju lapisan neuron masukan.
4. Pengubahan bobot koneksi yang dimiliki setiap neuron sesuai dengan tingkat kesalahan masing-masing neuron tersebut.

Algoritma propagasi balik memiliki beberapa parameter yang dapat menentukan tingkat efektivitas pembelajaran jaringan saraf tiruan. Parameter-parameter tersebut adalah:

1. *Maximum epoch*
   Menentukan berapa iterasi pembelajaran yang akan dilakukan pada jaringan saraf tiruan. Semakin besar *maximum epoch* maka tingkat kesalahan jaringan saraf tiruan akan semakin menurun. Namun, penentuan *maximum epoch* yang terlalu besar akan menyebabkan jaringan saraf tiruan terlalu mengikuti pola data pelatihan dan meningkatkan kesalahan yang mungkin terjadi ketika jaringan saraf tiruan diberikan data masukan dari permasalahan sebenarnya. Hal ini dikenal dengan *overfitting*.

2. Laju pembelajaran
   Laju pembelajaran menunjukan seberapa cepat jaringan saraf tiruan akan menyesuaikan diri dengan data pelatihan yang diterimanya.

3. Momentum
   Momentum menunjukan seberapa besar pelatihan pada iterasi pelatihan saat itu hendak dipengaruhi oleh iterasi pelatihan sebelumnya.

3. Sistem Kriptografi Kunci Publik

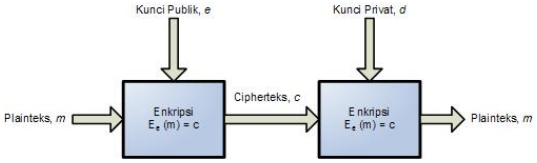
3.1 Latar Belakang Kriptografi Kunci Publik

Kebutuhan akan kriptografi kunci publik muncul karena permasalahan pendistribusian kunci pada kunci privat. Pada kriptografi kunci privat, keamanan kunci yang harus benar-benar dijaga menyebabkan pendistribusian kunci menjadi suatu hal yang sangat krusial bagi terjaganya keamanan data yang dienkripsi. Hal ini menyebabkan kunci privat harus dikirimkan melalui saluran aman yang pada umumnya membutuhkan biaya yang mahal.

Permasalahan tersebut dipecahkan dalam kriptografi kunci publik dengan cara memisahkan kunci yang dibutuhkan untuk proses enkripsi dengan kunci yang dibutuhkan untuk proses dekripsi. Karena itu, kriptografi kunci publik dikenal juga dengan sebutan kriptografi kunci nirsimetri.

Keuntungan dari kriptografi kunci simetri adalah tidak adanya kebutuhan keamanan yang tinggi dalam pendistribusian kunci sebagaimana yang terdapat dalam kriptografi kunci privat. Selain itu, jumlah kunci yang dibutuhkan menjadi dapat

ditekan karena tidak dibutuhkan banyak kunci yang berbeda-beda untuk berkomunikasi dengan banyak pihak. Dengan menggunakan kriptografi kunci publik, cukup dibutuhkan dua kunci saja untuk mendekripsi dan mengenkripsi pesan yang ditujukan untuk pihak yang berbeda-beda tersebut.



Gambar 3. Skema kriptografi nirsimetri

3.2 Karakteristik Kriptografi Kunci Publik

Sistem kriptografi kunci publik memiliki dua karakteristik sebagai berikut:

1. Komputasi untuk enkripsi/dekripsi pesan mudah dilakukan
2. Secara komputasi hampir tidak mungkin (*infeasible*) menurunkan kunci privat, d, bila diketahui kunci publik e, pasangannya.

Untuk memenuhi kedua karakteristik tersebut, terdapat dua permasalahan matematika yang umum digunakan, yaitu:

1. Perkalian vs. Pemfaktoran
   Mengalikan dua buah bilangan prima (axb=n) adalah mudah, namun memfaktorkan n menjadi faktor-faktor primanya lebih sulit.
2. Perpangkatan vs. Logaritmik Diskrit
   Melakukan perpangkatan modulo ($b=a^x$ mod n) adalah mudah, tetapi menemukan x dari $a^x \equiv b \pmod n$ lebih sulit.

Dua permasalahan tersebut sering dijadikan dasar pembangkitan pasangan kunci pada kriptografi kunci publik, yaitu:

1. Pemfaktoran
   Diberikan bilangan bulat n. Faktorkan n menjadi faktor primanya.
   Contoh: 10 = 2 x 5
   60 = 2 x 2 x 3 x 5
   252601 = 41 x 61 x 101
   $2^{13} - 1$ = 3391 x 23279 x 65993 x 1868569 x 1066818132868207
   Semakin besar nilai n, semakin sulit memfaktorkan. Algoritma yang menggunakan prinsip ini contohnya RSA.
2. Logaritma Diskrit
   Temukan x sedemikian sehingga $a^x \equiv b \pmod n$ sulit dihitung.
   Contoh: jika $3^x \equiv 15 \pmod{17}$ maka x=6.
   Semakin besar nilai a, b, dan n, semakin sulit pula memfaktorkan. Algoritma yang menggunakan prinsip ini antara lain adalah ElGamal dan DSA.

3.3 Kelebihan Kriptografi Kunci Publik

Kriptografi kunci publik memiliki kelebihan-kelebihan sebagai berikut jika dibandingkan dengan kriptografi kunci simetri:

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi. Tidak ada kebutuhan mengirim kunci-kunci privat sebagaimana pada sistem kriptografi kunci simetri
2. Pasangan kunci publik/kunci privat tidak perlu dibuah, bahkan dalam periode waktu yang panjang.
3. Dapat digunakan untuk mengamankan pengiriman kunci simetri
4. Beberapa algoritma kunci publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

3.3 Kekurangan Kriptografi Kunci Publik

Kriptografi kunci publik memiliki kekurangan-kekurangan sebagai berikut jika dibandingkan dengan kriptografi kunci simetri:

1. Enkripsi dan dekripsi data pada umumnya lebih lambat daripada sistem simetri, karena enkripsi dan dekripsi menggunakan bilangan yang besar dan melibatkan operasi perpangkatan yang besar.
2. Ukuran cipherteks lebih besar daripada plainteks.

4. Metode Eliminasi Sistem Persamaan Linear Gauss-Jordan

4.1 Deskripsi Metode Gauss-Jordan

Sistem persamaan linear adalah suatu himpunan persamaan dengan koefisien pangkat pada variabel-variabelnya yang tidak melebihi satu. Contoh dari sistem persamaan linear dapat dilihat pada gambar 4.

$$\begin{aligned} x + y + 2z &= 8 \\ -1x - 2y + 3z &= 1 \\ 3x - 7y + 4z &= 10 \end{aligned}$$

Gambar 4. Contoh Sistem Persamaan Linear

Dengan menggunakan metode eliminasi Gauss-Jordan, maka nilai variabel-variabel dalam sebuah sistem persamaan linear dapat dicari. Metode tersebut mengubah sistem persamaan yang telah direpresentasikan dalam bentuk matriks dengan operasi baris dasar untuk menjadikan matriks sistem persamaan linear tersebut menjadi berbentuk eselon baris tereduksi.

Kriteria matriks yang berbentuk eselon baris adalah sebagai berikut:

1. Jika terdapat sebuah baris yang semuanya tidak terdiri dari bilangan nol, maka angka bukan nol pertama dalam baris tersebut adalah satu (satu utama).
2. Jika terdapat baris yang seluruhnya terdiri dari bilangan nol, maka baris tersebut diletakan di bagian bawah matriks.
3. Pada dua baris berurutan yang tidak sepenuhnya terdiri dari bilangan nol, satu utama di baris yang lebih bawah terdapat pada kolom yang lebih kanan dibandingkan dengan baris di atasnya.

Contoh matriks dengan bentuk eselon baris dapat dilihat pada gambar 5.

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ 0 & 1 & -5 & -9 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

Gambar 5. Contoh matriks eselon baris

Metode eliminasi yang menggunakan bentuk matriks eselon baris adalah metode eliminasi Gauss. Sedangkan metode eliminasi Gauss-Jordan menggunakan bentuk matriks eselon baris tereduksi yang merupakan pengolahan lebih lanjut dari bentuk matriks eselon baris.

Kriteria matriks yang berbentuk eselon baris tereduksi adalah sama dengan kriteria matriks eselon baris, dengan tambahan kriteria bahwa setiap kolom yang memiliki satu utama terdiri dari bilangan nol untuk seluruh baris sisanya.

Variasi yang dilakukan pada metode eliminasi Gauss-Jordan untuk menghasilkan matriks eselon baris tereduksi adalah operasi substitusi-balik (*back-substitution*). Pada matriks eselon baris penyederhanaan matriks dilakukan dari atas ke bawah; subsitusi-balik melakukan penyederhanaan lebih lanjut dengan melakukan operasi penyederhanaan-balik dari baris paling bawah ke baris paling atas.

Contoh matriks eselon baris tereduksi dapat dilihat pada gambar 5.

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

Gambar 6. Contoh matriks eselon baris tereduksi

Untuk mencapai bentuk eselon baris tereduksi, metode eliminasi Gauss-Jordan memiliki berbagai macam operasi seperti operasi substitusi dan operasi eliminasi. Operasi substitusi antara mempertukarkan posisi dua baris dalam sistem persamaan linear, sedangkan operasi eliminasi mengurangkan sebuah baris dengan nilai-nilai dari baris lainnya.

5.2 Langkah-langkah Penyelesaian

Sebelum dilakukan penyelesaian sistem persamaan linear dengan menggunakan metode eliminasi Gauss-Jordan, sistem persamaan linear tersebut direprsentasikan dalam bentuk matriks. Reprsentasi bentuk matriks tersebut dengan meletakan koefisien-koefisien persamaan linear pada setiap kolom pada baris yang bersesuaian. Untuk menghasilkan nilai variabel yang diinginkan, maka setidaknya dibutuhkan persamaan linear sejumlah variabel yang ada.

Sebagai contoh, sistem persamaan linear pada gambar 4 dapat direpresentasikan dalam bentuk matriks sebagai berikut:

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ -1 & -2 & 3 & 1 \\ 3 & -7 & 4 & 10 \end{array}\right]$$

Gambar 7. Matriks sistem persamaan linear

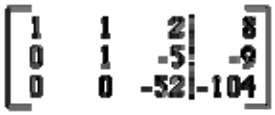
Kemudian, dilakukan operasi eliminasi untuk menghilangkan bilangan bukan nol pada kolom pertama baris-baris di bawah baris pertama.

$$\left[\begin{array}{ccc|c} 1 & 1 & 2 & 8 \\ 0 & -1 & 5 & 9 \\ 0 & -10 & -2 & -14 \end{array}\right]$$

Gambar 8. Matriks sistem persamaan linear

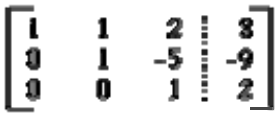
Langkah selanjutnya adalah menghilangkan bilangan bukan nol pada kolom kedua baris-baris

di bawah baris kedua serta menjadikan baris kedua dalam bentuk satu utama.



Gambar 9. Matriks sistem persamaan linear

Langkah selanjutnya adalah menghilangkan bilangan bukan nol pada kolom ketiga baris-baris di bawah baris ketiga serta menjadikan baris tersebut dalam bentuk satu utama. Langkah ini menghasilkan matriks yang sudah dalam bentuk eselon baris namun belum dalam bentuk eselon baris tereduksi.



Gambar 10. Matriks sistem persamaan linear

Langkah selanjutnya adalah menghilangkan bilangan bukan nol pada kolom ketiga baris-baris di atas baris ketiga.

Gambar 11. Matriks sistem persamaan linear

Langkah terakhir adalah menghilangkan bilangan bukan nol pada kolom kedua baris-baris di atas baris kedua. Langkah ini menghasilkan matriks yang sudah dalam bentuk eselon baris tereduksi.

Gambar 12. Matriks sistem persamaan linear

Dengan mengingat bahwa kolom pertama dari matriks sistem persamaan linear tersebut merepresentasikan variabel x, maka dapat disimpulkan bahwa nilai x yang memenuhi sistem persamaan linear adalah 3. Demikian pula untuk kolom kedua dan kolom ketiga. Solusi akhir dari sistem persamaan linear adalah (x=3, y=1, dan z=2).

5. Perancangan Algoritma Kriptografi Kunci Publik Menggunakan Jaringan Saraf Tiruan

Penggunaan jaringan saraf tiruan pada algoritma ini hanya sebagai alat komputasi cipherteks dari plainteks. Sifat jaringan saraf tiruan yang mempertimbangkan seluruh data masukan berdasarkan bobot masing-masing untuk setiap data keluarannya digunakan karena memenuhi prinsip diffusion pada algoritma kriptografi, yaitu penyebaran pengaruh satu bit plainteks atau kunci sebanyak mungkin ke cipherteks. Sifat-sifat jaringan saraf tiruan lainnya seperti pembelajaran dan penggunaan bobot bias tidak digunakan.

Selain itu, node-node yang digunakan untuk jaringan saraf tiruan ini bersifat linear, dimana nilai keluaran suatu node dihitung dengan rataan dari hasil perkalian masing-masing nilai masukan dengan bobot yang bersangkutan (2.1). Perhitungan nilai keluaran tidak melalui fungsi pemetaan seperti sigmoid atau logaritmik.

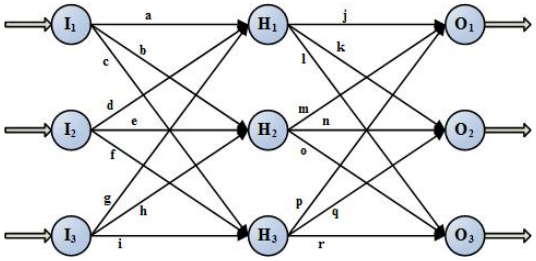
Dengan penggunaan node yang bersifat linear, maka jaringan saraf tiruan yang dihasilkan kehilangan sifat nonlinear yang menjadi salah satu karakteristik jaringan saraf tiruan. Hal ini untuk memudahkan perhitungan yang diperlukan dalam proses dekripsi pesan.

5.1 Arsitektur Jaringan Saraf Tiruan

Arsitektur jaringan saraf tiruan yang digunakan terdiri dari tiga lapisan, yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Masing-masing lapisan terdiri dari tiga unit neuron. Untuk lapisan tersembunyi, pengujian dilakukan dengan menggunakan jumlah unit neuron sebanyak dua dan tiga buah. Ditemukan bahwa dalam penggunaan unit neuron pada lapisan tersembunyi dengan jumlah kurang dari jumlah unit neuron pada lapisan masukan atau lapisan keluaran, maka tidak terjadi pemetaan satu-ke-satu antara plainteks dengan cipherteks. Pengujian dekripsi cipherteks dengan menggunakan jaringan saraf tiruan dengan dua node tersembunyi tersebut menghasilkan solusi plainteks yang berbeda dengan plainteks awal. Hal ini diperkirakan karena adanya informasi yang hilang ketika ketiga nilai masukan dipetakan ke dua nilai tersembunyi. Dengan menggunakan node tersembunyi sebanyak tiga buah, maka tidak ada informasi yang hilang ketika dilakukan pemetaan nilai antara setiap lapisan. Jumlah node yang lebih banyak dari pada jumlah node masukan tidak akan memberikan manfaat tambahan dan akan

memberatkan komputasi yang dibutuhkan ketika proses enkripsi dan dekripsi.

Hubungan antara node-node pada jaringan saraf tiruan adalah hubungan penuh, dimana setiap node pada satu lapisan terhubungkan dengan seluruh node pada lapisan selanjutnya. Bobot bias tidak digunakan dalam arsitektur ini.



Gambar 13. Arsitektur jaringan saraf tiruan pada algoritma

Untuk keperluan penjelasan algoritma, setiap node-node masukan diberikan kode I_n, node tersembunyi diberikan kode H_n, dan node keluaran diberikan kode O_n; dimana n menunjukan urutan node pada lapisan masing-masing. Selain itu, ke-18 hubungan antara node-node pada setiap lapisan diberikan kode huruf 'a' sampai dengan 'r'.

5.2 Penggunaan Kunci Privat

Kunci privat yang digunakan terdiri dari 288 bit yang terbagi menjadi 18 nilai diskrit yang masing-masing terdiri dari 16 bit. Masing-masing nilai diskrit tersebut digunakan sebagai nilai bobot pada jaringan saraf tiruan. Setiap nilai diskrit memiliki pengkodean nilai antara 0 sampai dengan 65535 (2^{16}-1). Nilai tersebut kemudian dipetakan menjadi nilai pecahan antara 0 sampai dengan 1 dengan menggunakan fungsi

$$W = n / 65535 \qquad (5.1)$$

dimana W adalah nilai bobot dan n adalah nilai diskrit.

Peletakan 18 nilai bobot yang telah didapat pada jaringan saraf tiruan dilakukan secara berurutan dari hubungan 'a' sampai dengan hubungan 'r' pada gambar 13.

5.3 Pembangkitan Kunci Publik

Kunci privat yang digunakan terdiri dari 144 bit yang terbagi menjadi 9 nilai diskrit yang masing-masing terdiri dari 16 bit. Sembilan nilai kunci publik tersebut didapatkan dari operasi penambahan dan perkalian dari nilai-nilai diskrit bobot pada kunci privat. Dengan menggunakan contoh pada gambar 13, maka sembilan nilai kunci publik tersebut adalah sebagai berikut:

1. $K_1 = (a*j+b*m+c*p)$ (5.2)
2. $K_2 = (d*j+e*m+f*p)$ (5.3)
3. $K_3 = (g*j+h*m+i*p)$ (5.4)
4. $K_4 = (a*k+b*n+c*q)$ (5.5)
5. $K_5 = (d*k+e*n+f*q)$ (5.6)
6. $K_6 = (g*k+h*n+i*q)$ (5.7)
7. $K_7 = (a*l+b*o+c*r)$ (5.8)
8. $K_8 = (d*l+e*o+f*r)$ (5.9)
9. $K_9 = (g*l+h*o+i*r)$ (5.10)

Setiap bilangan merupakan agregasi dari enam bilangan lainnya pada kunci privat. Kerahasiaan kunci privat tetap terjaga karena sulitnya memfaktorkan masing-masing nilai-nilai pada kunci privat dari hasil perkalian nilai-nilai tersebut.

5.4 Langkah-langkah Enkripsi Algoritma

5.4.1 Persiapan data

Berkas plainteks mengalami pemotongan menjadi blok-blok yang masing-masing blok memiliki ukuran sebesar 3 byte. Jika pada blok terakhir terdapat ruang sisa, maka dilakukan padding sehingga blok tersebut terisi sepenuhnya. Bit padding yang digunakan dibangkitkan secara acak dan informasi panjang byte padding yang digunakan disertakan dalam berkas cipherteks.

Pembangkitan secara acak ini untuk mencegah kripanalis melakukan *known-plaintext attack* jika blok terakhir hanya terdiri dari satu byte plainteks dan dua byte padding.

Setiap byte plainteks yang terdiri 8-bit dipetakan menjadi nilai pecahan antara 0 sampai dengan 1. Pemetaan tersebut menggunakan rumus

$$B' = (2n)^{-1} + B(n^{-1}) \qquad (5.11)$$

dimana B' adalah nilai pecahan plainteks dan B adalah nilai integer plainteks. Nilai n adalah banyaknya pengkodean yang mungkin. Untuk cipher yang hanya menangani berkas alfabetis, nilai ini adalah 26, sedangkan untuk cipher yang menangani berkas biner, nilai ini adalah 256. Algoritma ini menggunakan nilai n sebesar 256.

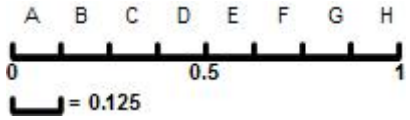
Rumus tersebut menjadikan pengkodean nilai pecahan plainteks menjadi memiliki ruang kesalahan yang cukup besar antara satu

pengkodean dengan pengkodean lainnya. Rentang kesalahan tersebut adalah sebesar :

$$\Delta e = \pm (2n)^{-1} \qquad (5.12)$$

dimana Δe adalah rentang kesalahan dan n adalah banyaknya pengkodean yang mungkin.

Sebagai contoh, untuk n=8, ilustrasinya dapat dilihat pada gambar berikut ini:



Gambar 14. Contoh pengkodean n=8

Dalam contoh tersebut, pengkodean untuk huruf A adalah $(2*8)^{-1} + 0(8^{-1}) = 0.0625$. Pengkodean ini memiliki rentang kesalahan sebesar $\pm$ 0.0625 atau antara ($0 \leq x \leq 0.125$).

Guna dari pengkodean tersebut adalah untuk meminimalisir kemungkinan kesalahan saat proses dekripsi karena terjadi pembulatan saat penyimpanan data cipherteks dan kunci publik. Dari pengujian algoritma didapatkan bahwa tingkat kesalahan rata-rata dari plainteks hasil dekripsi adalah kurang dari 0.01%. Dengan menggunakan pengkodean (5.11) dan n=256, maka tingkat kesalahan maksimal yang diperbolehkan adalah < 0.39% untuk B=255.

Pada pengujian algoritma, pembulatan pada bilangan pecahan dilakukan sampai dengan sembilan angka di belakang koma. Pada sistem penyimpanan berkas yang diusulkan, pembulatan dilakukan sampai dengan lima angka di belakang koma. Namun diharapkan tidak terjadi kesalahan yang dikarenakan perbedaan pembulatan ini. Jika pada pengujian selanjutnya dengan menggunakan pembulatan sampai dengan lima angka di belakang koma ditemukan terdapat kesalahan, maka sistem penyimpanan berkas dapat diubah menjadi menggunakan tiga byte cipherteks untuk mengkodekan satu byte plainteks. Pengkodean seperti ini mengakibatkan pembulatan sampai dengan delapan angka di belakang koma.

5.4.2 Proses Enkripsi

Setelah jaringan saraf tiruan dengan arsitektur seperti pada gambar 13 diinisialisasi dengan nilai-nilai kunci privat pada bobotnya, byte pada setiap blok dimasukan ke jaringan saraf tiruan tersebut. Hasil keluaran disimpan lagi dalam blok yang berukuran tiga byte. Masing-masing byte dalam blok tersebut disimpan dengan menggunakan pengkodean bilangan pecahan dengan ukuran dua byte. Berkas cipherteks akhir memiliki ukuran dua kali ukuran berkas plainteks awal.

Pengkodean satu byte plainteks dalam dua byte cipherteks adalah untuk mereduksi terjadinya kesalahan pembulatan saat penyimpanan berkas cipherteks.

5.5 Langkah-langkah Dekripsi Algoritma

Berkas cipherteks dibaca setiap enam byte dan disimpan dalam satu blok yang terdiri dari tiga nilai masukan. Dua byte cipherteks dikonversikan menjadi satu nilai masukan dengan menggunakan persamaan yang sama dengan persamaan (5.1).

Kemudian dibentuk sebuah matriks berukuran 3x4 dengan nilai-nilai sebagai berikut:

$$\begin{bmatrix} K_1 & K_2 & K_3 & O_1 \\ K_4 & K_5 & K_6 & O_2 \\ K_7 & K_8 & K_9 & O_3 \end{bmatrix}$$

Gambar 15. Matriks dekripsi

dimana K_n adalah nilai kunci publik ke-n yang bersesuaian dengan rumus (5.2) sampai dengan (5.10) dan O_n adalah byte cipherteks ke-n dari blok cipherteks.

Matriks tersebut lalu diselesaikan dengan menggunakan metode eliminasi Gauss-Jordan untuk sistem persamaan linear. Setelah eliminasi Gauss-Jordan menghasilkan nilai-nilai solusi, nilai-nilai tersebut dikalikan dengan 9 untuk mengkoreksi rataan yang dilakukan setiap lapisan jaringan saraf tiruan pada proses enkripsi. Nilai 9 didapat dari jumlah node masukan (3) yang dikalikan dengan jumlah node tersembunyi (3).

5.6 Pembuktian Proses Dekripsi Algoritma

Dari rumus (2.1), didapatkan bahwa nilai yang terdapat pada node tersembunyi adalah sebagai berikut:

$$3 * N(H_n) = \sum_{m=0}^{k} I_m * w_{mn} \qquad (5.13)$$

dimana $N(H_n)$ adalah nilai pada node tersembunyi, I_m adalah nilai pada node masukan ke-m, W_{mn} adalah nilai bobot pada hubungan antara node m dan n, dan k adalah jumlah node pada satu lapisan.

Kemudian untuk node keluaran, dari rumus (2.1) dapat dihasilkan persamaan nilai node keluaran dari nilai-nilai node tersembunyi.

$$3 * N(O_n) = \sum_{m=0}^{k} H_m * w_{mn} \qquad (5.14)$$

dimana $N(O_n)$ adalah nilai pada node keluaran, H_m adalah nilai pada node tersembunyi ke-m, W_{mn} adalah nilai bobot pada hubungan antara node m dan n, dan k adalah jumlah node pada satu lapisan.

Dari rumus (5.1) dan (5.2), dapat dirumuskan nilai keluaran jaringan saraf tiruan terhadap nilai masukannya.

$$N(O_n) = \sum_{i=1}^{k} \sum_{j=1}^{k} W_{(Hi,On)} * I_j * W_{(Ij,Hi)} \qquad (5.15)$$

dimana N(On) adalah nilai untuk node keluaran ke-n, $W_{(Hi,On)}$ adalah nilai bobot untuk hubungan antara node tersembunyi ke-i dan node keluaran ke-n, I_j adalah nilai untuk node masukan ke-j, dan $W_{(Ij,Hi)}$ adalah nilai bobot untuk hubungan antara node masukan ke-j dan node tersembunyi ke-i.

Dengan mengambil contoh node pada gambar 13, maka rumus (5.15) dapat dinyatakan sebagai berikut:

$$N(O_1) = I_1 * (a*j + b*m + c*p) + I_2 * (d*j + e*m + f*p) + I_3 * (g*j + h*m + i*p) \qquad (5.16)$$

$$N(O_2) = I_1 * (a*k + b*n + c*q) + I_2 * (d*k + e*n + f*q) + I_3 * (g*k + h*n + i*q) \qquad (5.17)$$

$$N(O_3) = I_1 * (a*l + b*o + c*r) + I_2 * (d*l + e*o + f*r) + I_3 * (g*l + h*o + i*r) \qquad (5.18)$$

Dengan mengambil contoh persamaan (5.14), perhatikan bahwa jika nilai (a*j + b*m + c*p) dinyatakan sebagai x, (d*j + e*m + f*p) dinyatakan sebagai y, dan (g*j + h*m + i*p) dinyatakan sebagai z, maka persamaan tersebut menjadi sebuah persamaan linear sebagai berikut:

$$N(O_1) = x * I_1 + y * I_2 + z * I_3 \qquad (5.18)$$

Bentuk persamaan linear tersebut dapat diselesaikan dengan menggunakan metode eliminasi Gauss-Jordan jika diketahui nilai-nilai x, y, dan z serta nilai $N(O_1)$. Selain itu, metode eliminasi Gauss-Jordan membutuhkan jumlah persamaan dalam sistem persamaan linear sejumlah variabel yang terdapat dalam persamaan linear tersebut, yang dalam hal ini adalah tiga persamaan.

Dengan menjadikan nilai-nilai x, y, dan z sebagai kunci publik, serta menggunakan persamaan (5.17) dan (5.18) dalam bentuk yang sama dengan persamaan (5.18), maka kedua prasyarat tersebut dapat dipenuhi. Penyelesaian metode eliminasi Gauss-Jordan akan menghasilkan nilai-nilai I_n dari nilai-nilai O_n yang diberikan. Dengan demikian, plainteks asal (I) dapat didekripsi dari cipherteks (O) dengan menggunakan kunci publik.

7. Pengujian Algoritma

Dalam pengujian algoritma, tidak dilakukan dengan cara menguji terhadap *string* karakter tertentu namun terhadap satu set bilangan acak. Hasil dekripsi diperbandingkan dengan bilangan acak tersebut untuk melihat tingkat kesalahan enkripsi-dekripsi dari algoritma.

Pengujian dilakukan dengan membangkitkan kunci privat secara acak. Kunci privat tersebut kemudian dipecah menjadi 18 nilai terpisah. Seluruh pengujian kemudian dilakukan dengan menggunakan pasangan kunci privat dan kunci publik tersebut. Dengan menggunakan contoh seperti pada gambar 13, nilai kunci privat dapat dilihat pada tabel berikut:

Tabel 1. Nilai Kunci Privat

Kode Bobot	Nilai
a	0.076854366
b	0.788522458
c	0.184339143
d	0.413002162

Tabel 1. Nilai Kunci Privat (lanjutan)

Kode Bobot	Nilai
f	0.064815178
g	0.455414418
h	0.141559532
i	0.167656667
j	0.113963039
k	0.670971810
l	0.642297996
m	0.224819694
n	0.602943802
o	0.523404662
p	0.388865477
q	0.492418767
r	0.535581597

Dari nilai kunci privat tersebut, dihasilkan sembilan nilai kunci publik yang digunakan:

Tabel 2. Nilai Kunci Publik

Kunci Publik	Nilai
K_1	0.257717
K_2	0.129308
K_3	0.148922
K_4	0.617774
K_5	0.461996
K_6	0.473480
K_7	0.560808
K_8	0.432772
K_9	0.456399

Pengujian dilakukan dengan menggunakan data plainteks sebagai berikut:

Tabel 3. Nilai Data Plainteks

Data Plainteks	Nilai
I_1	0.712039
I_2	0.748759
I_3	0.475233

Dengan menggunakan kunci privat dari tabel 1, maka dihasilkan data cipherteks sebagai berikut:

Tabel 3. Nilai Data Cipherteks

Data Cipherteks	Nilai
O_1	0.039011
O_2	0.112313
O_3	0.104473

Nilai-nilai tersebut kemudian dimasukan ke dalam matriks dekripsi sesuai dengan pada gambar 15. Langkah-langkah dekripsi selanjutnya menggunakan langkah-langkah pada metode eliminasi Gauss-Jordan.

$$\begin{bmatrix} 0.257717 & 0.129308 & 0.148922 & 0.039011 \\ 0.617774 & 0.461996 & 0.473480 & 0.112313 \\ 0.560808 & 0.432772 & 0.456399 & 0.104473 \end{bmatrix}$$

Gambar 15. Matriks dekripsi pengujian

Langkah pertama pada metode eliminasi Gauss-Jordan adalah mengubah nilai pada baris kedua kolom pertama menjadi nol dengan melakukan eliminasi.

$$\begin{bmatrix} 0.257717 & 0.129308 & 0.148922 & 0.039011 \\ 0 & 0.152031 & 0.116499 & 0.018799 \\ 0.560808 & 0.432772 & 0.456399 & 0.104473 \end{bmatrix}$$

Gambar 16. Langkah Eliminasi Gauss-Jordan

Langkah selanjutnya adalah mengubah nilai pada baris ketiga kolom pertama menjadi nol dengan melakukan eliminasi.

$$\begin{bmatrix} 0.257717 & 0.129308 & 0.148922 & 0.039011 \\ 0 & 0.152031 & 0.116499 & 0.018799 \\ 0 & 0.151389 & 0.132335 & 0.019582 \end{bmatrix}$$

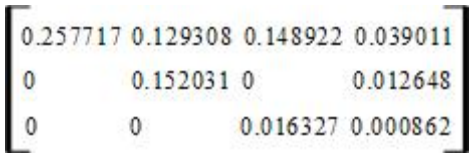
Gambar 17. Langkah Eliminasi Gauss-Jordan

Langkah selanjutnya adalah mengubah nilai pada baris ketiga kolom kedua menjadi nol dengan melakukan eliminasi.

$$\begin{bmatrix} 0.257717 & 0.129308 & 0.148922 & 0.039011 \\ 0 & 0.152031 & 0.116499 & 0.018799 \\ 0 & 0 & 0.016327 & 0.000862 \end{bmatrix}$$

Gambar 18. Langkah Eliminasi Gauss-Jordan

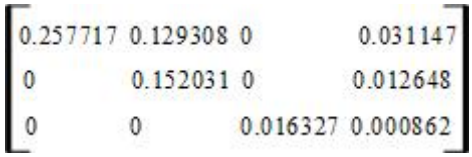
Langkah selanjutnya adalah mengubah nilai pada baris kedua kolom ketiga menjadi nol dengan melakukan eliminasi.



$$\begin{bmatrix} 0.257717 & 0.129308 & 0.148922 & 0.039011 \\ 0 & 0.152031 & 0 & 0.012648 \\ 0 & 0 & 0.016327 & 0.000862 \end{bmatrix}$$

Gambar 19. Langkah Eliminasi Gauss-Jordan

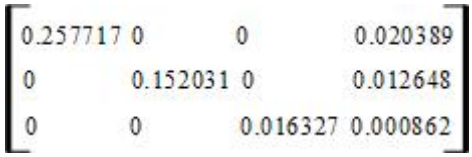
Langkah selanjutnya adalah mengubah nilai pada baris pertama kolom ketiga menjadi nol dengan melakukan eliminasi.



$$\begin{bmatrix} 0.257717 & 0.129308 & 0 & 0.031147 \\ 0 & 0.152031 & 0 & 0.012648 \\ 0 & 0 & 0.016327 & 0.000862 \end{bmatrix}$$

Gambar 20. Langkah Eliminasi Gauss-Jordan

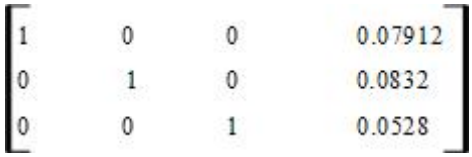
Langkah selanjutnya adalah mengubah nilai pada baris pertama kolom kedua menjadi nol dengan melakukan eliminasi.



$$\begin{bmatrix} 0.257717 & 0 & 0 & 0.020389 \\ 0 & 0.152031 & 0 & 0.012648 \\ 0 & 0 & 0.016327 & 0.000862 \end{bmatrix}$$

Gambar 21. Langkah Eliminasi Gauss-Jordan

Langkah terakhir adalah menormalisasikan nilai pada diagonal matriks menjadi 1 untuk mendapatkan nilai hasil dekripsi.



$$\begin{bmatrix} 1 & 0 & 0 & 0.07912 \\ 0 & 1 & 0 & 0.0832 \\ 0 & 0 & 1 & 0.0528 \end{bmatrix}$$

Gambar 22. Hasil Akhir Eliminasi Gauss-Jordan

Dari metode eliminasi Gauss-Jordan, dihasilkan nilai-nilai solusi yang memenuhi. Nilai solusi tersebut kemudian dikalikan dengan faktor pengali 9.

Tabel 4. Nilai Dekripsi Cipherteks

Data Plainteks	Data Cipherteks	Data Dekripsi Cipherteks	Tingkat Kesalahan
0.712039	0.039011	0.71208	0.005721%
0.748759	0.112313	0.74880	0.005529%
0.475233	0.104473	0.47520	0.006946%

Dari tabel tersebut, data dekripsi cipherteks memiliki tingkat kesalahan paling tinggi sebesar 0.0069%.

8. Kesimpulan dan Saran Pengembangan

Dat tabel 4, dapat diambil kesimpulan bahwa algoritma enkripsi dan dekripsi dengan menggunakan jaringan saraf tiruan berjalan dengan baik. Tingkat kesalahan sebesar 0.0069% merupakan kesalahan dari pembulatan dan tidak menyebabkan terjadinya kesalahan dekripsi cipherteks.

Untuk pengembangan lebih lanjut, diharapkan pengujian dapat dilakukan dengan menggunakan berkas plainteks, sehingga tingkat akurasi algoritma dapat dievaluasi. Selain itu, untuk meningkatkan keamanan sistem, maka jumlah node masukan, tersembunyi, dan keluaran pada jaringan saraf tiruan dapat ditingkatkan. Namun, peningkatan jumlah node pada jaringan saraf tiruan akan berdampak pada meningkatnya waktu komputasi dan panjang kunci yang dibutuhkan secara kuadratik karena meningkatnya pula jumlah hubungan antara node-node pada jaringan saraf tiruan tersebut.

Kemudian, penggantian metode eliminasi Gauss-Jordan dengan metode eliminasi Gauss dapat mempersingkat waktu komputasi. Hal ini disebabkan karena kompleksitas algoritma metode eliminasi Gauss lebih kecil daripada kompleksitas algoritma metode eliminasi Gauss-Jordan.

DAFTAR PUSTAKA

[1] Laksmi, Agrita. (2002). Penyusunan Melodi pada Berkas MIDI dengan Jaringan Syaraf Tiruan. Departemen Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung.

[2] Mitchell M., Tom. (1997). Machine Learning. The McGraw-Hill Companies, Inc.

[3] Munir, Rinaldi. (2006). Bahan Kuliah IF5054 Kriptografi. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[4] Schmitt, R. S. (2006), Gauss – Jordan Elimination,

http://home.att.net/~srschmitt/index.html, diakses pada 23 Desember 2006.

[5] Schmitt, R. S. (2006), Gauss Jordan 3x3 Solver, *http://home.att.net/~srschmitt/script_gauss_jordan33.html* , diakses pada 23 Desember 2006.