Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi

Vol. 3, No. 2, Juli 2007 ISSN 0216 - 0544

APLIKASI OPERASI ALJABAR MATRIK DALAM AUTENTIFIKASI (PENYANDIAN) DATA DAN SEBAGAI INPUT VEKTOR PEMBELAJARAN PADA *ALGORITMA FEED FORWARD*

Yudha Herlambang

Jurusan Teknik Informatika
Universitas Trunojoyo

ABSTRAK

Sebagaimana telah kita ketahui bersama, bahwa dalam dunia maya yang sangat menentukan di setiap sendi kehidupan pada era gloalisasi seperti saat ini, kemudahan dengan menggunakan transaksi via internet makin sering dilakukan untuk semakin memudahkan dan membuat semua urusan semakin praktis. Dengan adanya *e-mail, e-banking, e-commerce,* dan sebagainya, maka tak ada lagi pembatas dimensi waktu dan tempat ,yaitu antar negara, bahkan antar benua, antara pembeli dan penjual atau antara pihak-pihak yang bertransaksi tak perlu harus bertatap muka. Namun demikian seringkali tak disadari bahwa di balik kecanggihan teknologi informasi yang semakin memudahkan kita dalam kehidupan sehari-hari pasti terdapat sisi lemahnya atau sisi kekurangan teknologi tersebut. Di antaranya ialah factor keamanan atau security. Tak dapat disangkal bahwa perlu adanya langkah pengamanan atas transmisi data, yaitu : *firewall*, antivirus, *decoding, encoding, autentifikasi, Intrusion Detecting System (IDS)*, Scanning dan sebagainya. Dalam paper kali ini penulis memaparkan salah satu contoh teknik penyandian untuk keamanan data yang cukup sederhana dan hanya melibatkan operasi matematis matrik. Di samping itu operasi aljabar matrik juga dapat diterapkan pada vektor input pembelajaran dalam *algoritma feed forward* (umpan maju) , sebagai salah satu teknik pembelajaran dalam *neural network single layer* untuk pengenalan output logika dasar (*or, and , nor, nand*, dan lainnya)

Kata Kunci : *e-commerce, firewall, autentifikasi, feed forward, encoding, decoding.*

1. PENDAHULUAN

Keamanan data merupakan salah satu kebutuhan yang penting. Saat ini sistem komputer yang terpasang makin mudah diakses. Sistem akses jarak jauh menyebabkan masalah keamanan menjadi salah satu kebutuhan yang tidak bisa diabaikan. Di samping itu kecenderungan lain saat ini adalah memberikan tanggung jawab sepenuhnya ke komputer untuk mengelola aktifitas pribadi dan bisnis seperti *system transfer* dana elektronis yang melewatkan uang sebagai aliran bit dan lain sebagainya. Untuk itu diperlukan sistem komputer yang memiliki tingkat keamanan yang dapat terjamin. Keamanan komputer adalah menjamin data atau informasi tidak sampai terbaca, tidak sampai dimodifikasi atau diubah-ubah oleh orang lain yang tidak diberi otorisasi.

Keamanan system dibagi menjadi tiga bagian :

1. Keamanan *eksternal*

   Keamanan eksternal berkaitan dengan fasilitas komputer dari penyusup dan bencana kebakaran atau bencana alam

2. Keamanan *interface* pemakai, dalam hal ini yang berkaitan dengan identifikasi pemakai sebelum pemakai diizinkan mengakses data atau program.

3. Keamanan *internal*, hal ini berkaitan dengan beragam kendali yang dibangun pada sistem perangkat keras dan system perangkat lunak yang menjamin operasi yang handal dan tidak terganggu untuk menjaga integritas data . Sementara itu kebutuhan keamanan sistem komputer dapat dikategorikan menjadi aspek-aspek berikut :

a. *Privacy / Confidentiality*

Inti utama aspek *privacy* atau *confidentiality* adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. *Privacy* lebih terarah pada data-data yang sifatnya *privacy*. Adapun *confidentiality* biasanya berhubungan dengan data yang diberikan pada pihak lain untuk keperluan tertentu dan hanya diperbolehkan hanya untuk keperluan tertentu tersebut.

b. *Integrity*

Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik informasi. Adanya virus, *Trojan horse*, ataupun pemakai lain yang mengubah informasi tanpa ijin merupakan contoh masalah yang harus dihadapi.

c. *Authentification*

Aspek ini berhubungan dengan metode untuk menyatakan bahwa informasi benar-benar orisinil, orang yang mengakses atau yang memberikan informasi ketika diperlukan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.

d. *Availability*

Aspek ini adalah terkait dengan adanya ketersediaan informasi ketika diperlukan. Ssitem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.

Enkripsi merupakan salah satu cara yang dilakukan untuk mengamankan sistem atau informasi dari hal yang akan menyebabkan aspek-aspek di atas tidak terpenuhi, seperti untuk menjaga integritas data atau informasi.

2. MANFAAT

Manfaat penulisan makalah ini antara lain bahwa langkah authentifikasi atau penyandian data selama proses transmisi adalah semata-mata demi keamanan atau *security* data dimaksud hingga sampai di tujuan, yaitu pada sasaran yang berhak. Pemanfaatan proses autentifikasi data ini telah berkembang pada berbagai bidang, antara lain :

2.1. Jasa Telekomunikasi

Kebanyakan standar yang dibuat untuk aplikasi jasa telekomunikasi belum membahas masalah aplikasi keamanan. Seiring dengan semakin banyaknya insiden intruksi terhadap jaringan telekomuinikasi, maka seiring dengan ini aspek keamanan mulai dipertimbangkan dalam jasa telekomunikasi. Bagi pelanggan, terkadang ada informasi konfidensial baik berupa suara (*voice*), data, maupun gambar yang akan dikirimkan ke lawan bacanya. Untuk mengamankan dapat dipergunakan enkripsi. Perkembangan teknik enkripsi memungkinkan proses enkripsi secara *real time* yang tidak mengganggu proses komunikasi. Bagi operator telekomunikasi, selain data *confidential* perusahaan yang telah disebutkan di atas, enkripsi juga diperlukan pada transfer data untuk keperluan manajemen jaringan dengan *transfer on-line data billing*. Di era layanan multimedia, enkripsi diperlukan *copyright* dari informasi yang diberikan agar tetap terjaga.

2.2. Militer dan Pemerintahan

Kepentingan militer merupakan aplikasi enkripsi paling klasik. Dalam lingkungan militer, enkripsi di antaranya digunakan dalam pengiriman pesan rahasia sebagai bagian dari strategi militer untuk mengelabui lawan.

2.3. Data Perbankan

Transaksi atau transfer uang antgar bank dewasa ini telah menjadi rutinitas dalam bisnis perbankan. Informasi transfer uang antar bank harus selalu dalam keadaan terenkripsi, sebab tidak tertutup kemungkinan pembuiatan pesan palsu yang memerintahkan transfer secara tidak sah. Hal ini semakin beresiko apabila telah melibatkan transaksi *e-banking* atau *e-commerce*.

2.4. Pengamanan email

Penggunaan surat elektronik atau email yang semakin meluas dan menggeser penggunaan surat konvensional membuat perlu adanya perlindungan terhadap pesan yang dikirimkan via email ini.

3. LANDASAN TEORI

Algoritma kriptografi atau disebut *chipper*, merupakan persamaan matematika yang digunakan untuk proses enkripsi dan dekripsi yang masing-masing mempunyai hubungan matematis yang cukup erat. Enkripsi merupakan proses untuk mengamankan sebuah pesan *(plaintext)* menjadi pesan tersembunyi *(chipertext)*. Deskripsi merupakan proses kebalikan dari enkripsi (dari *chipertext* menjadi *plaintext*). Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi data disandikan dengan menggunakan key. Untuk membuka data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi.

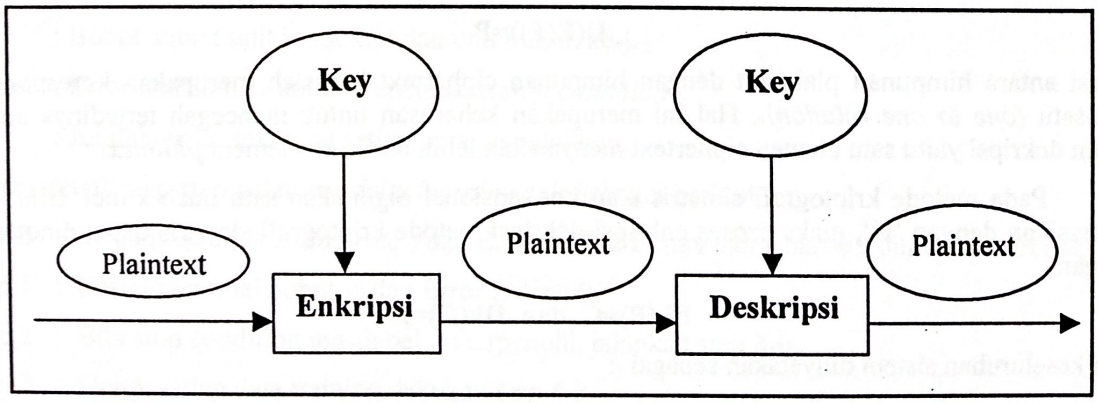
3.1. Terminologi

Kriptografi (*Cryptografi*) berasal dari bahasa Yunani yaitu dari kata " *Crypto* " berarti " *secret* " (rahasia) dan "*Graphy*" yang berarti "*writing*" (tulisan). Kriptografi merupakan suatu cabang ilmu yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah. Proses transformasi dari plaintext menjadi *ciphertext* disebut proses *Encipherment* atau enkripsi (*encryption*). Menurut ISO 7498-2 , terminology yang lebih tepat digunakan adalah "*enchiper*", sedangkan proses mentransformasikan kembali *ciphertext* menjadi *plaintext* disebut proses dekripsi *(decryption)*. Menurut ISO 7498-2 , terminology yang lebih tepat untuk proses ini adalah "*dechiper*".Untuk mengenkripsi dan mendekripsi data, kriptografi menggunakan suatu algoritma (*cipher*) dan kunci (*key*). *Cipher* adalah fungsi matematika yang digunakan untuk mengenkripsi dan mendekripsi data. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data. Algoritma kriptografi modern tidak lagi mengandalkan keamanannya pada kerahasiaan algoritma, tetapi kerahasiaan kunci. *Plaintext* yang sama bila disandikan denan kunci yang berbeda akan menghasilkan *ciphertext* yang berbeda pula. Dengan demikian algoritma kriptografi dapat bersifat umum dan boleh diketahui oleh siapa saja, namun tanpa adanya pengetahuan tentang kunci, data yang tersandi tetap saja tak dapat dipecahkan. Sistem kriptografi atau *Cryptosystem* adalah sebuah algoritma kriptografi ditambah semua kemungkinan *plaintext*, *ciphertext* dan kunci.

3.2. Algoritma Kriptografi

Berdasarkan kunci yang dipakai, maka algoritma kriptografi dapat dibedakan atas dua golongan , yaitu

3.2.1. Symmetric Algorithms : merupakan algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Proses enkripsi – dekripsi algoritma kriptografi simeteris dapat dilihat pada gambar berikut di bawah ini :



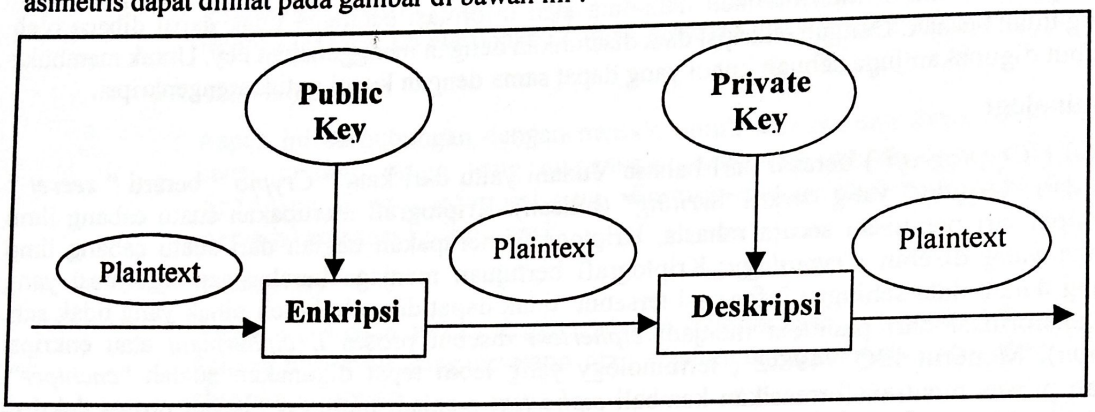
Gambar 1. Proses Enkripsi-Deskripsi kunci simetris.

Algoritma kriptografi simetris dibagi menjadi 2 kategori, yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Blok Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi

pada satu bit atau satu byte data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau byte data per blok.

3.2.2. Asymmetric Algorithms.

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, namun kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Proses enkripsi-dekripsi algoritma asimetris dapat dilihat pada gambar di bawah ini :



Gambar 2. Proses Enkripsi-Deskripsi kunci Asimetris.

Pada algoritma *public key* ini, semua orang dapat mengenkripsi data dengan memakai public key penerima yang telah diketahui secara umum. Akan tetapi data yang telah terenkripsi tersebut hanya dapat didekripsi dengan menggunakan private key yang hanya dapat diketahui oleh penerima saja.

4. DASAR MATEMATIS

Dasar matematik yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu himpunan berisi elemen *plaintext* dan himpunan berisi elemen *ciphertext*. Enkripsi dan dekripsi merupakan fungsi transformasi antara dua himpunan tersebut. Bila himpunan *plaintext* dinotasikan dengan P dan himpunan ciphertext dinotasikan dengan C, sedang fungsi enkripsi dinotasikan dengan E dan fungsi dekripsi dengan D, maka proses enkripsi –dekripsi dapat dinyatakan dalam notasi matematis dengan :

$$E(P) = C \text{ dan } D(C)=P$$

Karena proses enkripsi dekripsi bertujuan memperoleh kembali data asal, maka :

$$D(E(P))=P$$

Relasi antara himpunan plaintext dengan himpunan ciphertext haruslah merupakan korespondensi satu-satu *(one to one relation)*.. Hal ini merupakan keharusan untuk mencegah terjadinya ambigu dalam dekripsi yaitu satu elemen ciphertext menyatakan lebih dari satu element *plaintext*.

Pada metode kriptografi simetris atau konvensional digunakan satu buah kunci. Bila kunci dinotasikan dengan 'K', maka proses enkripsi-dekripsi metode kriptografi simetris dapat dinotasikan dengan :

$$Ek(P)=C \text{ dan } Dk(C)=P$$

Dan keseluruhan sistem dinyatakan sebagai :

$$Dk(Ek(P))=P$$

Pada metode kriptografi asimetris digunakan kunci umum untuk enkripsi dan kunci pribadi untuk dekripsi. Bila kunci umum dinotasikan dengan 'PK' dan kunci pribadi dinotasikan dengan 'SK', maka proses enkripsi-dekripsi metode kriptografi nirsimetris dapat dinotasikan dengan :

Epk(P)=C dan Dsk(C)=P

Dan keseluruhan sistem dinyatakan sebagai :

Dsk(Epk(P))=P

5. ARSITEKTUR JARINGAN *NEURAL NETWORK*

Arsitektur Jaringan Syaraf Tiruan pada *Delta Learning Rule* hanya terdiri atas unit input dan unit output . Arsitektur yang cukup sederhana ini disebut single layer. Unit input menerima input dari luar , sedangkan unit output mengeluarkan respon sesuai dengan inputnya. Metode pembelajaran ini termasuk jenis *supervised learning* yang paling sederhana. Topologi jaringan yang digunakan pada aturan pembelajaran *delta learning rule* adalah jaringan *feed forward* dengan metode *supervised* (terawasi). Jaringan *feedforward* merupakan hubungan antar *node* dan tidak terdapat perputaran. Jaringan *feed forward* biasanya menghasilkan sebuah respon terhadap input secara cepat . Metode supervised merupakan metode yang hasil sebenarnya diketahui secara pasti dan diberikan pada jaringan pada saat proses pelatihan sehingga jaringan tersebut dapat menyesuaikan bobotnya untuk mencocokkan output yang dihasilkannya terhadap target yang seharusnya dicapai oleh jaringan. Dalam metode tersebut, seolah – olah ada guru yang mengajari jaringan.

5.1. Algoritma Delta Learning Rule

Saat umpan maju (*feedforward*), setiap unit input (X) akan menerima sinyal input dan akan menyebarkan sinyal tersebut pada tiap unit output. Kemudian, setiap unit output (Y) juga akan menghitung aktivasinya (y) untuk menghasilkan respons terhadap input yang diberikan jaringan. Saat proses pelatihan (*training*), setiap unit output membandingkan aktivasinya dengan nilai target (*desired output*) untuk menentukan besarnya *error*. Perhitungan *error* ini digunakan untuk menentukan *stop condition*.

Berikut diberikan notasi-notasi yang kerap digunakan pada algoritma pelatihan di atas :

X : Data training untuk input. X = (x1,……,xt,……..xn)

t : Data training untuk output (*target/ desired output*). T = (t1,……,tk,…..tm)

a : *Learning Rate* yakni parameter yang mengontrol perubahan bobot selama pelatihan. Bila *Learning Rate* besar, jaringan semakin cepat belajar, namun hasilnya kurang akurat. *Learning Rate* biasanya dipilih antara 0 dan 1.

X1 : yaitu unit input ke-i

Y1 : Unit output ke-j.

Wij : Bobot antara unit input ke-i dan unit output ke-j.

E.max : Error maksimum untuk menentukan *stop condition*.

E : Jumlah *error* sekarang setiap putaran pelatihan.

Secara detail, step-step pelatihan *delta learning rule* sebagai berikut :

Step 0 : Inisialisasi a (*learning rate*) dan E.max. Nilai E.max dan a harus lebih besar dari nol.

Step 1 : Inisialisasi nilai bobot w dan Error E diset 0.

Step 2 : Bila stop condition masih belum terpenuhi, jalankan step 3-9.

Step 3 : Untuk setiap data training, lakukan step 4-8.

5.2. Umpan Maju (*Feedforward*)

Step 4 : Setiap unit input (X1, i=1,….,n) menerima sinyal input xi dan menyebarkan sinyal tersebut pada seluruh unit pada unit output. Di sini perlu diketahui bahwa input xi yang dipergunakan adalah *input training* data yang diskalakan. Pertama, input yang mungkin dipakai dalam sistem dicari nilai terendah dan tertingginya. Kemudian, skala yang digunakan tergantung dari fungsi aktivasinya. Bila yang dipakai adalah fungsi *sigmoid biner* yang memiliki harga terendah = 0 dan harga tertinggi = 1, nilai input terendah juga dianggap = 0 dan nilai tertinggi dianggap = 1. Nilai-nilai di antaranya bervariasi antara 0 dan 1. Sedangkan, bila yang digunakan fungsi *sigmoid bipolar* , maka range nilainya juga bervariasi mulai-1 hingga 1.

Step 5 :Setiap unit output (Yj, j=1,…..,m) akan menjumlahkan sinyal-sinyal input yang sudah berbobot, termasuk biasnya.

$y\_in = \sum_{i-1}^{n} x_i w_{ij}$ dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal output dari unit output yang bersangkutan.

$y_j = f(y\_in_j)$,

lalu mengirim sinyal output tersebut ke seluruh unit pada unit output.

5.3. Pembaharuan Bobot (*Weight Adjusting*) Dan Bias

Step 6: Menghitung perubahan bobot dari setiap unit input (Xi, i=1,……,n) ke unit output (Yj,j=1,……,m).

Step 7 : Setiap unit output(Yj,j-1,……,m) akan memperbaharui bias dan bobotnya dari setiap unit input (i=1,…..,n).

5.4. Perhitungan Total Error

Step 8 : Menghitung Error dengan menambahkan error yang sekarang ke variable E.

$E = E + 0.5(y_j - t_j)^2$.

Step 9 : Memeriksa *stop conditioning.*

Bila E< E.max, maka stop condition terpenuhi dan pelatihan selesai.

Bila E>E.max , maka E diset nol dan iterasi (putaran) pelatihan baru dimulai lagi dengan kembali ke step 3.

6. CONTOH KASUS & PEMBAHASAN APLIKASI MATRIK PADA AUTENTIFIKASI

Contoh kasus penerapan *Encoding* dan *Decoding* sebagai langkah autentifikasi (penyandian pesan-pesan rahasia. Sebagaimana telah kita ketahui bahwa *Encoding* merupakan kegiatan untuk menyembunyikan pesan, sehingga orang yang tak berhak atau tak memiliki otorisasi untuk melakukan akses, tidak akan mampu mengetahui pesan yang sebenarnya, sedangkan *encoding* adalah kegiatan untuk menterjemahkan pesan yang telah diencoding, sehingga dapat diterima sebagaimana pesan aslinya.

Apabila kita perhatikan urutan huruf-huruf berikut :

a	b	c	d	e	f	g	h	i	j	K	l	m
1	2	3	4	5	6	7	8	9	10	11	12	13

n	o	p	q	r	s	t	u	v	w	X	y	z
14	15	16	17	18	19	20	21	22	23	24	25	26

Misalnya : pesan „ ***Pergi ke Pati*** „ yang mana oleh urut-urutan huruf-huruf di atas disampaikan dengan pesan tanpa encoding, yaitu :

p	e	r	g	i		k	e		P	a	t	i	
16	5	18	7	9	27	11	5	27	16	1	20	9	27

Maka apabila pilih sembarang *matrik encoding* , misalkan :

$\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}$, maka pesan terkirim dalam bentuk *plain text* tersandi menjadi :

$$\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}16\\ 5\end{bmatrix}=\begin{bmatrix}26\\ 31\end{bmatrix},\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}18\\ 7\end{bmatrix}=\begin{bmatrix}32\\ 39\end{bmatrix},\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}9\\ 27\end{bmatrix}=\begin{bmatrix}63\\ 90\end{bmatrix},\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}11\\ 5\end{bmatrix}=\begin{bmatrix}21\\ 26\end{bmatrix}$$

$$\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}27\\ 16\end{bmatrix}=\begin{bmatrix}59\\ 75\end{bmatrix},\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}01\\ 20\end{bmatrix}=\begin{bmatrix}41\\ 61\end{bmatrix},\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}\begin{bmatrix}9\\ 27\end{bmatrix}=\begin{bmatrix}63\\ 90\end{bmatrix}$$

Maka diperoleh *plaintext* sebagai berikut :

26	31	32	39	63	90	21	26	59	75	41	61	63	90

Pada pihak yang menerima pesan di atas, tentunya untuk dapat membaca pesan tersebut harus mengubah pesan yang diterima *dengan melakukan kegiatan decoding* , yaitu dengan mengalikan invers dari matriks encoding, yaitu :

Invers matrik encoding : $\begin{bmatrix}1 & 2\\ 1 & 3\end{bmatrix}^{-1}=\frac{1}{1.3-1.2}\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}=\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}$

$$\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}26\\ 31\end{bmatrix}=\begin{bmatrix}16\\ 5\end{bmatrix},\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}32\\ 39\end{bmatrix}=\begin{bmatrix}18\\ 7\end{bmatrix},\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}63\\ 90\end{bmatrix}=\begin{bmatrix}9\\ 27\end{bmatrix}$$

$$\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}21\\ 26\end{bmatrix}=\begin{bmatrix}11\\ 5\end{bmatrix},\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}59\\ 75\end{bmatrix}=\begin{bmatrix}27\\ 16\end{bmatrix},\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}41\\ 61\end{bmatrix}=\begin{bmatrix}01\\ 20\end{bmatrix}$$

$$\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}63\\ 90\end{bmatrix}=\begin{bmatrix}9\\ 27\end{bmatrix}$$

Demikianlah pesan semula dapat diperoleh ulang, seperti saat dikirimkan.

Terakhir , tinggal dilakukan langkah identifikasi ke bentuk alphabet seperti halnya data mentah yang dikirimkan semula.

16	5	18	7	9	27	11	5	27	16	1	20	9	27
p	e	R	g	i		k	e		P	a	t	i	

Sebenarnya bisa saja penyelesaian di atas disederhanakan atau dipermudah, yaitu dengan langsung mengalikan invers matrik dengan semua komponen data *plaintext*, pada langkah *decoding*. Jadi tak perlu dilakukan perkalian dua baris 1 kolom. Hal ini hanya untuk menghindari ketidakefisienan.

$$\begin{bmatrix}3 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}26 & 32 & 63 & 21 & 59 & 41 & 63\\ 31 & 39 & 90 & 26 & 75 & 61 & 90\end{bmatrix}=\begin{bmatrix}16 & 18 & 9 & 11 & 27 & 01 & 9\\ 5 & 7 & 27 & 5 & 16 & 20 & 27\end{bmatrix}$$

Dengan demikian hasil outputnya setelah dilakukan *decoding* sama dengan tabel di atas.

Cara di atas dapat lebih disederhanakan lagi, yaitu apabila dipilih matrik *encoding* yang ordenya sesuai dengan urutan data yang akan dikalikan, dalam hal ini tidak harus selalu 2 baris 1 kolom, namun bisa saja langsung misalkan 4,5,atau bahkan 7 data terurut sekaligus.

Contohnya sebagai berikut , misalkan kita pilih sembarang matrik *Encoding* (E),

$$E=\begin{bmatrix} 1 & 3 & 1 & 2 & 1 & 1 & 4 \\ 3 & 2 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 5 & 1 & 2 & 4 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 0 & 3 & 2 \\ 1 & 0 & 0 & 3 & 1 & 2 & -1 \end{bmatrix}$$

dan invers matrik E =

$$\begin{bmatrix} 0{,}1159 & 0{,}1594 & -0{,}0932 & 0{,}1698 & -0{,}1491 & 0{,}1014 & -0{,}1242 \\ 0{,}1449 & 0{,}5507 & -0{,}1335 & -0{,}2122 & 0{,}1863 & -0{,}3768 & 0{,}1553 \\ 0{,}0435 & -0{,}4348 & -0{,}0186 & -0{,}0994 & 0{,}7702 & 0{,}0870 & -0{,}0248 \\ 0{,}2464 & -0{,}4638 & -0{,}1056 & 0{,}1035 & 0{,}0311 & 0{,}1594 & 0{,}1925 \\ -0{,}2609 & 0{,}6087 & 0{,}2547 & 0{,}0248 & -0{,}1925 & -0{,}5217 & 0{,}0062 \\ -0{,}1159 & 0{,}1594 & 0{,}1211 & -0{,}1874 & -0{,}0062 & 0{,}1014 & 0{,}1615 \\ 0{,}1304 & -0{,}3043 & 0{,}0871 & 0{,}1304 & -0{,}2609 & 0{,}2609 & -0{,}2174 \end{bmatrix}$$

Pada langkah *Encoding* dilakukan langkah perkalian matrik *Encoding* (E) dengan input data sebanyak 7 data langsung dan sebanyak 2 kolom langsung :

$$\begin{bmatrix} 1 & 3 & 1 & 2 & 1 & 1 & 4 \\ 3 & 2 & 1 & 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 5 & 1 & 2 & 4 & 3 & 1 & 2 \\ 2 & 1 & 2 & 1 & 1 & 1 & 1 \\ 3 & 1 & 1 & 1 & 0 & 3 & 2 \\ 1 & 0 & 0 & 3 & 1 & 2 & -1 \end{bmatrix} \cdot \begin{bmatrix} 16 & 5 \\ 5 & 27 \\ 18 & 16 \\ 7 & 01 \\ 9 & 20 \\ 27 & 9 \\ 11 & 27 \end{bmatrix} = \text{Plaint Text} = \begin{bmatrix} 143 & 241 \\ 168 & 178 \\ 223 & 261 \\ 225 & 211 \\ 127 & 126 \\ 181 & 140 \\ 89 & 19 \end{bmatrix} =$$

Pada saat data matrik *PlaintText* telah diterima oleh tujuan , maka dilakukan pembalikan sandi atau *Decoding* untuk mengetahui kembali isi pesan oleh penerima, dengan langkah *inverse* matrik yang dikalikan matrik *Plainttext* sebagai berikut :

$$\begin{bmatrix} 0{,}1159 & 0{,}1594 & -0{,}0932 & 0{,}1698 & -0{,}1491 & 0{,}1014 & -0{,}1242 \\ 0{,}1449 & 0{,}5507 & -0{,}1335 & -0{,}2122 & 0{,}1863 & -0{,}3768 & 0{,}1553 \\ 0{,}0435 & -0{,}4348 & -0{,}0186 & -0{,}0994 & 0{,}7702 & 0{,}0870 & -0{,}0248 \\ 0{,}2464 & -0{,}4638 & -0{,}1056 & 0{,}1035 & 0{,}0311 & 0{,}1594 & 0{,}1925 \\ -0{,}2609 & 0{,}6087 & 0{,}2547 & 0{,}0248 & -0{,}1925 & -0{,}5217 & 0{,}0062 \\ -0{,}1159 & 0{,}1594 & 0{,}1211 & -0{,}1874 & -0{,}0062 & 0{,}1014 & 0{,}1615 \\ 0{,}1304 & -0{,}3043 & 0{,}0871 & 0{,}1304 & -0{,}2609 & 0{,}2609 & -0{,}2174 \end{bmatrix} \cdot \begin{bmatrix} 143 & 241 \\ 168 & 178 \\ 223 & 261 \\ 225 & 211 \\ 127 & 126 \\ 181 & 140 \\ 89 & 19 \end{bmatrix} =$$

$$\begin{bmatrix} 16 & 5 \\ 5 & 27 \\ 18 & 16 \\ 7 & 1 \\ 9 & 20 \\ 27 & 9 \\ 11 & 27 \end{bmatrix}$$ = Matriks data awal sebelum dilakukan *Encoding* (pesan tanpa *Encoding*)

Tampak bahwa hasil akhir menunjukkan matrik data awal sebelum dilakukan langkah penyandian data. Sehingga matrik yang terakhir diperoleh tinggal dilakukan langkah identifikasi ke Alphabet saja, sehingga menunjukkan pesan awal ***: " Pergi ke Pati ".***

6.1.Contoh Kasus & Pembahasan Aplikasi Matrik Sebagai Vektor Input Pembelajaran Pengenalan Fungsi Logika Pada Algoritma *Learning Feed Forward* (*Delta Learning Rule*)

Contoh aplikasi *Delta Learning Rule.*

Contoh aplikasi *delta learning rule* pada pembelajaran fungsi logika AND seperti berikut :

Bobot (*Weight*) diinisialisasi dengan nilai random (misal : w1=0.1 ; w2=-0.2). Pada fungsi AND, inputnya adalah 00,01,10,11 dan target yang diharapkan 0,0,0,1. Sedangkan E (*error*) diinisialisasi 0. Ditentukan a(*learning rate*) =0.1 dan E.max 0.1.

Kita lakukan iterasi 1 :

Input 1	Input 2	Target	w1*x1	w2*x2	Y-In	Y Out	error	W1 new	W2 new
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.50000	***-0.50000***	0.10000	-0.20000
0.00000	1.00000	0.00000	0.00000	-0.20000	-0.20000	0.45	***-0.45***	0.10000	-0.2045
1.00000	0.00000	0.00000	0.10000	0.00000	0.10000	0.52498	***-0.52498***	0.047502	-0.2045
1.00000	1.00000	1.00000	0.04750	-0.24502	-0.157	0.46	***0,54***	0,101502	-0,1505

Perhitungan atas pembaharuan bobot adalah sebagai berikut :

Pasangan data training I **(epoch I)**

➔ Y-in diperoleh dari : $y\_in = \sum_{i-1}^{n} x_i w_{ij} = [w1 \quad w2] \begin{bmatrix} x1 \\ x2 \end{bmatrix}$ w1.x1 + w2.x2 = **0.**

Sehingga Y1 (Y.Out) = $\frac{1}{1+e^{-(y\_in)}} = \frac{1}{1+e^{-(0)}} = \frac{1}{1+1} = \frac{1}{2} = 0.5$,

Dan error = (target -Y.out) = (0 – 0.5) = -0.5.

Maka dengan demikian langkah perhitungan *adjustment weight* adalah :

w1(baru) = w1(lama) + (a (error).x1) = 0.1 + (0.1(-0.5).0) = **0.1.**

w2 (baru) = w2 (lama) + (a.(error).x2) = -0.2+ (0.1(-0.5).0) = **-0.2**

Pasangan data training II (epoch II)

➔ Y-in diperoleh dari : $y\_in = \sum_{i-1}^{n} x_i w_{ij} = [w1 \quad w2] \begin{bmatrix} x1 \\ x2 \end{bmatrix}$ w1.x1 + w2.x2 = **-0,2.**

Sehingga Y1 (Y.Out) = $\frac{1}{1+e^{-(y\_in)}} = \frac{1}{1+e^{0,2}} = 0,45$,

Dan error = (target -Y.out) = (0 – 0.45) = -0.45.

Maka dengan demikian langkah perhitungan *adjustment weight* adalah :

w1(baru) = w1(lama) + (a (error).x1) = 0.1 + (0.1(-0.45).0) = **0.1.**

w2 (baru) = w2 (lama) + (a.(error).x2) = -0.2+ (0.1(-0.45).1) = **-0,2045**

Pasangan data training III (epoch III)

➔ Y-in diperoleh dari : $y\_in = \sum_{i-1}^{n} x_i w_{ij} = [w1 \quad w2] \begin{bmatrix} x1 \\ x2 \end{bmatrix}$ w1.x1 + w2.x2 = **0,1**

Sehingga Y1 (Y.Out) = $\frac{1}{1+e^{-(y\_in)}} = \frac{1}{1+e^{-0,1}} = 0,524979$,

Dan error = (target -Y.out) = (0 – 0.524979) = **-0.524979.**

Maka dengan demikian langkah perhitungan *adjustment weight* adalah :

w1(baru) = w1(lama) + (a (error).x1) = 0.1 + (0.1(-0.524979).1) = **0.047502**

w2 (baru) = w2 (lama) + (a.(error).x2) = -0.2045+(0.1(-0.524979).0)=**-0.2045**

Pasangan data training IV (epoch IV)

➔ Y-in diperoleh dari : $y\_in = \sum_{i-1}^{n} x_i w_{ij} = [w1 \quad w2] \begin{bmatrix} x1 \\ x2 \end{bmatrix}$ w1.x1 + w2.x2 =

0,047502+(-0,2045)=**-0,157**

Sehingga Y1 (Y.Out) = $\frac{1}{1+e^{-(y\_in)}} = \frac{1}{1+e^{0,157}} = \frac{1}{1+1,17} = \frac{1}{2,17} = 0,46.$

Dan error = (target -Y.out) = (1 – 0.46) = **0,54**

Maka dengan demikian langkah perhitungan *adjustment weight* adalah :

w1(baru) = w1(lama) + (a (error).x1) = 0.047502 + (0.w1(0,54)(1)) =**0,101502**

w2 (baru) = w2 (lama) + (a.(error).x2) =-0,2045+(0,1(0,54)(1))= **-0,1505**

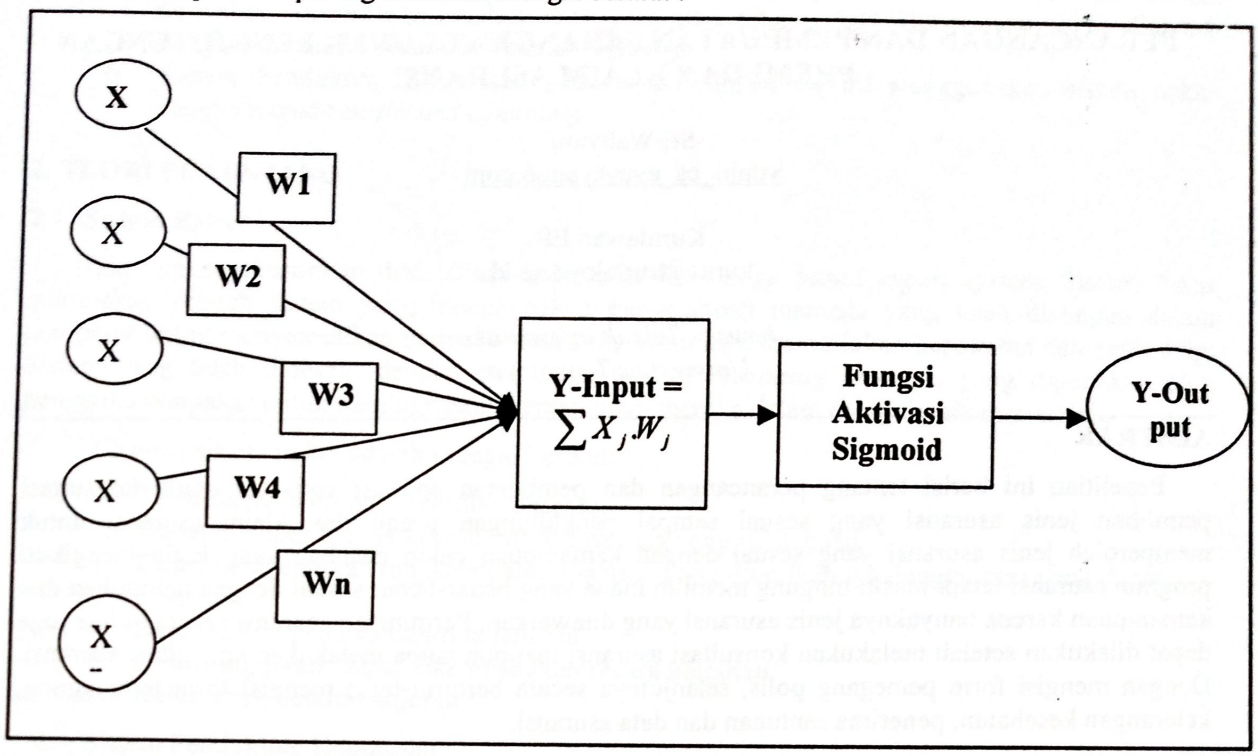
Demikian seterusnya pada *Learning* untuk Iterasi II dan iterasi-iterasi atau perulangan loop berikutnya , sedemikian hingga diperoleh target yang diinginkan dan keluaran jaringan pembelajaran yang hampir sama atau dengan kata lain *error* atau kesalahannya mendekati nol yaitu error mendekati tingkat yang diharapkan atau kesalahan seminimum mungkin, terutama kesalahan rms (*root mean square*) atau kesalahan rata-rata kuadrat minimum, yaitu : $E_{minimum} = E_{initial} + 0,5\sum(y - t_j)^2$.

6.2. Penggunaan Aplikasi Matriks Pada Langkah Pembelajaran Dengan *Metode Feed Forward* (*Algoritma Delta Rule*)

Setelah kita uraikan algoritma di atas, maka tampak peranan operasi aljabar matrik,yaitu perkalian matrik pada saat menghitung nilai Y-input yang akan diumpankan pada fungsi aktivasi untuk memperoleh Y-Output yang akan dibandingkan dengan target yang dikehendaki. Pada setiap training pasangan data I hingga IV, nilai Y-input ialah :

$$\text{Y-input} = \sum x_j.W_j = x_1W_1 + x_2W_2 + ...x_nW_n = [x_1 \quad x_2 \quad x_3 \quad ..... \quad x_n] \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ ..... \\ W_n \end{bmatrix}$$

Untuk lebih jelasnya uraian di atas, dengan menggunakan arsitektur Jaringan *Single Layer Perceptron* dapat digambarkan sebagai berikut :



Gambar 3. Arsitektur Jaringan Pembelajaran Sederhana , Single Layer Network

7. KESIMPULAN

1. Telah diperoleh bukti bahwa aplikasi invers matrik dan perkalian matrik dapat dipergunakan dalam proses penyandian data yang terkait dengan tingkat pengamanan data. Invers matrik dan *orde matrik Encoding* dapat disesuaikan dimensinya tergantung kondisi dan banyaknya data yang akan di Encode atau pesan yang akan disandikan. Untuk proses perkalian dan invers matrik yang berdimensi besar, dapat dipergunakan bantuan software aplikasi Matlab 6.00.
2. Telah ditunjukkan pula bahwa aplikasi perkalian matrik dapat pula dipergunakan pada proses pembelajaran pada algoritma *Feed Forward* yang termasuk salah satu rumpun metode pembelajaran pada *Artificial Neural Network*. Terutama pada proses pencarian fungsi Y-Input yang akan diumpankan pada fungsi Aktivasinya pada bentuk arsitektur jaringan pembelajaran yang Single Layer atau sederhana (bukan *Multilayer Perceptron*)

   Matrik pertama berupa matrik Baris yaitu matrik yang berisi Bobot (*Weight Matrix*) dan matrik kedua yaitu Matrik Vektor Pasangan Input Pembelajaran (berupa matrik Kolom).

8. DAFTAR PUSTAKA

1. Ain Khurun, *"Perancangan dan Pembuatan Perangkat Lunak Enkripsi File Kunci Simetris Dengan Algoritma Kriptografi Rijndael"*, Tugas Akhir Jurusan Teknik Informatika , Universitas Trunojoyo Madura, 2006.
2. http://www.sttelkom.ac.id /kuliahmatriks, 8 April 2007.
3. Setiawan Koeswara, *" Artificial Intelligence"*, Bina Pustaka Ilmu, Jakarta 2003.