Penggunaan Algoritma Diffie-Hellman dalam Melakukan Pertukaran Kunci

Michael Ingga Gunawan 13511053
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
michael.ingga@students.itb.ac.id

***Abstrak*—Makalah ini membahas tentang pertukaran kunci dalam penyampaian informasi yang hanya ditujukan kepada pihak-pihak tertentu. Untuk menjaga tersebarnya informasi rahasia ini, maka digunakanlah kriptografi pada kunci publik agar pihak-pihak lain yang tidak diinginkan tidak dapat mengetahui informasi tersebut. Metode kriptografi yang dapat digunakan dalam pertukaran kunci tersebut adalah dengan menggunakan algoritma Diffie-Hellman dan algoritma Rivest-Shamir-Adleman(RSA). Kedua algoritma ini menjamin keamanan tingkat tinggi dalam pertukaran kunci rahasia antar pihak-pihak tertentu.**

Pada makalah ini, saya akan membahas pertukaran kunci dengan menggunakan algoritma Diffie-Hellman karena algoritma ini menghasilkan performa yang lebih baik dibandingkan algoritma Rivest-Shamir-Adleman(RSA).

***Kata Kunci*—Diffie-Hellman, Kriptografi, Kunci, RSA.**

I. PENDAHULUAN

Saat ini, perkembangan teknologi yang terjadi begitu pesat. Pertukaran informasi yang terjadi antar kota dan antar Negara pun semakin cepat sehingga semua informasi tersebut dapat diketahui secara global. Hal ini menyebabkan persebaran informasi yang sudah tersebar di dunia maya tidak dapat ditarik kembali dengan mudah.

Oleh karena itu, di dalam memasukkan data pribadi dan file-file pribadi ke dalam dunia maya harus dilakukan secara bijak dan aman. Kartu identitas, nomor ATM, kartu kredit, alamat, nomor telepon, foto, video, dll merupakan hal yang tidak boleh tersebar secara sembarangan di dunia maya karena data-data pribadi tersebut dapat digunakan oleh pihak-pihak lain untuk hal-hal yang tidak kita inginkan.

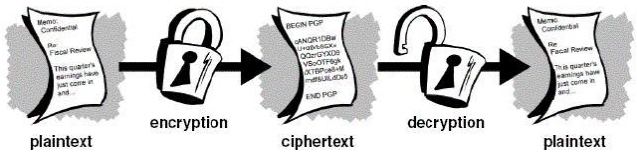
Untuk menangani hal-hal tersebut agar tidak tersebar ke orang yang tidak diinginkan, maka dilakukanlah perubahan-perubahan data pribadi tersebut menjadi suatu sandi-sandi yang tidak mempunyai makna. Tentunya hal ini dilakukan tidak secara sembarangan karena perubahan-perubahan tersebut menggunakan algoritma tertentu sehingga ketika sudah diubah menjadi sandi tertentu dapat dikembalikan lagi menjadi keadaan awal. Cara dan metode-metode untuk menjaga kerahasiaan suatu data atau informasi ini disebut kriptografi. Metode tersebut dalam pengiriman data dapat diilustrasikan seperti ini: data awal → enkripsi → chiperteks (sandi) → dekripsi → data awal. Dalam melakukan enkripsi dan dekripsi tersebut diperlukanlah suatu kunci untuk mengubah dari data awal menjadi sandi tersebut dan untuk mengubah dari sandi tersebut menjadi data awal. Sehingga bagian paling penting dalam penyamaran data ini adalah pada bagian kunci tersebut. Kunci tersebut tidaklah boleh jatuh kepada pihak-pihak yang tidak diinginkan.

Oleh karena itu, dalam pertukaran kunci kepada pihak yang diinginkan diperlukan suatu metode yang aman. Metode tersebut yang dapat digunakan dalam pertukaran kunci ini adalah dengan algoritma Diffie-Hellman dan algoritma Rivest-Shamir-Adleman(RSA). Pada makalah ini kita akan memfokuskan pembahasan pada algoritma Diffie-Hellman yang digunakan dalam pertukaran kunci tersebut.

II. DASAR TEORI

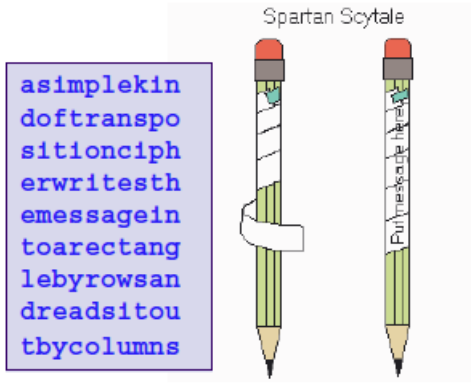
A. *Kriptografi*

Secara etimologi, kata kriptografi berasal dari bahasa Yunani yaitu "kriptos" dan "graphia". Kriptos berarti sesuatu yang rahasia atau misterius dan Graphia berarti tulisan. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan suatu data atau informasi dengan cara mengubah data atau informasi tersebut menjadi sesuatu bentuk sandi yang tidak dapat diketahui oleh orang lain kecuali pengirim dan penerima pesan sandi tersebut. Pesan yang akan dirahasiakan dan belum diubah menjadi sandi disebut *plainteks*. Yang nantinya *plainteks* tersebut akan diubah menjadi *chiperteks*, yaitu pesan yang telah diubah menjadi sandi tertentu. Proses ini disebut *enkripsi* dan proses untuk mengubah dari *chiperteks* kembali ke *plainteks* disebut *dekripsi*. Gambar 1 menjelaskan diagram proses yang dimaksud.



Gambar 1: Proses enkripsi dan dekripsi suatu plainteks

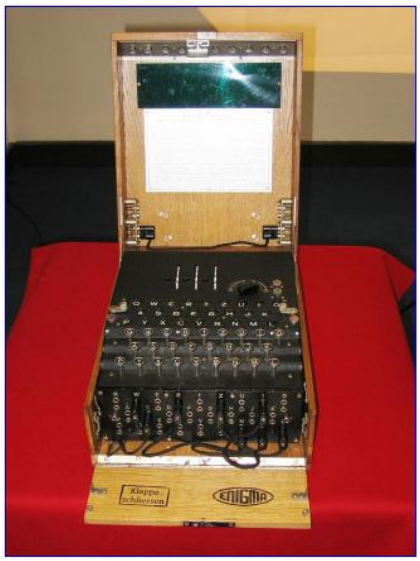
Dalam menjaga kerahasiaan tersebut, terdapat kriptografi yang hanya mengacak huruf (Transposisi) dan mengganti huruf dengan huruf yang lainnya (Substitusi). Contoh dari kriptografi transposisi adalah *Rail fence* (semua huruf disambung tanpa spasi), *simple transposition* (pesan ditulis mendatar dan dikirimkan secara vertical), dan *Spartan scytale* seperti pada gambar 2. Sedangkan contoh kriptografi substitusi adalah *Caesar cipher*, yaitu menggeser 3 huruf ke kanan seperti pada gambar 3, misal nama "ANTO" diubah menjadi *cipherteks*, yaitu "dqwr". Contoh lainnya adalah *enigma*(rotor) yang digunakan oleh Jerman pada perang dunia ke 2 dapat dilihat pada gambar 4.



Gambar 2: Contoh kriptografi dengan mengacak huruf(Transposisi)

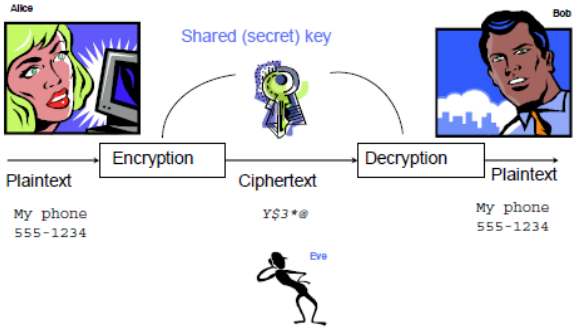
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
d e f g h i j k l m n o p q r s t u v w x y z a b c

Gambar 3: Contoh kriptografi substitution dengan menggunakan Caesar cipher



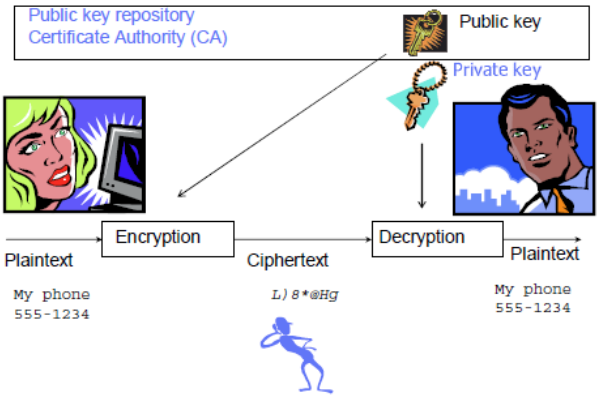
Gambar 4: Contoh kriptografi substitution dengan menggunakan enigma(rotor)

Berdasarkan kesamaan kunci pada proses enkripsi dan dekripsi, algoritma sandi dibedakan menjadi 2, yaitu algoritma sandi kunci-simetris dan algoritma sandi kunci-asimetris. Algoritma sandi kunci-simetris adalah metode kriptografi dengan menggunakan kunci yang sama pada saat melakukan *enkripsi* dan *dekripsi*. Algoritma ini sering disebut juga sebagai *Private Key* karena kunci yang digunakan tidak boleh diketahui oleh orang lain. Proses *enkripsi* dan *dekripsi* dengan menggunakan kunci simetrik dapat dilihat pada gambar 5.



Gambar 5: Proses enkripsi dan dekripsi dengan menggunakan kunci simetrik

Pada Algoritma sandi kunci-asimetris penggunaan kunci untuk *enkripsi* dan *dekripsi* berbeda. Kunci untuk *enkripsi* tidak sama dengan kunci *dekripsi*. Algoritma ini sering juga disebut sebagai *Public Key* karena kunci *enkripsi* tersebut bersifat tidak rahasia dan dapat diketahui oleh orang lain. Contohnya adalah algoritma *knapsack*, *RSA*, dan *Diffie-Hellman*. Proses *enkripsi* dan *dekripsi* dengan menggunakan kunci asimetrik dapat dilihat pada gambar 6.



Gambar 6: Proses enkripsi dan dekripsi dengan menggunakan kunci asimetrik

B. Algoritma Rivest-Shamir-Adleman (RSA)

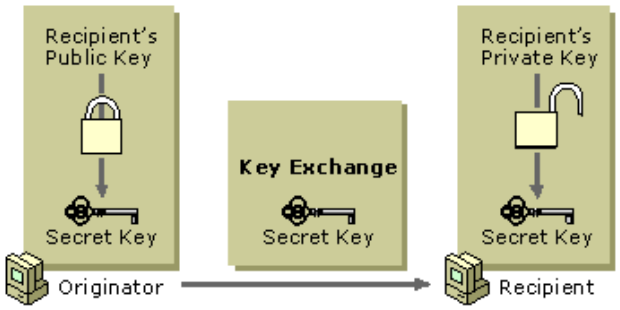
Algoritma ini mendasarkan proses *enkripsi* dan *dekripsi* nya pada konsep bilangan prima dan aritmatika modulo. Kunci *enkripsi* dan kunci *dekripsi* tersebut merupakan bilangan bulat. Kunci *enkripsi* tersebut bersifat publik, sedangkan kunci *dekripsi* nya bersifat privat. Kekuatan algoritma RSA ini terletak pada tingkat kesulitan dalam memfaktorkan bilangan non primamenjadi faktor primanya. Secara ringkas, algoritma RSA adalah sebagai berikut:

1. Pilih 2 buah bilangan prima sembarang yang tidak boleh diketahui oleh orang lain, sebut a dan b.
2. Hitung $n = a \times b$. Besaran n tidak dirahasiakan.
3. Hitung $m = (a - 1) \times (b - 1)$. Setelah itu, a dan b dapat dihapus untuk mencegah diketahuinya oleh

orang lain.

4. Bangkitkan kunci *dekripsi*, *d*, dengan kekongkuenan $ed \equiv 1 \pmod{m}$. Lakukan *enkripsi* terhadap isi pesan dengan persamaan $c_i = p_i^e \bmod n$, yang dalam hal ini p_i adalah blok *plainteks*, c_i adalah *chiperteks* yang diperoleh, dan *e* adalah kunci *enkripsi*. Harus dipenuhi persyaratan bahwa nilai p_i harus terletak dalam himpunan nilai 0, 1, 2, …, $n - 1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan.
5. Proses *dekripsi* dilakukan dengan menggunakan persamaan $p_i = c_i^d \bmod n$, yang dalam hal ini *d* adalah kunci *dekripsi*.

Pada pertukaran kunci dengan menggunakan algoritma RSA, kunci rahasia ini di *enkripsi* terlebih dahulu bersama dengan kunci publik. Hanya penerima pesan yang diinginkan saja yang dapat *mendekripsi* kunci rahasia tersebut karena untuk *mendekripsikannya* dibutuhkan kunci privat yang dimiliki penerima pesan. Oleh karena itu, pihak lain yang mendapatkan kunci yang sudah di *enkripsi* pun tidak dapat *mendekripsikannya*. Penggunaan algoritma RSA di dalam pertukaran kunci ini digunakan dalam Encrypting File System (EFS) di sistem operasi *windows*. Proses pertukaran kunci dengan menggunanakan algoritma RSA dapat dilihat pada gambar 7.



Gambar 7: Basic RSA Key Exchange

III. PERTUKARAN KUNCI DENGAN MENGGUNAKAN ALGORITMA DIFFIE-HELLMAN

A. Algoritma Diffie-Hellman

Algoritma ini pertama kali diperkenalkan oleh Whitfield Diffie dan Martin Hellman pada tahun 1975. Mereka berdua adalah peneliti pada universitas Stanford. Mereka memperkenalkan algoritma ini untuk memberi solusi atas pertukaran informasi secara rahasia.

Algoritma ini tidak berdasarkan pada proses *enkripsi* dan *dekripsi*, melainkan lebih kepada proses matematika yang dilakukan untuk menghasilkan kunci rahasia yang dapat disebarkan secara bebas tanpa harus khawatir karena kunci rahasia tersebut hanya dapat *didekripsi* hanya oleh pengirim dan penerima pesan. Dasar dari algoritma ini adalah matematika dasar dari aljabar eksponen dan aritmatika modulus.

Langkah-langkah dalam pertukaran kunci dengan menggunakan algoritma Diffie-Hellman adalah sebagai berikut:

1. Pilih bilangan prima yang besar, ***p***, dan bilangan integer yang tidak melebihi dari nilai ***p***, ***g***, biasa disebut bilangan basis atau generator. Kedua bilangan tersebut dapat diketahui secara publik.
2. Pilih sebuah bilangan acak oleh pengirim, ***x***, bilangan ini tidak boleh diketahui oleh orang lain.
3. Pilih sebuah bilangan acak oleh penerima, ***y***, bilangan ini tidak boleh diketahui oleh orang lain.
4. Pengirim menghitung $\mathbf{A = g^x \bmod p}$. Bilangan A ini dapat diketahui secara publik.
5. Penerima menghitung $\mathbf{B = g^y \bmod p}$. Bilangan B ini dapat diketahui secara publik.
6. Lakukan pertukaran bilangan A dan B terhadap pengirim dan penerima.
7. Lalu Pengirim menghitung $\mathbf{k_a = B^x \bmod p}$.
8. Penerima menghitung $\mathbf{k_b = A^y \bmod p}$.
9. Berdasarkan hukum aljabar nilai $\mathbf{k_a}$ sama dengan $\mathbf{k_b}$ atau bisa disebut $\mathbf{k_a = k_b = k}$**.** Sehingga pengirim dan penerima tersebut mengetahui kunci rahasia tersebut "**k"**.

Bukti dari $\mathbf{k_a = k_b = k}$:

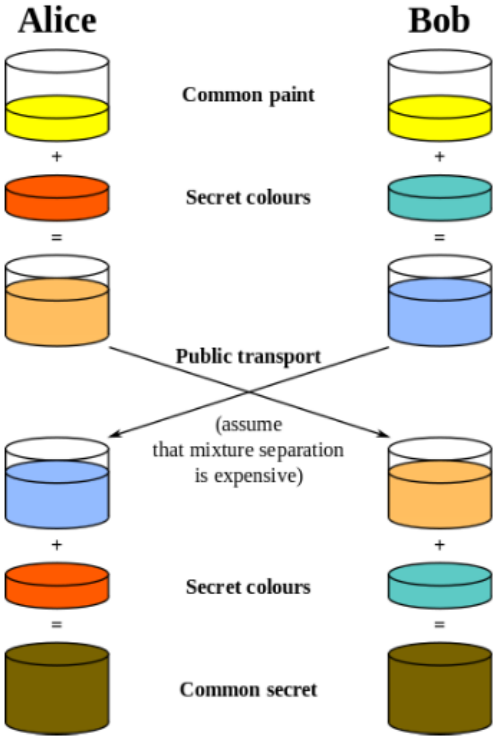
$$\mathbf{k_a = k_b}$$
$$\mathbf{B^x \bmod p = A^y \bmod p}$$
$$\mathbf{(g^y \bmod p)^x \bmod p = (g^x \bmod p)^y \bmod p}$$
$$\mathbf{(g^y)^x \bmod p = (g^x)^y \bmod p}$$
$$\mathbf{g^{yx} \bmod p = g^{xy} \bmod p}$$

Contoh penggunaan dari algoritma ini adalah:

1. Alice dan Bob menetapkan $\boldsymbol{p = 23}$ dan $\boldsymbol{g = 5}$.
2. Eve (penyadap) tahu nilai ***p*** dan ***g***.
3. Alice memilih nilai $\boldsymbol{x = 6}$ dan Bob memilih nilai $\boldsymbol{y = 15}$.
4. Alice menghitung nilai $\mathbf{A = 5^6 \bmod 23 = 8}$.
5. Bob menghitung nilai $\mathbf{B = 5^{15} \bmod 23 = 19}$.
6. Alice dan Bob bertukar nilai **A** dan **B**.
7. Eve menyadap mereka dan tahu nilai A dan B.
8. Alice melakukan perhitungan $\mathbf{k_a = 19^6 \bmod 23 = 2}$.
9. Bob melakukan perhitungan $\mathbf{k_b = 8^{15} \bmod 23 = 2}$.
10. Eve mengetahui nilai ***p*** , ***g***, **A**, dan **B** tetapi dia tidak dapat mengetahui kunci rahasia, **k** dari Bob dan Alice.

Alice dan Bob dapat mengetahui kunci rahasia tersebut dan dapat bertukar pesan dengan aman tanpa harus diketahui oleh Eve. Eve hanya dapat mengetahui nilai p, g, A, dan B tetapi tidak dapat menghitung kunci rahasia dari mereka berdua. Sehingga Eve tidak dapat mengetahui pesan rahasia apa antara Alice dan Bob.

Hal ini dapat diilustrasikan seperti pada gambar 8.



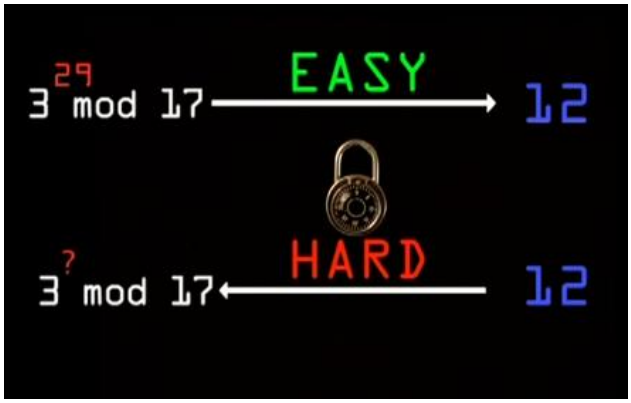
Gambar 8: Ilustrasi penggunaan algoritma Diffie-Hellman

Algoritma ini tidak hanya terbatas pada 2 pengguna saja. Jumlah pengguna yang ingin menggunakan pertukaran kunci menggunakan algoritma Diffie-Hellman ini tidak dibatasi. Hal ini hanya berlaku jika memenuhi 2 prinsip yang harus dilakukan:

1. Bilangan *p* dan *g* yang telah disetujui oleh semua anggota.
2. Setiap anggota harus melakukan pertukaran data yang diperlukan oleh anggota lainnya sehingga semua data dapat didapatkan secara merata $\mathbf{g}^{\mathbf{abc....n}}$

B. Tingkat Keamanan Algoritma Diffie-Hellman

Tingkat keamanan dari algoritma ini tinggi, jika nilai *p* dan *g* dipilih secara benar. Karena untuk mengetahui atau menebak nilai rahasia yang dimiliki oleh penerima dan pengirim harus menyelesaikan persamaan Diffie-Hellman terlebih dahulu. Ini merupakan masalah logaritma diskrit yang perhitungan tersebut tidak dapat diselesaikan untuk nilai bilangan *p* yang sangat besar. Menghitung logaritma diskrit dari bilangan modulo *p* memakan waktu yang kurang lebih sama seperti dengan memfaktorkan bilangan non prima menjadi faktor primanya, seperti yang digunakan di algoritma RSA. Oleh karena itu, algoritma ini tingkat keamanannya setingkat dengan dengan algoritma RSA. Ilustrasi dalam pengkonversian kembali persamaan Diffie-Hellman dapat dilihat pada gambar 9.



Gambar 9: Ilustrasi dalam pengkonversian kembali persamaan Diffie-Hellman

C. Perbandingan Antara Algoritma RSA dan Algoritma Diffie-Hellman

Perbandingan antar algoritma RSA dan algoritma Diffie-Hellman ini adalah dalam proses penerapannya. Algoritma RSA lebih menekankan pada proses *enkripsi* dan proses *dekripsi* dari suatu kunci asimetris tersebut, sedangkan pada algoritma Diffie-Hellman lebih pada proses penyamaran kunci yang dimiliki dan proses matematika yang dilakukan agar dapat menghasilkan kunci rahasia akhir yang sama antara pengirim dan penerima. Kedua algoritma tersebut mengandalkan kesulitan dalam pemfaktoran bilangan besar.

Kedua algoritma ini sama-sama efektif dan amannya. Hanya saja algoritma Diffie-Hellman lebih cepat dalam pemrosesannya karena memang algoritma Diffie-Hellman dikhususkan dalam pertukaran kunci ini dan mempunyai algoritma yang lebih sederhana, jika dibandingkan dengan algoritma RSA. Algoritma RSA sendiri digunakan dalam *enkripsi* dan *dekripsi*, *Digital Signature*, serta pertukaran kunci dan mempunyai algoritma yang lebih rumit sehingga jika nilai bilangan yang ditentukan besar maka proses yang dilakukan akan menjadi semakin lama.

Implementasi dari Algoritma RSA digunakan dalam Encryption File System (EFS) pada windows. Impelementasi dari algoritma Diffie-Hellman digunakan dalam Secure Shell(SSH), Internet Protocol Security (IPSec), Public Key Infrastructure (PKI), dll.

Kelemahan dari algoritma Diffie-Hellman tersebut adalah jika ada seseorang yang menyamar menjadi salah satu anggota dan terjadi pertukaran kunci tersebut. Maka, orang yang tidak diinginkan tersebut dapat mengetahui pesan rahasia tersebut. Hal ini dapat ditanggulangi dengan cara membuat kata sandi antar anggota, jika ingin melakukan suatu pertukaran kunci. Dengan melalui cara itu, orang yang tidak diinginkan tidak dapat mengetahui kunci tersebut. Sedangkan pada algoritma RSA, kelemahannya terletak pada nilai eksponen yang terlalu kecil, nilai modulus yang familiar, dan ukuran kunci yang terlalu kecil, sehingga dapat dilakukan *brute force attack.*

IV. KESIMPULAN

Penggunaan Algoritma Diffie-Hellman dalam pertukaran kunci dapat dilakukan secara aman dan efektif dalam pemrosesan jika dibandingkan dengan algoritma RSA yang cenderung lebih lama dalam pemrosesan algoritmanya. Proses pertukaran kunci ini dapat dilakukan lebih dari 2 orang asal memenuhi 2 prinsip yang telah dibahas tadi. Algoritma Diffie-Hellman lebih memfokuskan dalam perubahan nilai kunci dan proses matematis dalam penentuan kunci akhir yang sama. Sedangkan Algoritma RSA lebih memfokuskan pada saat *enkripsi* dan *dekripsi*.

Kedua algoritma tersebut memiliki tingkat keamanan yang relatif sama kuatnya dan implementasinya pun banyak digunakan di dunia keamanan jaringan. Kedua Algoritma ini sama-sama mengandalkan kesulitan pemfaktoran dalam bilangan yang bernilai sangat besar. Pertukaran kunci dengan cara yang aman dapat dilakukan dengan algoritma Diffie-Hellman dan algoritma RSA.

REFERENCES

[1] Munir, Rinaldi. *Struktur Diskrit*, edisi keempat. 2008. Bandung: Penerbit ITB.

[2] http://technet.microsoft.com/en-us/library/cc962035.aspx diakses tanggal 17 Desember 2012 jam 22.00

[3] http://www-ee.stanford.edu/~hellman/publications/24.pdf diakses tanggal 18 Desember 2012 jam 20.00

[4] http://www.sarjanaku.com/2012/11/pengertian-kriptografi-definisi.html diakses tanggal 18 Desember 2012 jam 20.15

[5] http://budi.insan.co.id/courses/ec5010/04-kriptografi.pdf diakses tanggal 18 Desember 2012 jam 20.45

[6] http://jurnal.untad.ac.id/jurnal/index.php/JIMT/article/view/135/107 diakses tanggal 18 Desember 2012 jam 21.30

[7] http://www.math.uchicago.edu/~may/VIGRE/VIGRE2009/REUPapers/Scarvalone.pdf diakses tanggal 18 Desember 2012 jam 22.30

[8] http://www.sans.org/reading_room/whitepapers/vpns/review-diffie-hellman-algorithm-secure-internet-protocols_751 diakses tanggal 18 Desember 2012 jam 23.45

PERNYATAAN

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

Bandung, 18 Desember 2012

Michael Ingga Gunawan
13511053