ISSN: 2302-3805

Implementasi Algoritma One Time Pad Pada Penyimpanan Data Berbasis Web

Hengky Mulyono [1] **, Rodiah** [2]
[1,2] *Teknik Informatika Universitas Gunadarma*
Jl. Margonda Raya No.100, Pondok Cina Depok
email : hengkymulyono301@gmail.com [1], *rodiah@staff.gunadarma.ac.id* [2]

Abstrak

Salah satu hal yang perlu diperhatikan dalam menjaga keamanan sebuah sistem adalah proses autentikasi. Proses ini dilakukan untuk memastikan bahwa user yang mengakses data maupun informasi pada sistem tersebut adalah user yang memiliki wewenang. Ada beberapa metode untuk melakukan autentikasi, salah satunya dengan menggunakan teknik penyandian data (kriptografi). Kriptografi diterapkan pada data maupun informasi dengan mengkodekan atau menyembunyikan data aslinya sehingga hanya pihak yang memiliki kunci yang dapat mengakses data atau informasi tersebut. Penelitian ini akan mengimplementasikan algoritma One Time Pad (OTP) untuk melakukan penyandian terhadap data dan informasi yang disimpan. Data atau informasi yang disimpan dalam aplikasi akan berbentuk ciphertext sehingga user akan mendapatkan kunci untuk mengakses data atau informasi tersebut. Pembuatan aplikasi ini diharapkan dapat menjaga kerahasiaan dan keamanan data dengan baik, dimana pihak yang dapat mengakses data atau informasi yang asli hanya pihak yang memiliki kunci.

Kata kunci :
ciphertext, kunci, one time pad, plaintext

1. Pendahuluan

Keamanan dalam sebuah sistem terintegrasi akan memanfaatkan jaringan komputer sebagai bagian yang sangat penting. Salah satu masalah yang dihadapi dalam keamanan pada jaringan komputer adalah bagaimana sistem dapat memastikan bahwa *user* yang mengakses data maupun informasi pada sistem tersebut adalah *user* yang benar-benar memiliki wewenang. Ada beberapa metode untuk melakukan autentikasi, salah satunya adalah menggunakan *password*, namun penggunaan *password* belum menjamin keamanan dan kerahasiaan sebuah data maupun informasi. Salah satu contoh yang paling sederhana adalah seorang admin yang memiliki wewenang untuk mengakses *database* tentu akan dengan mudah dapat mengambil data atau informasi yang disimpan didalam *database*. Kerahasiaan merupakan faktor penting untuk menjaga isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi. Salah satu cara untuk mengatasi hal ini adalah dengan teknik penyandian data atau yang lebih dikenal dengan kriptografi.

Kriptografi sendiri sudah digunakan sejak zaman dahulu hingga saat ini dimana perkembangan teknologi informasi begitu pesat. Beberapa algoritma kriptografi sengaja diciptakan dengan algoritma yang begitu rumit dengan tujuan agar pesan (data) yang dienkrip tidak mudah dipecahkan. Penelitian ini akan mengimplementasikan salah satu dari algoritma kriptografi yaitu *One Time Pad* (OTP) dimana algoritma ini menggunakan kunci (key) yang sama dalam proses enkripsi maupun deskripsi. Algoritma ini akan mengharuskan pengirim dan penerima menyetujui suatu key tertentu sebelum terjadi komunikasi diantara kedua belah pihak.

Ruang lingkup yang akan diangkat pada penelitian ini adalah implementasi algoritma OTP pada aplikasi penyimpanan data dan informasi berbasis *web* dimana data dan informasi berupa data *alphanumeric* dan *file dengan* format teks (*txt*). Proses penyandian data dilakukan disisi *web server*. Aplikasi dibuat dengan menggunakan *framework CodeIgniter 2.1.0, Jquery, HTML*, dan *CSS*.Dengan mengimplementasikan algoritma OTP pada aplikasi penyimpanan data dan informasi berbasis *web*, diharapkan keamanan data maupun informasi dapat terjaga dengan baik

2. Tinjauan Pustaka

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti rahasia dan *graphia* berarti tulisan. Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain. Teknik penyandian data (kriptografi) yang diterapkan pada data maupun informasi, dilakukan dengan mengkodekan atau menyembunyikan data aslinya. Dalam kriptografi, pesan yang akan dirahasiakan disebut *plaintext* dan pesan yang sudah diacak disebut *ciphertext* [1]. Perkembangan komputer dan sistem komunikasi pada tahun 60-an berdampak pada permintaan dari pihak-pihak tertentu sebagai sarana untuk melindungi informasi dalam bentuk digital dan untuk menyediakan layanan keamanan. Dimulai dari usaha Feistel dari IBM di awal tahun 70-an dan mencapai puncaknya pada 1977 dengan pengangkatan DES (*Data Encryption Standard*) sebagai standar pemrosesan informasi federal Amerika Serikat untuk mengenkripsi informasi yang tidak belum diklasifikasi[2].

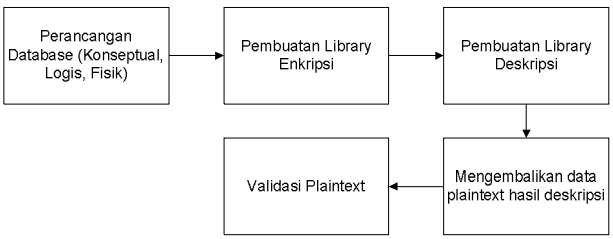
ISSN: 2302-3805

Skema enkripsi yang akan dibangun pada penelitian ini menerapkan teknik pada kriptografi modern, dimana kerahasiaan terletak pada kunci (*key*) menggunakan One Time Pad (OTP). Sampai saat ini, algoritma OTP merupakan salah satu algoritma kriptografi yang tidak dapat dipecahkan [3].

Pengembangan sejarah kriptografi juga terjadi pada 1976 saat Diffie dan Hellman mempublikasikan "*New Directions in Cryptography*". Tulisan ini memperkenalkan konsep revolusioner kriptografi kunci publik dan juga memberikan metode baru untuk pertukaran kunci, keamanan yang berdasar pada kekuatan masalah logaritma diskret. Meskipun Diffie dan Hellman tidak memiliki realisasi praktis pada ide enkripsi kunci publik saat itu, idenya sangat jelas dan menumbuhkan ketertarikan yang luas pada komunitas kriptografi. Pada 1978 Rivest, Shamir dan Adleman menemukan rancangan enkripsi kunci publik dan tanda tangan digital, yang sekarang disebut RSA. Rancangan RSA berdasar pada masalah faktorisasi yang sulit untuk kriptografi, dan menggiatkan kembali usaha untuk menemukan metode yang lebih efisien untuk pemfaktoran. Tahun 80-an menunjukkan peningkatan luas di area ini, sistem RSA masih aman. Sistem lain yang merupakan rancangan kunci publik ditemukan oleh Taher ElGamal pada tahun 1985. Rancangan inilah yang sering digunakan sampai saat ini [4].

3. Metode Penelitian

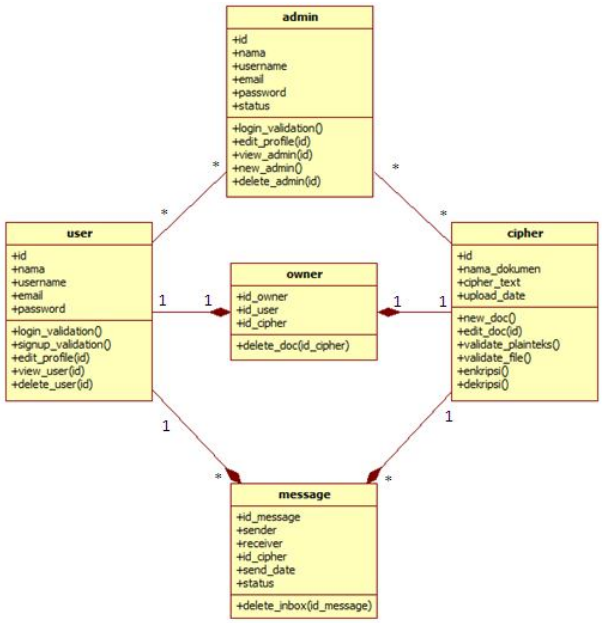
Secara umum, tahapan dari metode penelitian implementasi algoritma OTP, ditunjukkan pada gambar 1.



Gambar 1. Bagan umum implementasi OTP

Algoritma One Time Pad yang diimplementasikan pada aplikasi penyimpanan data dan informasi berbasis web terdiri atas langkah-langkah sebagai berikut :

1. Melakukan perancangan database yang terdiri atas perancangan database secara konseptual, perancangan database secara logis, dan perancangan database secara fisik. Secara konseptual, akan ditentukan entitas-entitas yang dibuat pada *database* aplikasi. Kemudian menentukan kebutuhan data-data pada tiap entitas yang dibuat, yang ditunjukkan pada class diagram gambar 2.



Gambar 2. Class Diagram Perancangan Database Konseptual

Entitas yang akan dibuat pada *database* aplikasi ini antara lain terdiri atas : Entitas *user* berisi semua data-data tentang user yang telah terdaftar pada aplikasi ini. Data-data ini didapat dari proses *signup* yang dilakukan *user* saat mendaftar pada aplikasi ini dan dapat digunakan untuk proses autentikasi *user*. Entitas *user* berisi data nama, email, *username*, dan *password user*.

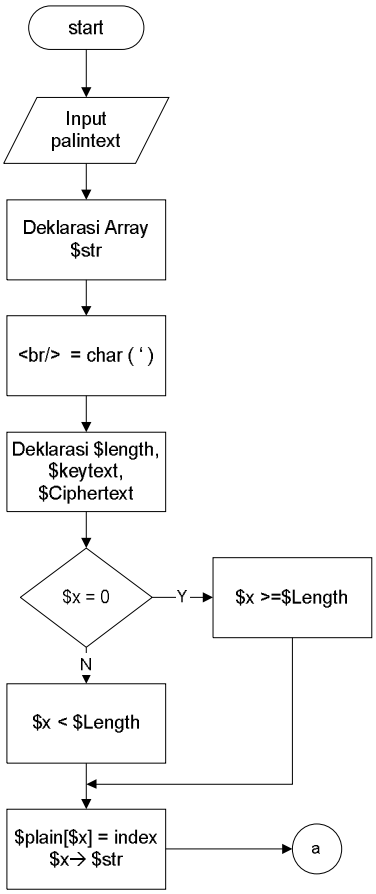
Entitas *cipher* berisi seluruh data atau informasi yang disimpan di aplikasi nama dokumen dan hasil proses enkripsi (*ciphertext*) menggunakan algoritma *One Time Pad* serta tanggal saat data atau informasi tersebut disimpan. Entitas *owner* berisi data *id user* dan *id cipher* yang digunakan untuk menunjukkan kepemilikan sebuah dokumen. Entitas *message* digunakan untuk menyimpan data atau informasi yang dikirimkan dari satu *user* ke *user* lainnya dan entitas admin digunakan untuk menyimpan data *administrator* yang berhak untuk mengelola aplikasi ini. Entitas admin berisi data nama, *email*, *username*, *password*, dan status.

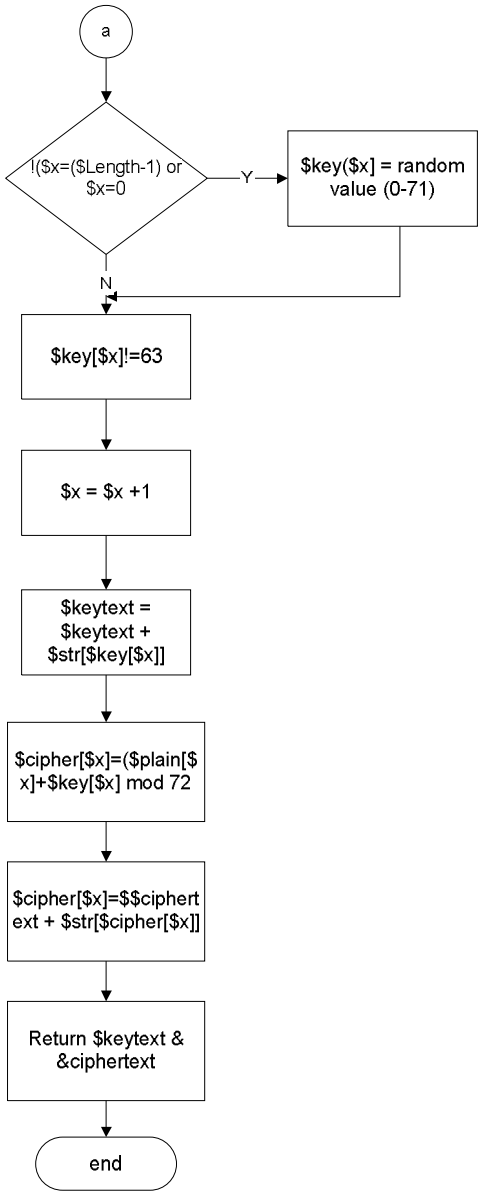
2. Membuat *Library* untuk proses Enkripsi *One Time Pad* dengan langkah-langkah sebagai berikut :
   1) Menerima data *plaintext* sebagai parameter dalam library
   2) Deklarasi *array $str* yang akan digunakan untuk menampung karakter – karakter yang diperbolehkan pada *plaintext* dan mengganti elemen html
 dengan karakter `.
   3) Mendeklarasikan variabel *$length* (menampung data panjang *plaintext*), *$keytext* (menampung data kunci dan variabel), dan *$ciphertext* (menampung data *ciphertext* hasil enkripsi)
   4) Membuat perulangan sebanyak nilai dari variabel *$length.*

ISSN: 2302-3805

5) Membuat kunci enkripsi. Kunci didapat dengan cara mengambil angka secara acak antara 0 - 71.
6) Membuat *ciphertext* dengan cara menjumlahkan indeks *plaintext* dengan kunci pada indeks yang sama lalu mod 72. Setelah *ciphertext* dan kunci didapatkan, program akan mengembalikan nilai berisi *ciphertext* hasil enkripsi dan kuncinya.

Gambar 3 menunjukkan diagram alir proses pembuatan library untuk enkripsi dengan algoritma OTP.





Gambar 3. Flowchart Enkripsi dengan OTP

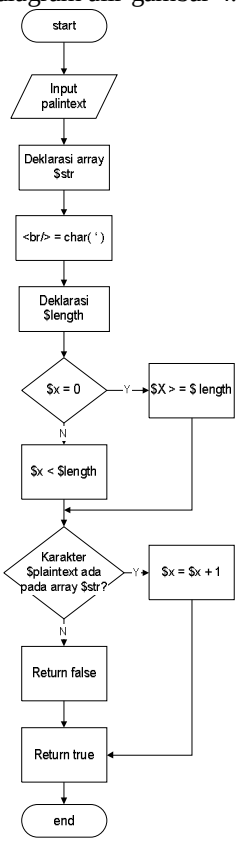
3. Membuat *library* untuk proses dekripsi pada *ciphertext* dengan langkah-langkah sebagai berikut :
   1) Menerima data *ciphertext* dan kunci sebagai parameter dalam *library* .
   2) Melakukan deklarasi *$plaintext* (menampung *plaintext*), *$c_length* (menampung data panjang *ciphertext*), dan *$k_length* (menampung data panjang kunci).
   3) Melakukan verifikasi kunci dengan membandingkan antara panjang kunci dengan panjang *ciphertext*. Pada kondisi panjang kunci dan panjang *ciphertext* tidak sama, maka kunci yang dimasukkan dinyatakan salah dan dalam hal ini *library* akan mengembalikan nilai *false*. Pada kondisi panjang kunci sama dengan panjang *ciphertext* algoritma dilanjutkan dengan memberikan nilai ke

ISSN: 2302-3805

variabel *$key[$x]* dengan *indeks* karakter *key* ke *$x* (variabel perulangan) pada *array $str* dan memberikan nilai ke variabel *$cipher[$x]* dengan *indeks* karakter *cipher*text ke *$x* pada *array $str*.

4) Menghitung indeks *plaintext* dengan melakukan perhitungan *$num = $cipher[$x] - $key[$x]* dan jika *$num* lebih besar dari 0, indeks *plaintext* dapat diperoleh dengan *$num mod* 72. Pada kondisi *$num* bernilai negatif, nilai *$num* dijadikan positif dengan dikalikan dengan -1 lalu indeks *plaintext* dapat diperoleh dari hasil pengurangan 72 dikurangi *$num*.
5) Menambahkan karakter yang ada *array $str* ke variabel *$plaintext* sesuai dengan indeks *plaintext* hasil perhitungan.
6) Mengganti karakter ` dengan \r\n untuk mencetak spasi pada *plaintext* yang akan ditampilkan.

4. Mengembalikan data berupa *plaintext* hasil dekripsi.
5. Membuat *library* validasi *plaintext*, yang dapat dilihat pada diagram alir gambar 4.



Gambar 4. Flowchart Validasi Plaintext

Berikut algoritma untuk melakukan validasi terhadap *plaintext* :

1) Melakukan pengecekan pada *plaintext* yang ingin disimpan oleh user dengan cara membandingkan setiap karakter pada *plaintext* dengan karakter-karakter yang ada di dalam *array $str*. Variabel *array $str* merupakan *array* untuk menampung karakter-karakter yang diperbolehkan pada *plaintext*.
2) Pada kondisi ditemukan karakter yang tidak terdapat pada *array $str*, maka *library* akan mengembalikan nilai *false* yang berarti *plaintext* yang ingin disimpan tidak valid.
3) Pada kondisi semua karakter pada *plaintext* juga terdapat pada *array $str* maka *library* ini akan mengembalikan nilai *true* yang berarti *plaintext* yang akan disimpan valid.

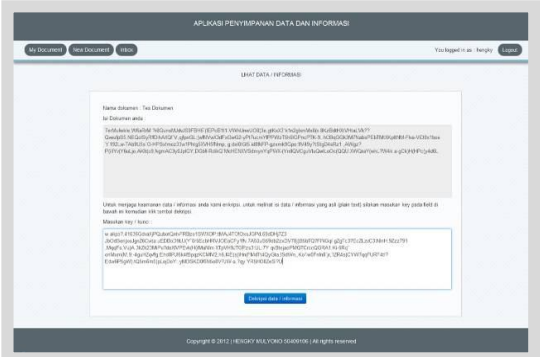
4. Hasil dan Pembahasan

4.1. Hasil

Hasil dari penelitian ini meliputi beberapa analisis dari uji coba terhadap implementasi algoritma OTP seperti pada proses autentikasi *user* dan proses *signup*. Gambar 5 menunjukkan proses autentikasi *user* jika berhasil.

Gambar 5. Tampilan jika proses Autentikasi berhasil

Untuk pemilihan judul dokumen, maka aplikasi akan menampilkan isi dokumen yang masih dalam bentuk *ciphertext* seperti pada dapat dilihat pada gambar 6.



Gambar 6. Tampilan dokumen yang berbentuk ciphertext

Pada kondisi *user* memasukkan kunci dokumen dan memilih tombol dekripsi data/informasi. Aplikasi akan mendekripsi *ciphertext* tersebut dan menampilkan hasilnya seperti ditunjukkan pada gambar 7.

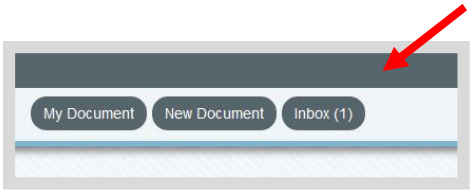
Gambar 7. Tampilan hasil dekripsi data atau informasi

ISSN: 2302-3805

Data yang telah berhasil di enkripsi akan disimpan seperti ditunjukkan pada gambar 8.

Gambar 8. Tampilan halaman simpan data hasil enkripsi

Gambar 9 adalah tampilan saat akan menyimpan data atau informasi baru dengan melalui *file* berformat .txt.



Gambar 9. Tampilan menu inbox saat ada dokumen yang dikirimkan

Pada tampilan menu *inbox* dipilih, terdapat pilihan *delete* dan kirim. Jika *user* memilih salah satu dari dokumen tersebut, aplikasi akan menampilkan isi dokumen yang masih dalam bentuk *ciphertext*. Jika *user* memilih pilihan *delete*, aplikasi akan menampilkan konfirmasi penghapusan dokumen dan jika *user* memilih pilihan kirim, aplikasi akan meminta nama *username* yang akan dijadikan penerima dokumen tersebut.

4.2. Analisis hasil uji coba

Pada aplikasi ini penulis membuat agar kunci tidak berupa karakter spasi atau *whitespace* di awal atau akhir. Jika *$x* yang dalam hal ini merupakan variabel perulangan sama dengan 0 atau *$length* – 1 akan dilakukan perulangan sampai kunci yang dihasilkan bukan merupakan spasi atau *whitespace*. Pada *array $str* karakter spasi atau *whitespace* berada pada indeks ke 63, sehingga perulangan akan terus dilakukan sampai kunci yang dihasilkan bukan bernilai 63.Validasi juga dilakukan diantaranya adalah semua *field* pada *form signup* meliputi *username* harus terdiri dari 5 - 100 karakter, *password* harus terdiri dari 7-100 karakter, dan *email* yang dimasukkan harus merupakan *email* yang valid.

Dalam proses verifikasi dilakukan pengecekan apakah OTP yang dimasukkan oleh user sama dengan OTP yang tersimpan dalam basis data atau tidak. Pada saat *user* berhasil masuk ke dalam sistem, maka akan dilakukan pencatatan nama user, tanggal, dan jam masuk untuk mengetahui siapa saja yang sudah masuk ke dalam sistem. Proses pengujian halaman *user* terdiri atas pengujian beberapa dokumen yang akan dilakukan proses enkripsi, seperti dapat dilihat pada gambar 10.

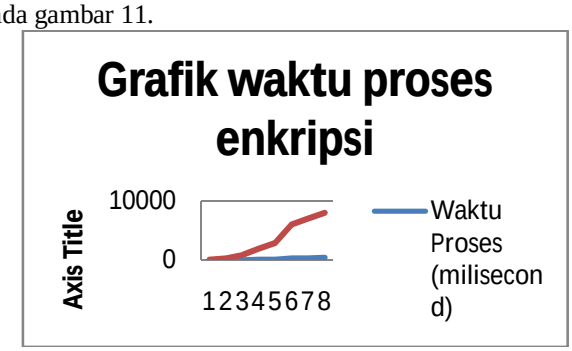
Gambar 10. Tampilan daftar dokumen pada menu my document

Tabel 1 menunjukkan hasil uji coba implementasi algoritma OTP dengan ukuran *file* sebelum proses enkripsi dengan ukuran *file* setelah proses enkripsi, dimana tidak mengalami perubahan. File ujicoba berformat .txt.

Tabel 1. Analisa Hasil Uji Coba

Waktu Proses (milisecond)	15	20	55	120	185	358	432	530
Ukuran file (byte)	200	400	990	2040	3062	6120	7125	8085

Kecepatan proses enkripsi dan dekripsi dipengaruhi dari kapasitas file dimana grafik hasil ujicoba dapat dilihat pada gambar 11.



Gambar 11. Grafik waktu proses enkripsi

5. Kesimpulan dan Saran

Berdasarkan pada analisis hasil pengujian terhadap implementasi algoritma *One Time Pad* pada aplikasi penyimpanan data dan informasi dapat di ambil kesimpulan bahwa aplikasi penyimpanan data dan informasi dengan mengimplementasikan algoritma *One Time Pad* ini dapat menjaga keamanan dan kerahasiaan data atau informasi yang tersimpan didalamnya dan dapat memastikan bahwa *user* yang mengakses data maupun informasi pada sistem tersebut adalah *user* yang benar-benar memiliki wewenang dalam hal ini adalah pihak yang memiliki kunci dari data atau informasi yang disimpan.

Untuk pengembangan aplikasi dapat dilakukan dengan variasi penyimpanan data atau informasi dengan format dokumen seperti doc, docx, odt, ppt dan format-format dokumen lainnya. Pengembangan selanjutnya juga dapat dilakukan dengan menambahkan fitur agar pengguna aplikasi dapat memperbaharui kunci dari data

ISSN: 2302-3805

atau informasi yang disimpan untuk lebih meningkatkan keamanaan dan kerahasiaan data dan informasi tersebut.

Daftar Pustaka

[1] Ariyus, Dony, 2008, "Pengantar Ilmu Kriptografi : Teori, Analisis dan Implememtasi", Penerbit ANDI OFFSET, Yogyakarta
[2] Friedrich L. Bauer, 1997, "Decrypted Secrets: Methods and Maxims of Cryptology", 1st edition, Springer, ISBN-13: 978-3540604181
[3] Jeff Connelly, 2008, "A Practical Implementation of a One-time Pad Cryptosystem", CPE 456.
[4] Sharad Patil, Ajay Kumar. 2002. Effective Secure Encryption Scheme [One Time Pad] Using Complement Approach, International Journal of Computer Science and Security (IJCSS), Volume (3) : Issue (2)

Biodata Penulis

Hengky Mulyono, saat ini sebagai mahasiswa semester 7, Jurusan Teknik Informatika Universitas Gunadarma. Saat ini sebagai asisten di Lembaga Pengembangan Komputerisasi (LEPKOM) Universitas Gunadarma

Rodiah, memperoleh gelar Sarjana Teknik Informatika (ST), Jurusan Teknik Informatika Universitas Gunadarma, lulus tahun 2003. Tahun 2006 memperoleh gelar Magister Manajemen Sistem Informasi (MMSI) dari Magister Sistem Informasi Universitas Gunadarma. Program Doktor pada Teknologi Informasi Universitas Gunadarma, lulus tahun 2012. Saat ini sebagai Staf Pengajar Teknik Informatika Universitas Gunadarma.