IMPLEMENTASI ALGORITMA AFFINE CIPHER DAN VIGENERE CIPHER UNTUK KEAMANAN LOGIN SISTEM INVENTORI TB MITA JEPARA

Yoga Religia[1]
[1]Program Studi Teknik Informatika, Fakultas Ilmu Komputer
Universitas Dian Nuswantoro Semarang
Jl. Nakula I No 5-11 Semarang 50131
Telp : (024) 3517361, Fax : (024) 3520165
Email : religia19@gmail.com[1]

Abstrak

Dalam penelitian ini dirumuskan masalah tentang bagaimana mengimplementasikan algoritma Affine cipher dan Vigenere cipher untuk keamanan login pada sistem inventori TB Mita Jepara agar dapat menjadi lebih aman dibandingkan saat menggunakan MD5. Sedangkan tujuan dari penelitian ini adalah untuk merancang keamanan login pada sistem inventori TB Mita menggunakan enkripsi Affine cipher dan Vigenere cipher, membuat enkripsi password login pada sitem inventori TB Mita menjadi lebih aman digunakan serta merancang enkripsi password login agar dapat diterapkan pada sitem inventori TB Mita menggunakan PHP. Affine cipher dan Vigenere cipher merupakan bagian dari algoritma simetris. Proses enkripsi dan proses dekripsi pada algoritma Affine cipher membutuhkan dua kunci, sedangkan proses enkripsi dan dekripsi menggunakan Vigener cipher membutuhkan satu kunci. Gabungan dari algoritma Affine cipher dan Vigenere cipher akan menghasilkan tiga kunci sehingga menjadi lebih kuat. Hasil dari keamanan password menggunakan Affine cipher dan Vigenere cipher dapat menjadi lebih kuat dibandingkan menggunakan algoritma MD5.

Kata kunci : Affine, Vigenere, MD5, sistem *login*

Abstract

In this research, it disscuses about the implemention of algorithm Affine cipher and Vigenere cipher login security of inventory system at Mita Material store Jepara in order to be more secure than using MD5. The purpose of this research is to design a login security of inventory system at Mita Material store using Affine cipher and Vigenere cipher encryption, to create the encryption of password login security in inventory system at Mita Material store more safety used and to design a login password encryption to be applied in the inventory system of Mita Material store using PHP. Affine cipher and Vigenere cipher are part of symmetric algorithm. Encryption and decryption processes of Affine cipher algorithm requires two keys, meanwhile the process of encryption and decryption using Vigener cipher requires one key. The combined of Affine cipher and Vigenere cipher algorithm will produce three keys, it becomes more powerful. The results of the security password using Affine cipher and Vigenere cipher can be more powerful than using the MD5 algorithm.

***Keywords** : Affine, Vigenere, MD5, login system*

1. LATAR BELAKANG

Hal yang penting dalam komunikasi menggunakan komputer adalah untuk menjamin kerahasian data/informasi, salah satunya dengan menggunakan enkripsi. Enkripsi adalah suatu proses untuk mengamankan suatu informasi dengan membuat informasi tersebut menjadi sebuah *ciphertext* (pesan rahasia) dengan menggunakan algoritma tertentu [1].

TB Mita adalah toko bangunan dikota Jepara yang menjual berbagai macam barang bangunan. Pelaksanaan transaksi penjualan pada toko ini sudah memanfaatkan sistem inventori yang menggunakan bahasa pemrograman PHP. Agar dapat memasuki sistem, *user* harus melakukan autentikasi dengan menginputkan *username* dan *password* terlebih dahulu. PHP merupakan bahasa pemrograman yang umum digunakan pada aplikasi *browser*. Salah satu fungsi yang digunakan oleh pemrograman PHP adalah fungsi *"session". Session* adalah sebuah variable sementara yang diletakkan dibagian server dimana PHP dapat mengambil nilai yang tersimpan diserver meskipun telah membuka halaman baru. Fungsi *session* inilah yang digunakan oleh PHP untuk melakukan autentikasi agar dapat mengatur hak akses dari suatu halaman web [2]. Pada proses autentikasi *password* pada TB Mita telah menggunakan enkripsi MD5 yang sudah disediakan oleh PHP.

Massage-Digest algorithm 5 (MD5) adalah algoritma enkripsi yang digunakan untuk membuat variable yang lebih sederhana dari data yang berukuran besar dengan memanfaatkan fungsi *hash*. Penggunaan MD5 untuk enkripsi *password* saat ini bisa dikatakan sudah tidak aman lagi [3]. Hal itu disebabkan karena sudah mulai banyak sistem peretas enkripsi dari MD5 yang dapat ditemukan dengan mudah diinternet. Sehingga diperlukan suatu metode lain yang dapat digunakan untuk enkripsi *password* agar tidak mudah diretaskan oleh pihak yang tidak bertanggung jawab.

Affine cipher merupakan sebuah enkripsi yang menggunakan sandi substitusi. Sedangkan vegenere cipher merupakan sebuah enkripsi dengan melakukan beberapa pergeseran yang direpresentasikan menggunakan satu kata kunci. Proses enkripsi dan proses dekripsi pada algoritma Affine cipher membutuhkan dua kunci, sedangkan proses enkripsi dan dekripsi menggunakan Vigener cipher membutuhkan satu kunci. Gabungan dari algoritma Affine cipher dan Vigenere cipher akan menghasilkan tiga kunci. Penggunaan tiga kunci ini mejadikan penggunaan algoritma Affine cipher dan Vigenere cipher menjadi lebih kuat dibandingkan hanya menggunakan Affine cipher saja atau Vigenere cipher saja [4], sebab selain jumlah kunci yang digunakan lebih banyak juga dibutuhkan dua kali proses dekripsi untuk mendapatkan *plaintext* yang diinginkan.

Berdasarkan pertimbangan diatas, maka diperlukan implementasi enkripsi menggunakan algoritma Affine cipher dan Vigenere cipher untuk pengamanan *password* pada sistem inventori TB Mita jepara.

2. TINJAUAN PUSTAKA

2.1. Enkripsi dan Dekripsi

Enkripsi merupakan bagian dari kriptografi yang digunakan untuk merubah suatu pesan atau informasi menjadi sandi-sandi yang bersifat rahasia. Enkripsi juga dapat diartikan sebagai cipher atau kode. Penyandian pesan atau informasi yang dilakukan menggunakan kunci, yang menjadikan pesan atau informasi tadi dapat dibaca. Tujuan dari enkripsi adalah untuk menyembunyikan pesan atau informasi dari pihak yang tidak berhak. Secara matematis, enkripsi dapat dituliskan sebagai berikut :

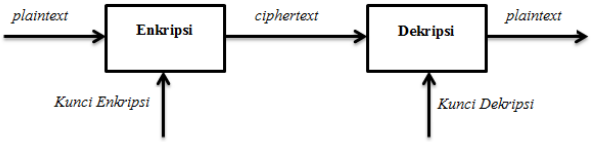
EK (M) = C (Proses Enkripsi)

Ketika proses enkripsi dilakukan, pesan *M* disandikan menggunakan kunci *K* kemudian menghasilkan pesan *C*. Pesan M dapat disebut sebagai *plaintext* sedangkan pesan *C* dapat disebut sebagai *ciphertext.*

Dekripsi merupakan kebalikan dari enkripsi. Dekripsi digunakan untuk mengembalikan sandi-sandi atau informasi yang telah diacak menjadi bentuk yang asli dengan menggunakan kunci yang sama untuk proses enkripsi. Secara matematis, dekripsi dapat dituliskan sebagai berikut :

DK (C) = M (Proses Dekripsi)

Ketika proses dekripsi dilakukan, pesan *C* yang merupakan *ciphertext* diuraikan dengan menggunakan kunci *K* sehingga menghasil- kan pesan *M* yang berupa *plaintext* [1].



Gambar 2.1. Diagram Proses Enkripsi dan Dekripsi

Gambar 2.1 menjelaskan bahwa untuk melakukan proses enkripsi diperlukan input berupa *plaintext* dan juga kunci ekripsi agar dapat menghasilkan *ciphertext.* Sedangkan untuk proses dekripsi diperlukan input berupa *ciphertext* dan juga kunci dekripsi untuk dapat menghasilkan *plaintext.*

Pada penggunaan algoritma klasik terkadang dibutuhkan sebuah tabel khusus untuk dapat melakukan perhitungan. Tabel ini digunakan untuk mengkonversi karakter yang akan dienkripsi menjadi bentuk decimal, dapat juga digunakan untuk mengkonversi bilangan decimal kedalam bentuk karakter. Adapun bentuk tabel konversi adalah sebagai berikut :

Tabel 2.1 Konversi Karakter ke Nilai Desimal

Huruf	A	B	C	D	E	F	G	H	I
Angka	0	1	2	3	4	5	6	7	8
Huruf	J	K	L	M	N	O	P	Q	R
Angka	9	10	11	12	13	14	15	16	17
Huruf	S	T	U	V	W	X	Y	Z	
Angka	18	19	20	21	22	23	24	25	

2.2. Algoritma MD5

MD5 (*Message-Digest algortihm 5*), merupakan prosedur terdefinisi atau fungsi matematika yang mengubah variabel dari suatu data yang berukuran besar menjadi lebih sederhana (fungsi *hash*) dengan menggunakan kriptografik secara luas dengan hash value 128-bit. MD5 biasanya dimanfaatkan dalam berbagai aplikasi untuk keamanan untuk meguji integritas sebuah *file*. Enkripsi menggunakan MD5 masih mendominasi sebagian besar aplikasi yang menggunakan pemrograman PHP. Banyak yang mengganggap bahwa enkripsi menggunakan MD5 sudah kuat karena enkripsi yang dihasilkannya bersifat '*one way hash*'. Padahal saat ini penggunaan MD5 untuk enkripsi *password* bisa dikatakan sudah tidak aman lagi [3]. Hal itu disebabkan karena sudah mulai banyak sistem peretas enkripsi dari MD5 yang dapat ditemukan dengan mudah diinternet.

Berapapun *string* yang di enkripsi menggunakan MD5 akan menghasilkan 32 karakter *ciphertext*. *Hash-hash* MD5 sepanjang 128-bit (16-byte), yang dikenal juga sebagai ringkasan pesan, secara tipikal ditampilkan dalam bilangan heksadesimal 32-digit. Sebagai contoh pesan ASCII sepanjang 0-byte bila dimasukkan dalam *hash* terkait MD5 akan menjadi :

MD5("") = d41d8cd98f00b204e9800998ecf8427e

Sedangkan untuk 12-byte sebagai masukan dan *hash* MD5 terkait panjang enkripsi yang dihasilkan akan sama.

MD5("enkripsi md5") = cd9611b17e651bf946a8341d692bc4e0

Ringkasan MD5 digunakan secara luas dalam dunia perangkat lunak untuk menyediakan semacam jaminan bahwa berkas yang diambil belum terdapat perubahan.

2.3. Affine Cipher

Affine cipher adalah teknik cipher yang merupakan perluasan dari Caesar cipher. Affine cipher tergolong dalam algoritma klasik yang merupakan algoritma penyandian yang sudah ada sebelum era digital sekarang ini. Algoritma klasik pada darasarnya hanya terdiri dari cipher subtitusi dan cipher tranposisi. Cipher subtitusi yaitu proses mensubtitusi karakter-karakter yang ada pada *plaintext.* Sedangkan cipher tranposisi yaitu proses pertukaran huruf-huruf yang terdapat dalam suatu *string.*

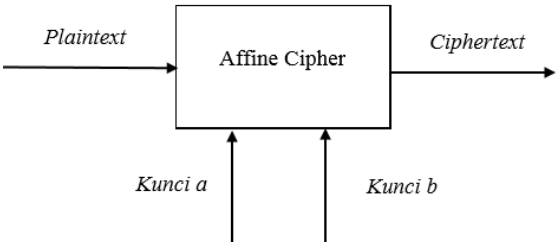
Affine cipher merupakan metode kriptografi yang menggunakan kunci simetris, yang mana kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk dekripsi [4]. Adapun terdapat dua proses dalam penggunaan Affine cipher, yaitu :

2.3.1. Proses Enkripsi Affine cipher

Proses enkripsi menggunakan Affine cipher membutuhkan 2 buah kunci yaitu kunci 1 (*a*) dan kunci 2 (*b*) untuk dapat menghasilkan *ciphertext. Plaintext* (P_i) akan dikonversikan menggunakan table konversi sehingga menjadi bentuk decimal, kemudian *ciphertext* (C_i) akan diperoleh dengan mengenkripsi *plaintext* dengan persamaan :

$$C_i = (a\ P_i + b) \bmod 26 \ldots\ldots\ldots (1)$$

C_i merupakan *ciphertext* dari pergeseran karakter yang terdapat pada *plaintext.* P_i merupakan pergeseran karakter pada *plaintext. a* merupakan kunci berupa bilangan bulat yang relatif prima dengan 26, apabila *a* tidak relatif prima dengan 26 maka dekripsi tidak akan bisa dilakukan. Sedangkan kunci *b* merupakan pergeseran nilai relatif prima dari *a.* Agar dapat memperoleh *ciphertext* maka perlu dilakukan perhitungan dengan persamaan *(1)* adapun hasil ang diperoleh masih berupa bilangan decimal, kemudian dari bilangan decimal tersebut akan dikonversi menggunakan tabel menjadi *ciphertext* yang diinginkan.



Gambar 2.2 Proses Enkripsi Affine Cipher

Gambar 2.2 menjelaskan bahwa untuk memperoleh *ciphertext* menggunakan Affine cipher dibutuhkan input berupa *plaintext* yang akan dienkripsi menggunakan dua buah kunci.

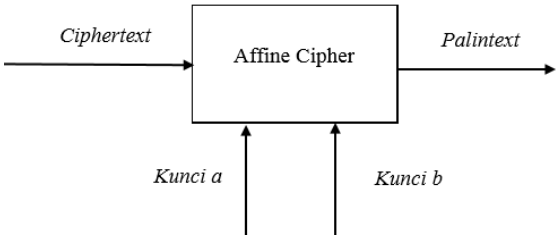
2.3.2. Proses Dekripsi Affine Cipher

Proses dekripsi menggunakan Affine cipher membutuhkan dua buah kunci yang mana kedua kunci yang dipakai haruslah sama dengan kunci yang digunnakan pada proses enkripsi. Agar dapat memperoleh *plaintext* maka kunci 1 (*a*) akan dirubah dalam bentuk invers *a (mod 26),* dinyatakan dengan a^{-1}. Jika a^{-1} ada, maka dekripsi akan dilakukan dengan persamaan

$$P_i = a^{-1}(C_i - b) \bmod 26 \ldots\ldots\ldots (2)$$

P_i merupakan *plaintext* dari pergeseran karakter yang terdapat pada *ciphertext.* C_i merupakan pergeseran karakter pada *ciphertext. a* dan *b* merupakan kunci yang sama dengan kunci yang digunakann pada proses enkripsi. Agar dapat memperoleh *plaintext* maka diperlukan perhitungan menggunakan persamaan *(2).* Sebelum melakukan perhitungan terlebih dahulu P_i dan C_i harus dikonversikan kedalam bentuk decimal menggunakan tabel konversi. Hasil dari perhitungan yang dilakukan akan berbentuk bilangan decimal yang kemudian

akan dikonversi menggunakan tabel konversi untuk memperoleh *plaintext.*



Gambar 2.3 Proses Dekripsi Affine Cipher

Gambar 2.3 menjelaskan bahwa untuk memperoleh *plaintext* menggunakan Affine cipher dibutuhkan input berupa *ciphertext* yang akan dienkripsi menggunakan dua buah kunci.

Kekuatan dari Affine cipher ini terletak pada kunci yang dipakai. Kunci ini merupakan nilai *integer* yang menunjukkan pergeseran karakter-karakter. Selain itu Affine cipher juga menggunakan barisan bilangan-bilangan yang berfungsi sebagai pengali kunci. Barisan yang digunakan dapat berupa bilangan tertentu seperti deret bilangan genap, deret bilangan ganjil, deret bilangan prima, deret *fibonaci* dapat juga deret bilangan yang dibuat sendiri. Dengan adanya kemungkinan pemilihan kunci yang dipilih lebih bervariatif dan lebih banyak algoritma enkripsi subtitusi lain menjadikan Affine cipher sebagai sistem enkripsi yang paling sempurna dibandingkan dengan algoritma enkripsi subtitusi lainnya [5].

2.4. Vigenere Cipher

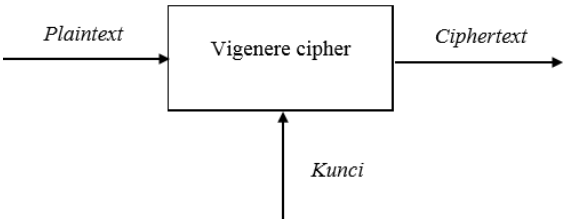
Vigenere cipher merupakan bagian dari algoritma kriptografi klasik yang sangat dikenal karena menggunakan rumus matematika, selain itu Vigenere cipher juga dapat menggunakan tabel Vigenere untuk melakukan enkripsi *plaintext* ataupun dekripsi *ciphertext.* Tabel Vigenere ini digunakann untuk memperoleh *ciphertext* berdasarkan kunci yang sudah ditentukan. Jika panjang kunci lebih pendek dari *plainext* maka maka kunci akan diulang penggunaannya secara periodik. Terdapat dua proses dalam penggunaan Vigenere cipher, yaitu :

2.4.1. Proses Enkripsi Vigenere cipher

Proses enkripsi menggunakan Vigenere cipher membutuhkan 1 buah kunci untuk dapat menghasilkan *ciphertext.* Kunci yang digunakan merupakan sebuah kata atau susunan dari beberapa huruf. Kemuadian dari kunci yang sudah ditentukan akan dikonversikan mengguakan tabel konversi sehingga menjadi bentuk desimal. Selain mengkonversi kunci yang digunakan, Vigenere cipher juga harus mengkonversi *Plaintext* (P_i) menggunakan table konversi agar menjadi bentuk desimal, kemudian *ciphertext* (C_i) akan diperoleh dengan mengenkripsi *plaintext* dengan persamaan :

$$C_i = (P_i + K_r) \bmod 26 \ldots\ldots\ldots (3)$$

C_i merupakan *ciphertext* dari pergeseran karakter yang terdapat pada *plaintext.* P_i merupakan pergeseran karakter pada *plaintext.* K_r merupakan kunci berupa hasil konversi tabel berbentuk bilangan desimal dari pergeseran karakter yang terdapat pada kunci yang digunakan. Hasil perhitungan yang dilakukan menggunakan persamaan *(3)* akan menghasilkan sebuah bilangan decimal untuk kemudian perlu dikonversi menggunakan tabel konversi untuk memperoleh *ciphertext.*



Gambar 2.4 Proses Enkripsi Vigenere

Gambar 2.4 menjelaskan bahwa untuk merubah *plaintext* menjadi *ciphertext* pada Vigenere cipher dibutuhkan input berupa *plaintext* dan sebuah kunci.

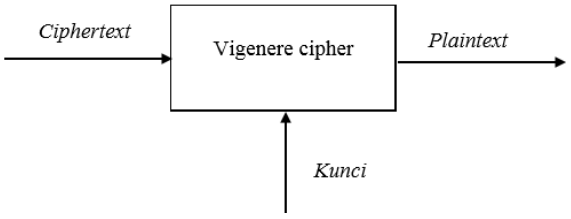
2.4.2. Proses Dekripsi Vigenere cipher

Proses dekripsi menggunakan Vigenere cipher membutuhkan 1 buah kunci untuk

dapat menghasilkan *plaintext.* Kunci yang digunakan merupakan kunci yang sama dengan kunci yang digunakan pada proses enkripsi. Kemuadian dari kunci yang sudah ada akan dikonversikan mengguakan tabel konversi sehingga menjadi bentuk desimal. Selain mengkonversi kunci yang digunakan, Vigenere cipher juga harus mengkonversi *ciphertext* (C_i) menggunakan table konversi yang juga menghasilkan bilanngan desimal, kemudian *plaintext* (P_i) akan diperoleh dengan mendekripsi *plaintext* dengan persamaan :

$$P_i = ((C_i - K_r) + 26) \bmod 26 \ldots\ldots\ldots (4)$$

P_i merupakan *plaintext* dari pergeseran karakter yang terdapat pada *ciphertext.* C_i merupakan pergeseran karakter pada *ciphertext.* K_r merupakan kunci berupa hasil konversi tabel berupa bilangan desimal dari pergeseran karakter yang terdapat pada kunci yang digunakan. Kemudian untuk mendapatkan P_i dapat dilakukan dengan terlebih dahulu mengurangi nilai C_i dengan nilai K_r hasil dari pengurangan yang sudah dilakukan akan dijumlahkan dengan angka 26 untuk kemudian hasil penjumlahan akan di modulo 26. Hasilnya akan berupa bilangan decimal, dari hasil bilangan decimal yang didapat akan dikonversi dengan tabel konfersi sehingga diperoleh karakter *plaintext* yang diinginkan.



Gambar 2.5 Proses Dekripsi Vigenere

Gambar 2.5 menjelaskan bahwa untuk merubah *ciphertext* menjadi *plaintext* pada Vigenere cipher dibutuhkan input berupa *ciphertext* dan sebuah kunci.

Vigenere cipher dikenal luas karena cara kerjanya yang mudah dimengerti dan dijalankan serta bagi para pemula akan sulit untuk dipecahkan. Pada saat kejayaannya, Vigenere cipher dijuluki sebagai *le chiffre indenchiffrable* (bahasa perancis: "sandi yang tak terpecahkan"). Metode pemecahan Vigenere cipher sendiri baru ditemukan pada abad ke-19 tepatnya ditahun 1854 oleh Charles Babbage.

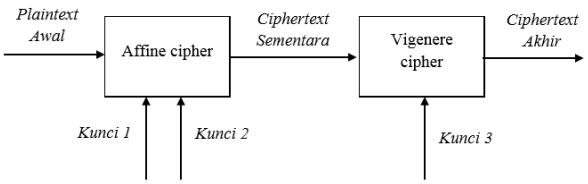
3. METODE

3.1. Analisis Sistem

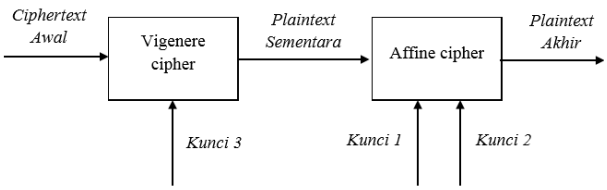
Pada proses autentikasi *password* pada TB Mita telah menggunakan enkripsi MD5 yang sudah disediakan oleh PHP. MD5 yang mulai diperkenalkan oleh seorang profesor MIT yang bernama Ronald Rivest pada sekitar tahun 1991 sempat dijadikan sebagai algoritma enkripsi standar dalam berbagai keperluan proses otentikasi. Akan tetapi pada tahun 1996, kelemahan pada MD5 mulai ditemukan. Sejak saat diketahui bahwa MD5 cenderung rentan terhadap serangan *collision*. Serangan *collision* adalah suatu peristiwa dimana dua nilai yang berbeda dapat memiliki nilai hash yang sama. Bahkan setelah tahun 2008, telah ditemukan cara untuk memanfaatkan *collision* ini untuk memalsukan sertifikat SSL yang menjadikan MD5 divonis tidak cocok untuk dipakai sebagai fungsi enkripsi yang membutuhkan ketahanan dari serangan *collision* [3]. MD5 juga tidak menggunakan kunci apapun untuk melakukan proses enkripsi, hal tersebut menjadikan proses dekripsinya tidak perlu menemukan kunci yang digunakan. Selain itu saat ini hasil dari enkripsi MD5 semakin mudah di dekripsi dengan banyaknya situs-situs diinternet yang menyediakan fasilitas dekripsi algoritma MD5.

Dengan kelemahan yang terdapat pada MD5, maka diperlukan algoritma pengganti MD5 untuk enkripsi *password* pada sistem inventori TB Mita Jepara. Affine cipher dan Vigenere cipher merupakan bagian dari algoritma

kriptografi klasik yang menggunakan kunci simetrik yang mana kunci yang digunakan untuk proses enkripsi sama dengan kunci yang digunakan untuk proses dekripsi. Penggunaan kunci pada algoritma Affine cipher dan Vigenere cipher ini menjadikan kriptanalisis membutuhkan waktu untuk menemukan kunci yang digunakan sebelum melakukan dekripsi pada *ciphertext.* Untuk proses enkripsi menggunakan algoritma Affine cipher dan Vigenere cipher akan dilakukan dengan cara mengenkripsi *plaintext* awal menggunakan Affine cipher yang akan menghasilkan *ciphertext* sementara. Selanjutnya dari *ciphertext* sementara tersebut kemudian akan dienkripsi lagi menggunkan Vigenere cipher untuk menghasilkan *ciphertext* yang akan digunakan. Sedangkan untuk dekripsi *ciphertext* maka akan didekripsikan menggunakan Vigenere cipher terlebih dahulu yang menghasilkan *plaintext* sementara. Kemudian dari *palaintext* sementara akan di dekripsikan kembali menggunakan Affine cipher yang menghasilkan *plaintext* awal (*plaintext* sebelum proses enkripsi). Proses enkrisi dan dekripsi menggunakan Affine cipher dan Vigenere cipher dapat dilihat pada gambar 3.1. dan gambar 3.2.



Gambar 3.1. Proses Enkripsi Menggunakan Affine dan Vigenere Cipher



Gambar 3.2. Proses Dekripsi Menggunakan Vigenere dan Affine Cipher

3.2. Desain Sistem

3.2.1. Pembuatan Database

Dalam pembuatan database akan disediakan sebuah tabel khusus untuk otentikasi *login*, yang mana dalam tabel tersebut akan digunakan tiga *field* yaitu terdiri dari :

a. *Field* "id" dengan tipe *integer. Field* "id" ini akan digunakan untuk mengecek posisi data yang dituju agar menjadi lebih mudah. Karena field "id" ini bersifat *auto increment* sehingga antara satu id dengan id yang lain tidak aka sama.
b. *Field* "username" dengan tipe *varchar*. *Field* ini digunakan untuk memberikan inisial pengguna pada sistem inventori TB Mita Jepara.
c. *Field* "password" dengan tipe data *varchar. Field* "password"akan digunakan sebagai tepat menyimpan kunci dari tiap *username.* Pada *field* "password" inilah yang isi *field*-nya akan dienkripsi menggunakan algoritma Affine cipher dan Vigenere cipher.

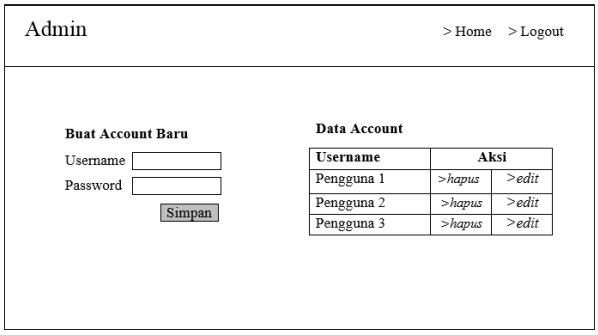
Adapun tampilan tabel yang akan dibuat akan seperti ini :

Id	Username	Password
1	Admin	ZMZXJGSKJL
2	Karyawan	YBHJKGQKHLP

Gambar 3.3. Desain Tabel Login

3.2.2 Pembuatan Account

Pembuatan *account* ini dibutuhkan untuk menambahkan, merubah ataupun mengurangi (menghapus) *account*. Untuk menambahkan *account* akan disediakan *form* pembuatan *account* baru. Sedangkan untuk merubah atau menghapus *account* akan disediakan tombol "Edit" dan tombol "Hapus". Adapun desain pembuatan *account* adalah sebagai berikut :



Gambar 3.4. Desain Halaman Pembuatan Account

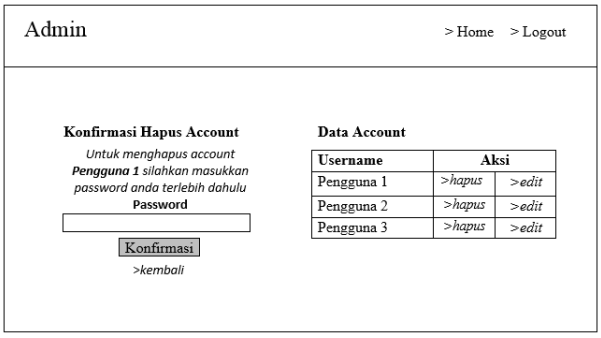
Dari gambar 3.4 menunjukkan tiga bagian utama yang terdapat dalam halaman pembuatan *account* yaitu :

- Bagian 1 : merupakan *header* dari menu Admin dari sistem inventori TB Mita. Dalam header ini terdapat dua pilihan menu yaitu menu *logout* yang digunakan untuk keluar dari sistem inventori TB Mita dan menu *home* yang digunakan untuk kembali ke beranda sistem inventori TB Mita
- Bagian 2 : merupakan *form* pembuatan *account* baru dengan memasukkan *username* dan *password* baru, adapun jumlah karakter pada *password* dibatasi sebanyak 4 sampai 20 karakter saja. setelah itu menekan tombol "simpan". Pada saat tombol "simpan" ditekan maka akan dimintai konfirmasi *password* pembuat *account.* Hal ini dilakukan agar hanya orang yang memiliki hak saja yang dapat membuat *account* baru.
- Bagian 3 : merupakan data *account* yang sudah ada atau sudah terdaftar dalam sistem

Pada bagian data *account* terdapat dua fungsi tambahan yaitu fungsi hapus *account* yang digunakan untuk menghapus *account* pengguna dan juga fungsi edit *account* yang digunakan untuk mengedit data *account* pengguna.

3.2.3. Pembuatan Fungsi Hapus Account

Fungsi hapus *account* dapat digunakan untuk mengurangi *account* yang sudah tidak aktif atau sudah tidak digunakan lagi agar tidak terjadi kepemilikan *account* ganda. Selain itu dapat juga digunakan apabila ada salah satu *account* (karyawan) yang hak akses nya dicabut oleh pemilik *account* utama (pemilik toko). Adapun tampilan desain dari fungsi hapus *account* adalah sebagi berikut :

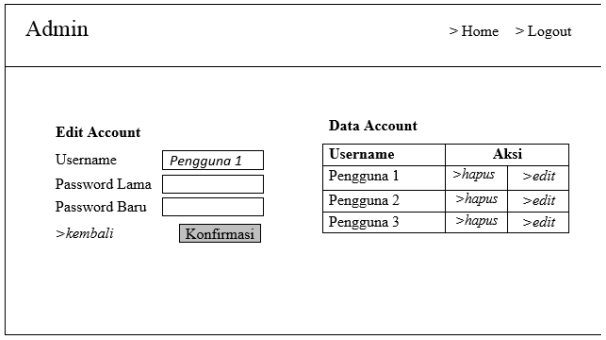


Gambar 3.5. Desain Halaman Hapus Account

Gambar 3.5. menunjukkan sebuah tampilan yang akan muncul ketika fungsi hapus pada kolom aksi ditekan. Dalam tampilan tersebut hanya yang memiliki *account* saja yang dapat menghapus *account* yaitu dengan cara memasukkan *password* terlebih dahulu kemudian menekenan tombol "konfirmasi". Sedangkan untuk membatalkan penghapusan dapat ditekan tombol "kembali".

3.2.4. Pembuatan Fungsi Edit Account

Fungsi edit *account* dapat digunakan untuk memperbarui *username* atau *password* dari sebuah *account*. Halaman edit *account* akan muncul ketika fungsi edit pada kolom aksi ditekan. Supaya dapat mengganti *username* atau *password* diperlukan masukan berupa *password* lama terlebih dahulu kemudian dapat dilanjutkan dengan menekan tombol "Konfirmasi", sedangkan untuk membatalkannya dapat dengan menekan tombol "kembali". Adapun desain dari halaman edit *account* akan ditunjukkan pada gambar 3.6 sebagai berikut:



Gambar 3.6. Desain Halaman Edit Account

3.2.5. Pembuatan Sistem Login

Sistem *login* ini dibutuhkan untuk membatasi pengguna dari sistem inventori TB Mita Jepara. Hanya pengguna yang memiliki *username* dan *password* yang *valid* saja yang dapat menggunakan sistem inventori. Adapun desain dari halaman sistem login adalah sebagai berikut :

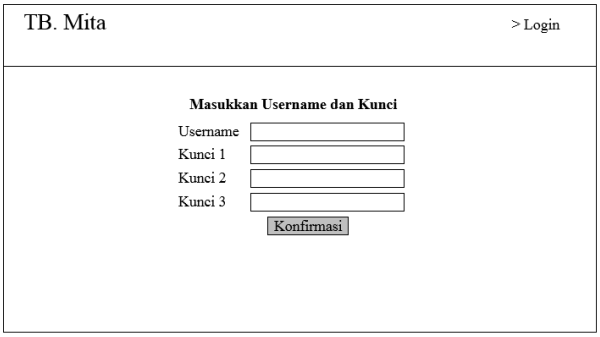
Gambar 3.7. Desain Halaman Login

Pada gambar 3.7 memiliki 3 bagian utama didalamnya. Adapun tiap bagiannya adalah sebagai berikut :

- Bagian 1 : merupakan *header* dari sistem inventori TB Mita Jepara
- Bagian 2 : merupakan bagain *form login* yang harus diisi untuk dapat masuk kedalam sistem inventori TB Mita Jepara.
- Bagian 3 : merupakan fungsi yang digunakan apabila pengguna lupa dengan *password* yang dimiliki.

3.2.6. Fungsi Lupa Password

Fungsi lupa *password* digunakan untuk kondisi dimana pengguna lupa dengan *password* yang dimiliki. Untuk mendapatkan *password* pengguna diharuskan memasukkan *username* dan juga tiga kunci dekripsi. Adapun tampilan pada halaman lupa *password* adalah sebagai berikut :



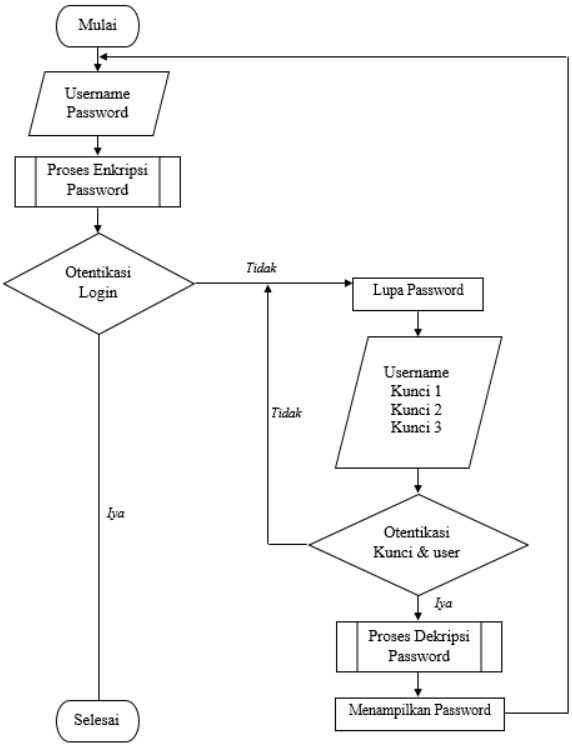
Gambar 3.8. Desain Halaman Lupa Password

Pada gambar 3.8 terdapat dua bagian utama, bagian-bagian tersebut yaitu :

- Bagian 1 : merupakan *header* dari sistem inventori TB Mita Jepara, didalam *header* ini terdapat menu *login* yang dapat digunakan untuk kembali pada sistem *login.*
- Bagian 2 : merupakan *form* yang harus diisi apabila lupa *password* agar dapat memperoleh *password*.

3.3. Implementasi

Penggunaan enkripsi dan dekripsi password menggunakan Affine cipher dan Vigenere cipher dibagian *login* mempunyai alur urutan yang dapat dilihat pada gambar 3.9.



Gambar 3.9. *Flowchart Proses Login*

Pada gambar 3.9 memperlihatkan alur proses yaitu untuk melakukan *login* dibutuhkan input berupa *username* dan *password,* kemudian *password* yang telah diinput akan dienkripsi terlebih dahulu sebelum melakukan otentikasi *login.* Bila hasil otentikasi *login* benar maka pengguna dapat langsung memasuki sistem, sedangkan apabila salah maka pengguna dapat mengulang *login* atau memanfaatkan fasilitas lupa *password.* Pada fasilitas lupa *password* penggua diharuskan mengisi *username* dan juga tiga kunci yang digunakan untuk proses enkripsi yang mana ketiga kunci ini hanya diketahui oleh admin. Apabila *username* tersedia dan juga kunci yang diberikan salah maka pengguna dapat mencoba untuk mengisi kembali *username* dan juga tiga kunci, tetapi apabila *username* tersedia dan juga kunci yang diberikan benar maka akan dilakukan proses dekripsi *password* sesuai dengan *password* dari *username* yang dimasukkan untuk kemudian memberikan *password* kepada pengguna untuk melakukan proses *login.*

4. PEMBAHASAN

4.1. Langkah Pengujian

Langkah-langkah yang akan dilakukan dalam menguji keamanan *password* mengggunakan MD5 pada TB Mita Jepara adalah:

1. Membuka database dan tabel penyimpanan *password*
2. Menyalin *password* MD5 pada *field password*
3. Membuka *search engine.*
4. Ketikkan pada *form* pencarian dengan kata kunci “MD5” atau “MD5 decrypt tool” .
5. Pilih alamat yang akan digunakan.
6. Masukkan *password* MD5 yang sudah disalin dari tahap ke-2 pada *form decryption* (dekripsi)
7. Kemudian tekan tombol dekripsi yang tersedia

Sedangkan langkah-langkah yang akan dilakukan dalam menguji keamanan *password* mengggunakan Affine cipher dan Vigenere cipher pada TB Mita Jepara adalah:

1. Membuka database dan tabel penyimpanan *password*
2. Menyalin *password* Affine cipher dan Vigenere cipher pada *field password*
3. Membuka *search engine.*
4. Ketikkan pada *form* pencarian dengan kata kunci “Affine cipher dan Vigenere cipher” atau “Affine cipher dan Vigenere cipher decrypt tool” .
5. Pilih alamat yang akan digunakan.
6. Masukkan *password* Affine cipher dan Vigenere cipher yang sudah disalin dari tahap ke-2 pada *form decryption* (dekripsi)
7. Kemudian tekan tombol dekripsi yang tersedia

4.2. Analisis Hasil Pembahasan

Berdasarkan pengujian keamanan algoritma MD5 dibandingkan dengan Affine cipher dan Vigenere cipher yang telah dilakukan menghasilkan :

a) Pencarian sistem kriptanalis untuk algoritma MD5 menggunakan *search engine* dapat dengan mudah ditemukan.
b) Untuk mendekripsikan *ciphertext* MD5 tidak membutuhkan kunci tertentu.
c) Pencarian sistem kriptanalis untuk Affine cipher dan Vigenere cipher menggunakan *search engine* tidak dapat ditemukan. Hanya dapat menemukan sistem kriptanalis untuk Affine cipher saja atau Vigenere cipher saja.
d) Untuk mendekripsikan *ciphertext* Affine cipher dan Vigenere cipher membutuhkan 3 kunci.
e) Mendekripsi *ciphertext* Affine cipher dan Vigenere cipher menggunakan sistem kriptanalis Affine cipher tidak dapat memperoleh *plaintext*.
f) Mendekripsi *ciphertext* Affine cipher dan Vigenere cipher menggunakan sistem kriptanalis Vigenere cipher tidak dapat memperoleh *plaintext*.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan pembahasan dan hasil evaluasi dari bab-bab sebelumnya, maka dapat diambil kesimpulan bahwa tujuan dibuatkannya laporan ini adalah untuk merancang keamanan *login* sistem inventori TB Mita menggunakan enkripsi Affine cipher dan Vigenere cipher agar *login* sistem inventori TB Mita menjadi lebih aman digunakan. Sehingga menghasilkan keamanan *login* pada sistem inventori TB Mita menggunakan Affine cipher dan Vigenere cipher yang lebih aman dibandingkan sistem *login* mengguakan algoritma MD5.

5.2. Saran

Dalam penerapan enkripsi menggunakan algoritma Affine cipher dan Vigenere cipher pada sistem login TB Mita Jepara, terdapat beberapa hal yang perlu diperhatikan supaya menjadi lebih baik kedepannya, diantaranya sebagai berikut :

1) Penerapan algoritma Affine cipher dan Vigeere cipher untuk keamanan *login* sistem ini dapat dijadikan sebagai referensi untuk dikembangkan menjadi keamanan *login* sistem yang lebih baik.
2) Penerapan algoritma Affine cipher dan Vigeere cipher untuk keamanan *login* sistem ini dapat dilakukan modifikasi pada algoritma yang digunakan.
3) Penerapan algoritma Affine cipher dan Vigeere cipher tidak hanya untuk keamanan *login* sistem saja tetapi juga pada database sistem.

DAFTAR PUSTAKA

[1] A. Septiarini and Hamdani, "Sistem Kriptografi Untuk Text Message," *Informatika Mulawarman*, vol. 6, no. 1, 2011.

[2] A. Septi and Fauziah, "Hijacking Session Pada Sistem Keamanan Komputer," vol. 3, 2009.

[3] S. Dewantono, "Kelemahan Fungsi Massage Digest 5," *Makalah IF2091*, 2011.

[4] Juliadi and dkk, "Kriptografi Klasik Dengan Metode Modifikasi Affine Cipher Yang Diperkuat Dengan Vigenere Cipher," *Buletin Ilmiah Matematika Statistik*, vol. 2, no. 2, pp. 87 - 92, 2013.

[5] Hartini and S. Primaini, "Kriptografi Password Menggunakan Modifikasi Metode Affine Cipher," vol. 2, no. 1, 2014.