APLIKASI ENKRIPSI DAN DEKRIPSI

1. **PENDAHULUAN**

Kriptografi telah menjadi bagian penting dalam dunia teknologi informasi saat ini. Hampir semua penerapan teknologi informasi menggunakan kriptografi sebagai alat untuk menjamin keamanan dan kerahasiaan informasi. Karena itu pulalah, kriptografi menjadi ilmu yang berkembang pesat. Dalam waktu singkat, amat banyak bermunculan algoritma-algoritma baru yang dianggap lebih unggul daripada pendahulunya. Namun, tetap saja cipher yang digunakan tidak lepas dari penemuan lama. Algoritma kunci simetri termasuk algoritma yang masih sering digunakan dalam pembuatan algoritma kriptografi. Pada saat ini, algoritma enkripsi kunci simetri yang banyak digunakan adalah algoritma blok, yang beroperasi pada suatu potongan pesan (blok) yang berukuran sama (biasanya 64 bit) pada  suatu saat.

Selain algoritma blok, ada juga algoritma aliran yang beroperasi pada potongan data yang bervariasi. Dibandingkan dengan cipher aliran, baik dalam desain dan penerapan, cipher blok dianggap lebih rumit.

Pada saat Blowfish dirancang, diharapkan mempunyai kriteria perancangan sebagai berikut :

a.    Cepat, Blowfish melakukan enkripsi data pada  microprocessors 32-bit. dengan rate 26 clock cycles per byte.

b.    Compact, Blowfish dapat dijalankan pada memori kurang dari 5K.

c.    Sederhana, Blowfish hanya menggunakan operasi-operasi Sederhana, Blowfish hanya menggunakan operasi-operasi sederhana: penambahan, XOR, dan lookup tabel pada operan 32-bit.

d.   Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh Blowfish dapat bervariasi dan bisa sampai sepanjang 448 bit [2].

Proposal ini bertujuan dapat mempelajari algoritma Blowfish secara keseluruhan agar dapat memahami cara kerja dan struktur algoritmanya, kemudian menerapkan strategi perancangan Blowfish sehingga algoritma ini dapat berjalan secara optimal. Atau paling tidak dapat melakukan enkripsi dengan baik pada aplikasi yang akan dibuat.

Aplikasi yang akan dibuat diharapkan dapat menggambarkan penggunaan algoritma Blowfish pada proses enkripsi teks sehingga lebih mudah memahami cara kerja dan struktur algoritma Blowfish.

**2. DASAR TEORI**

PHP merupakan bahasa pemrograman yang dapat Anda gunakan untuk membuat skrip yang lebih interaktif. Skrip ini kemudian akan diolah dalam web server yang hasilnya dapat dilihat dalam bentuk HTML. PHP memungkinkan Anda membuat situs yang lebih interaktif dan lebih mudah untuk dioperasikan. MySQL merupakan bahasa pemrograman database terbuka. Yang mana memungkinkan Anda untuk bisa membuat, merubah, dan mengakses beberapa database sekaligus dalam server Anda. Kombinasi keduanya biasanya digunakan untuk membuat toko daring, forum, permainan, dll.

PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (Form Interpreted), yang wujudnya berupa sekumpulan script yang digunakan untuk mengolah data form dari web.

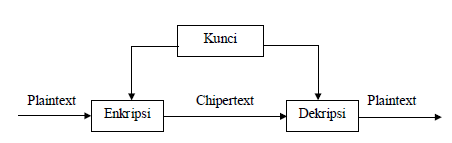
Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI, kependekan dari *Hypertext Preprocessing'/Form Interpreter*. Dengan perilisan kode sumber ini menjadi *open source*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0.

1. **FLOWCHART SISTEM**

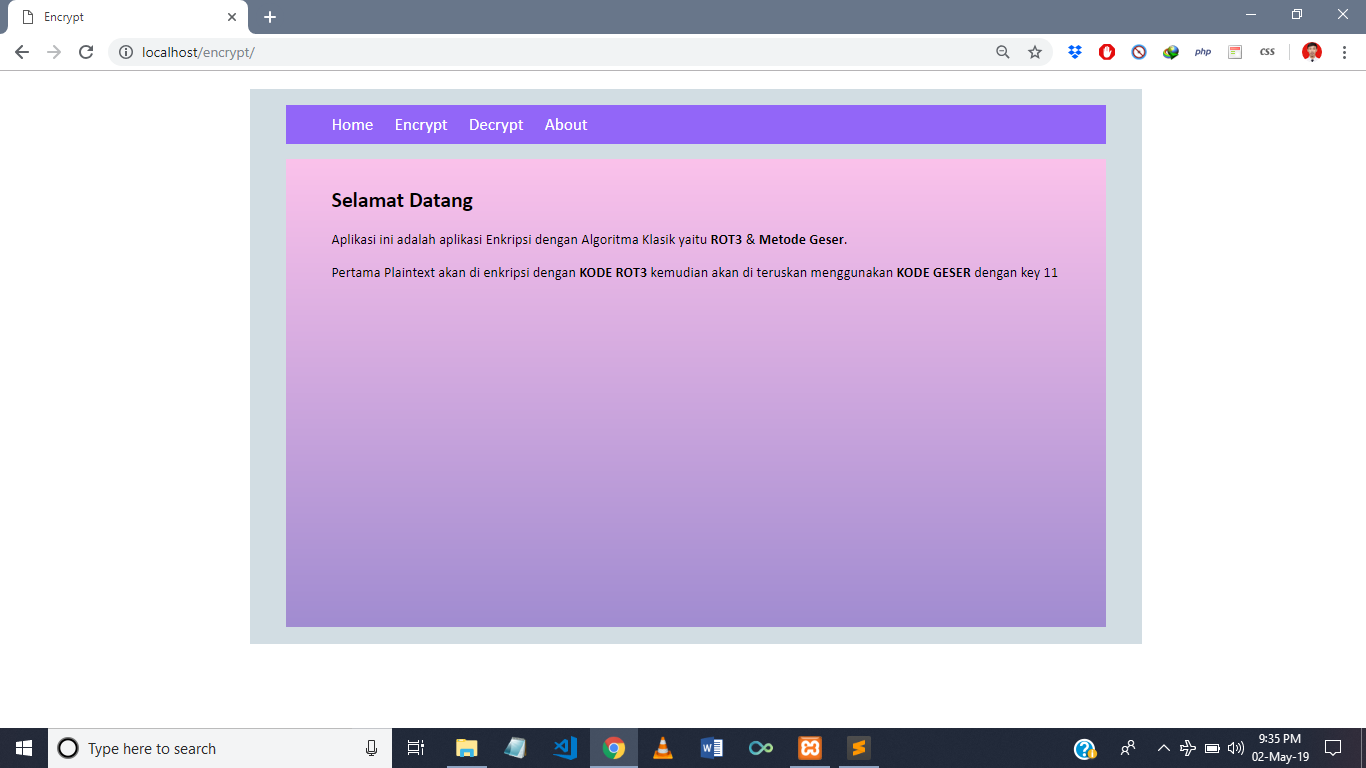
Berikut adalah flowchart sistem enkripsi dan dekripsi php



1. **IMPLEMENTASI**

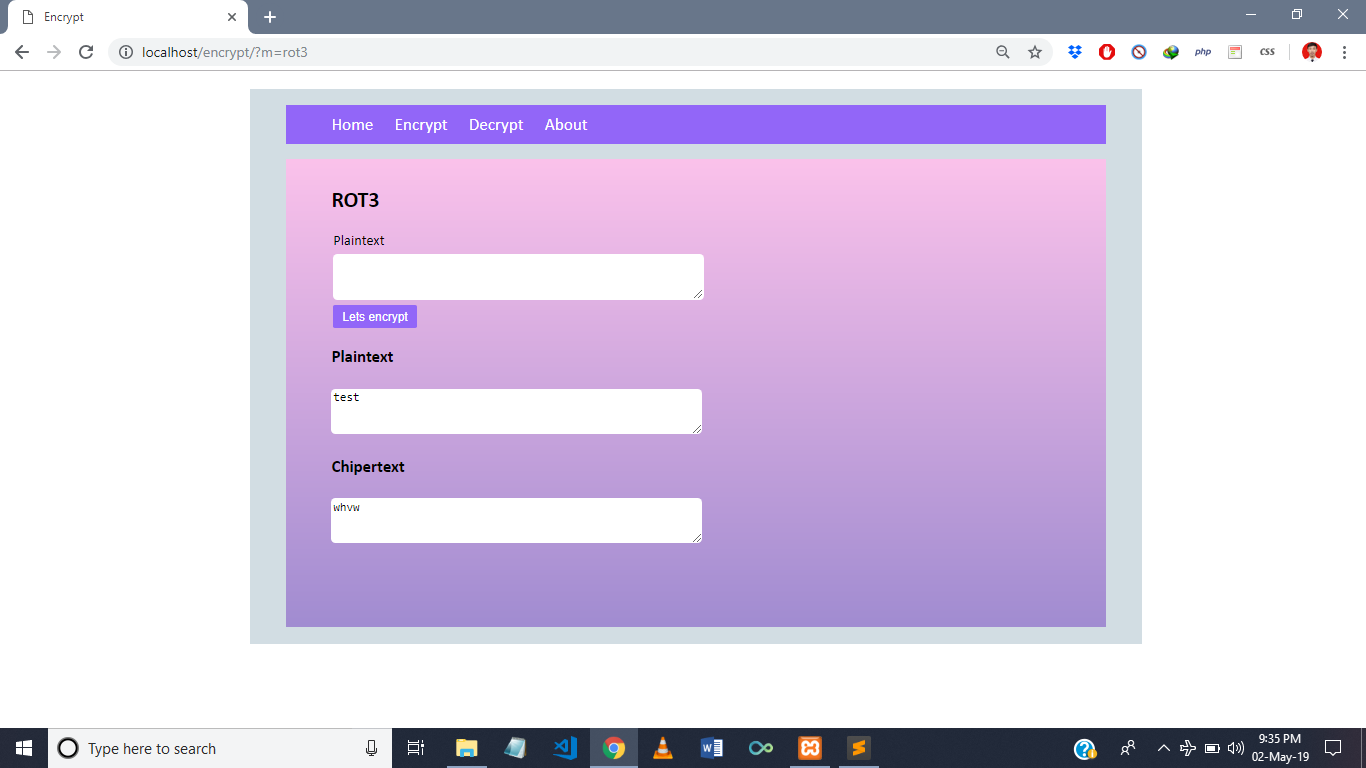
4.1 Enkripsi

Ini adalah halaman awal aplikasi enkripsi dan dekripsinya.



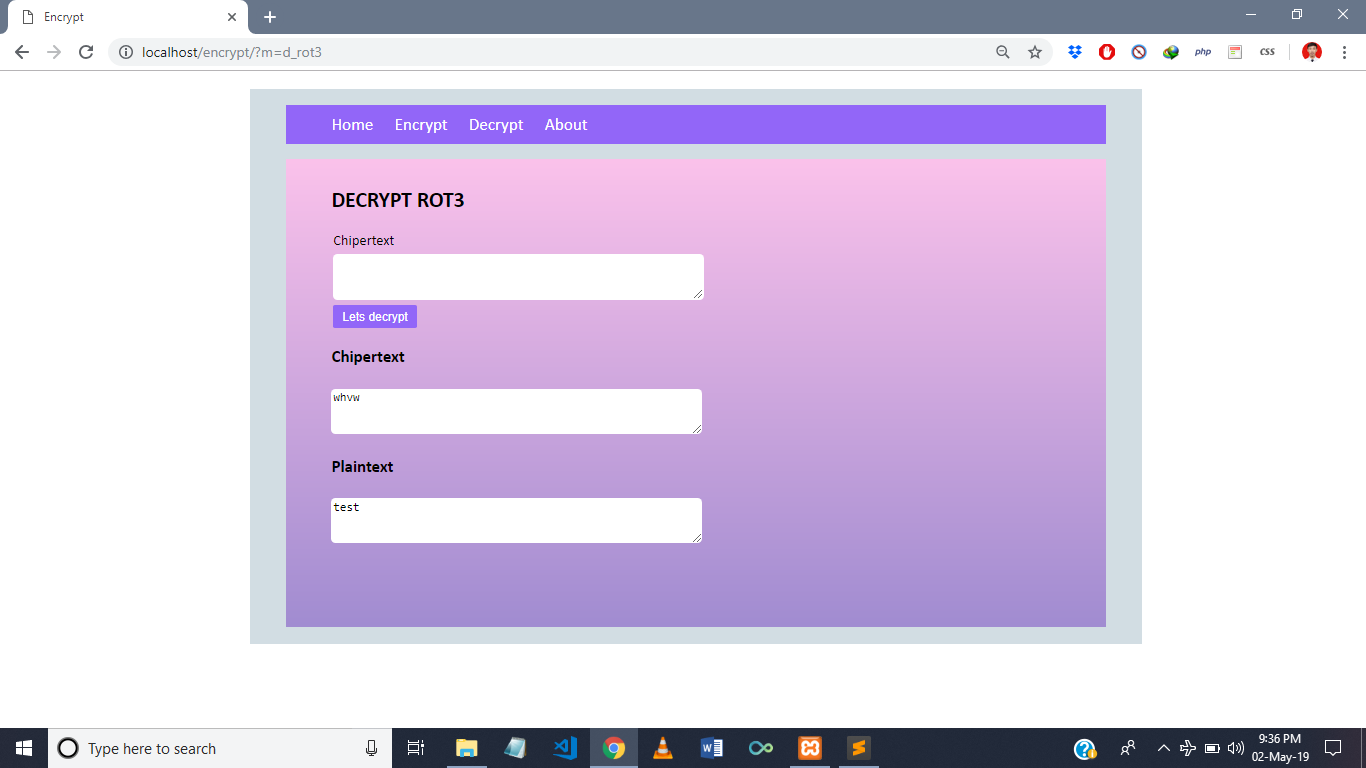
Tampilan awal aplikasi

Langkah awal untuk melakukan enkripsi adalah masuk ke halaman Encrypt dan asukkan text ke kolom input yang sudah disediakan. Setelah mengisi kolom input langkah selanjutnya adalah dengan menekan tombol lets encrypt. Secara otomatis text yang tadi diinputkan akan masuk lalu langsung diproses atau di encrypt menggunakan metode geser. Berikut tampilan nya.



4.2 Dekripsi

Selanjutnya adalah proses dekripsi, proses dekripsi disini mudah, yaitu dengan cara meng-copy text yang tadi sudah di enkripsi lalu letakkan atau tempel text yang tadi di kolom yang sudah disediakan. Tahap terakhir adalah dengan menekan tombol lets decrypt, maka secara otomatis php akan melakukan proses dekripsi.



**5. PENUTUP**

**5.1 Kesimpulan**

Dari proses diatas dapat disimpulkan bahwa aplikasi ini telah berhasil dibangun dan dapat berfungsi sesuai tujuan, yaitu mengamankan data ataupun informasi yang berupa text (plainteks) dengan mengacak text tersebut sehingga tidak dapat dibaca atau dimengerti. Aplikasi ini bisa mengubah sekaligus mengembalikan data yang telah di acak.

**6. DAFTAR PUSTAKA**

* A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
* <http://www.petanikode.com/2015/09/6-fungsi-enkripsi-di-php-untuk.html>
* B. Schneier, *Applied Cryptography - Protocol, Algorithm, and Source Code in C*, second edition, John Willey & Sons, 1996.
* http://taufikys.blogspot.co.id/2014/01/kriptografi-pada-sistem-basis-data.html

Nama : Anggih Septiawan

NIM : 170101002