

# **APLIKASI KRIPTOGRAFI METODE CAESAR CHIPER DAN MODIFIKASI DENGAN METODE AES 256 BERBASIS WEB**



Disusun oleh :

170101010 - Harun Al Rosyid

Teknik Informatika  
Pradita Institute  
2019

# APLIKASI KRIPTOGRAFI METODE CAESAR CHIPER DAN MODIFIKASI DENGAN METODE AES 256 BERBASIS WEB

Harun Al Rosyid

Teknik Informatika  
Pradita Institute  
2019

## Abstrak

Keamanan data adalah bagian yang sangat penting dalam komputer. Salah satu tindakan yang bisa kita lakukan adalah menerapkan kriptografi dalam penggunaan sandi dan menjaga file berharga kita. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi. Pada Paper ini akan membangun aplikasi kriptografi dengan metode Caesar chiper ROT3 yang kemudian dimodifikasi dengan metode Advanced Encryption Standard 256 (AES 256) untuk membandingkan tingkat keamanan hasil penyamaran sandi. Pada metode Caesar chiper ROT3 menggunakan sistem yang sederhana dengan menggunakan abjad tunggal yang digeser 3 huruf  $A=D$ . Sedangkan metode AES 256 tingkat kerumitan sandi bertambah. AES 256 memiliki blok kunci 256 bit Berdasarkan ukuran block 256, AES bekerja pada matriks berukuran  $4 \times 4$  di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit). Berdasarkan analisa keamanan diperoleh bahwa metode AES 256 lebih aman dari Caesar Cipher ROT3 karena ciphertext AES 256 menampilkan hasil yang lebih rumit dibanding Caesar cipher rot 3.

**Kata kunci:** Kriptografi, Caesar chiper Rot3, Advance Encryption Standard (AES) 256

## I. PENDAHULUAN

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu data, pesan dan informasi. Data menjadi hal vital di masa ini, terkait betapa pentingnya pihak atau orang berkepentingan yang dapat mengakses data tersebut. Apabila ada pihak yang tak berkepentingan mengakses data tersebut, maka dikhawatirkan akan terjadi hal yang tidak diinginkan. Sejak lahirnya konsep *open system*, semua data dapat mengalir bebas melewati jaringan komputer. Namun, hal ini menjadi resiko tersendiri bagi pengguna karena data tersebut dapat diakses oleh pihak yang tidak berkepentingan. Berbagai cara dilakukan untuk mendapatkan data dan informasi, mulai dari tingkatan yang mudah sampai pada cara-cara yang rumit (Andri M 2009). Salah satu cara untuk mengamankan data dari tindakan kejahatan adalah menggunakan konsep kriptografi.

Kriptografi merupakan seni atau ilmu untuk menjaga keamanan data. Konsep kriptografi bermula dari zaman tradisional hingga modern. Secara umum ada dua jenis kriptografi, yaitu tradisional/klasik dan modern. Kriptografi tradisional adalah suatu algoritma yang menggunakan satu kunci untuk mengamankan data. Dua teknik dasar yang biasa digunakan adalah substitusi dan transposisi (permutasi). Salah satu metode kriptografi tradisional adalah *Caesar Cipher* yang hanya mampu mengamankan data karakter a-z dan A-Z saja (Suriski dkk, 2010). Seiring berkembangnya data, munculah berbagai macam metode kriptografi modern yang dapat mengamankan semua data karakter. Kriptografi modern adalah algoritma yang lebih kompleks daripada kriptografi tradisional, hal ini disebabkan algoritma ini menggunakan komputer. Terdapat 3 algoritma pada kriptografi modern (Rachman 2010), yaitu:

1. Algoritma simetris adalah algoritma yang menggunakan kunci yang sama untuk enkripsi

dan dekripsinya. Aplikasinya digunakan oleh algoritma Data Encryption Standard (DES), Advance Encryption Standard (AES), International Data Encryption Algorithm (IDEA), A5, RC4

2. Algoritma Asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan satu lagi untuk dekripsi. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA

3. Algoritma hibrida adalah algoritma yang memanfaatkan dua tingkatan kunci, yaitu kunci rahasia (simetri) yang disebut juga *session key* (kunci sesi) untuk enkripsi data dan pasangan kunci rahasia kunci publik untuk pemberian tanda tangan digital serta melindungi kunci simetri. Algoritma kriptografi yang beroperasi dalam mode bit dapat dikelompokkan menjadi dua kategori: *Cipher* aliran (*stream cipher*) dan *Cipher* blok (*block cipher*) (Rachman 2010).

Metode kriptografi tradisional dan metode kriptografi modern memiliki perbedaan dalam hal pengamanan data mulai dari proses hingga hasil output yang dihasilkan. Untuk itu di penelitian ini dibangun aplikasi kriptografi dengan metode kriptografi tradisional dan metode kriptografi modern. Dalam penelitian akan dilakukan perbandingan implementasi keamanan data dari setiap metode dari sisi efisiensi waktu dan ukuran. Proses enkripsi dan dekripsi hanya dapat dilakukan untuk file teks pada masing-masing metode. Untuk metode *Caesar Cipher*, proses enkripsi/dekripsi file teks hanya dapat dilakukan untuk karakter huruf. Sedangkan pada metode Advanced Encryption Standard, proses enkripsi/dekripsi file dapat dilakukan untuk seluruh karakter huruf, angka dan simbol.

## II. LANDASAN TEORI

### 2.1 Enkripsi dan Dekripsi

Enkripsi adalah sebuah proses yang melakukan perubahan sebuah kode yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti. Enkripsi dapat diartikan sebagai kode atau *cipher*. Enkripsi merupakan sebuah sistem pengkodean menggunakan suatu tabel atau kamus yang telah didefinisikan untuk mengganti kata dari informasi yang dikirim. Sebuah *cipher* menggunakan suatu algoritma yang dapat mengkodekan semua aliran data (stream) bit dari sebuah pesan menjadi *cryptogram* yang tidak dimengerti (*unintelligible*). Oleh karena teknik cipher merupakan suatu sistem yang telah siap untuk diautomasikan, maka teknik ini digunakan dalam sistem keamanan komputer dan jaringan (Manan and Subari 2014).

Enkripsi dimaksudkan untuk melindungi informasi agar tidak terlihat oleh orang atau pihak yang tidak berhak. Informasi ini dapat berupa nomor kartu kredit, catatan penting dalam komputer, maupun *password* untuk mengakses sesuatu. Dengan mengenkripsi paket data yang lalu lalang di Internet, walaupun seseorang dapat menangkap paket-paket data tersebut, tetap saja tidak dapat memahami artinya. Enkripsi juga digunakan untuk verifikasi (Wirdasari, 2008). Saat mengunduh *software*, maka akan diketahui bahwa *software* yang diunduh adalah asli, bukan yang telah dipasang Trojan di dalamnya.

Terdapat tiga kategori enkripsi, yaitu (Wahana Komputer, 2003): kunci enkripsi rahasia, kunci enkripsi publik dan fungsi *one-way* atau fungsi satu arah yang adalah suatu fungsi dimana informasi dienkripsi untuk menciptakan "*signature*" dari informasi asli yang bisa digunakan untuk keperluan autentikasi.

Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikan informasi teracak menjadi bentuk aslinya. Algoritma yang digunakan harus terdiri dari susunan prosedur yang

direncanakan secara hati-hati yang harus secara efektif menghasilkan sebuah bentuk terenkripsi yang tidak bisa dikembalikan, bahkan sekalipun dengan algoritma yang sama.

## 2.2. Caesar Cipher

*Caesar Cipher* adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi substitusi dimana setiap huruf pada teks terang (*plaintext*) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet (Haryanto, Apriani and Sefyanto 2012). Pada *Caesar Cipher*, tiap huruf disubstitusi dengan huruf ketiga berikutnya dari susunan alphabet yang sama. Dalam hal ini kuncinya adalah pergeseran huruf. Susunan alphabet setelah digeser sejauh 3 huruf atau disebut ROT3 (rotate 3) membentuk sebuah tabel substitusi sebagai berikut:

Alfabet Biasa: 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

Alfabet Sandi: 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

Untuk menyandikan sebuah pesan, cukup mencari setiap huruf yang hendak disandikan di alfabet biasa, lalu dituliskan huruf yang sesuai pada alfabet sandi. Untuk memecahkan sandi tersebut digunakan cara sebaliknya. Contoh penyandian sebuah pesan sebagai berikut:

Teks Terang: JANGAN KE BLOK D Teks Sandi: MDQJDQ NH EORN G

Dengan mengkodekan setiap huruf alfabet dengan integer: 'A'=0, 'B'=1, ..., 'Z'= 25, maka secara matematis pergeseran 3 huruf alfabetik ekuivalen dengan melakukann operasi modulo terhadap *plaintext* menjadi *ciphertext* dengan persamaan:

$$C = E(P) = (P + 3) \bmod 26 \quad (1)$$

Pada persamaan 1, E adalah fungsi enkripsi, P adalah *plaintext*, C adalah *ciphertext*. Dilakukan modulo dengan 26 karena ada 26 huruf di dalam alphabet. Penerima pesan mengembalikan lagi *ciphertext* dengan operasi kebalikan, secara matematis dapat dinyatakan dengan persamaan:

$$P = D(C) = (C - 3) \bmod 26 \quad (2)$$

Dapat diperhatikan bahwa fungsi D adalah balikan (invers) dari fungsi E:

$$D(C) = E^{-1}(P) \quad (3)$$

Penggunaan dari *Caesar Cipher* ini dapat dimodifikasi dengan mengubah jumlah geseran (bukan hanya 3). Jadi *Caesar Cipher* dapat digunakan dengan geser 7, misalnya. Hal ini dilakukan untuk lebih menyulitkan orang yang ingin menyadap pesan sebab semua kombinasi harus dicoba (26 kemungkinan geser)

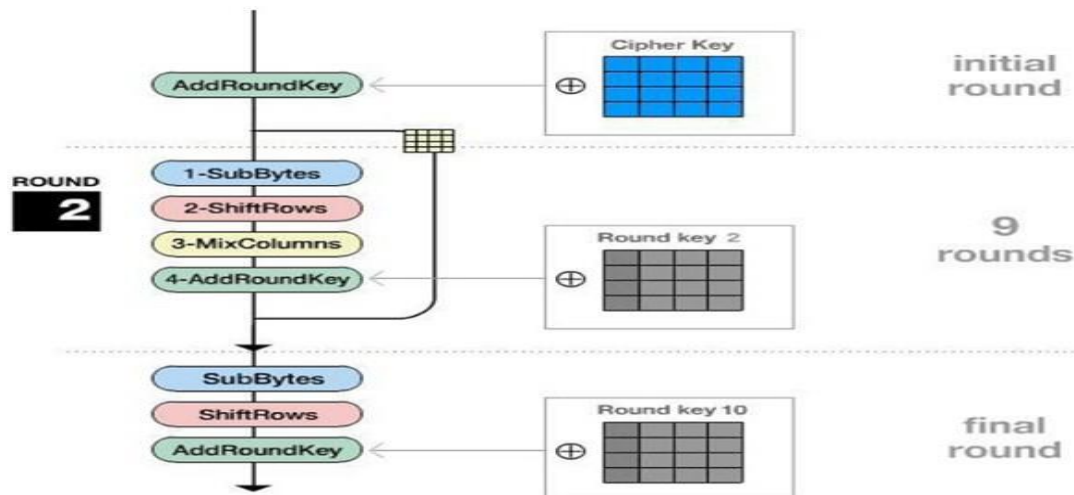
## 2.3. Advanced Encryption Standard (AES)

AES adalah lanjutan dari algoritma enkripsi standar DES (*Data Encryption Standard*) yang masa berlakunya diang- gap telah usai karena faktor keamanan. Kecepatan komputer yang sangat pesat diang- gap sangat membahayakan DES, sehingga ditetapkanlah algoritma baru Rijndael sebagai AES (Surian 2006). Kriteria pemilihan AES didasarkan pada 3 kriteria utama yaitu: keamanan, harga, dan karakteristik algoritma beserta implementasinya (Mu- nawar 2012). Keamanan merupakan faktor terpenting dalam evaluasi (minimal seaman *triple* DES), yang meliputi ketahanan terhadap semua analisis sandi yang telah diketahui dan diharapkan dapat menghadapi analisis sandi yang belum diketahui.

Di samping itu, AES juga harus dapat digunakan secara bebas tanpa harus membayar royalti, dan juga murah untuk diimplementasikan pada smart card yang memiliki ukuran memori kecil. AES juga harus efisien dan cepat (minimal secepat *Triple* DES) dijalankan dalam berbagai mesin

8 bit hingga 64 bit, dan berbagai perangkat lunak.

AES memiliki blok masukan dan ke- luaran serta kunci 128 bit. Untuk tingkat keamanan yang lebih tinggi, AES dapat menggunakan kunci 192 dan 256 bit (Lusi- ana, 2011). Setiap masukan 128 bit *plaintext* dimasukkan ke dalam *state* yang berbentuk bujursangkar berukuran 4×4 *byte*. *State* ini di XOR dengan *key* dan selanjutnya dio- lah 10 kali dengan substitusi-transformasi *linear-Addkey*, dan di akhir diperoleh *ciphertext*. Diagram AES dapat dilihat pada Gambar 1.



Gambar 1. Diagram AES

Berikut ini adalah operasi Rijndael (AES) yang menggunakan 128 bit kunci (Munir, 2006):  
Ekspansi kunci utama (dari 128 bit menjadi 1408 bit)

Pencampuran subkey

Diulang dari  $i=1$  sampai  $i=10$  Trans- formasi: *ByteSub* (substitusi per *byte*)

*Shift- Row* (Pergeseren byte per baris) *MixCol- umn* (Operasi perkalian GF(2) per kolom)

Pencampuran subkey (dengan XOR) Transformasi : *ByteSub* dan *ShiftRow*

## 2.4. PHP

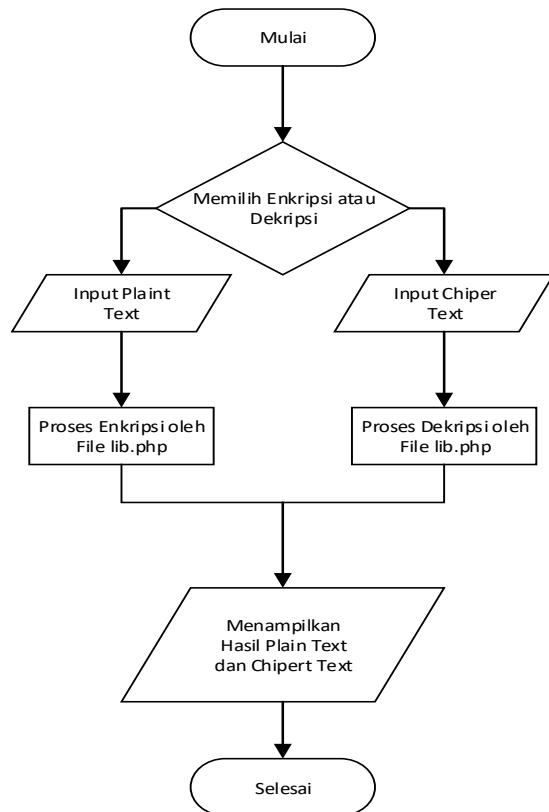
PHP (akronim dari PHP: Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi web. Berbeda dengan HTML yang hanya bisa menampilkan konten statis, PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website. Blog, Toko Online, CMS, Forum, dan Website Social Networking adalah contoh aplikasi web yang bisa dibuat oleh PHP. PHP adalah bahasa scripting, bukan bahasa tag-based seperti HTML. PHP termasuk bahasa yang cross-platform, ini artinya PHP bisa berjalan pada sistem operasi yang berbeda-beda (Windows, Linux, ataupun Mac). Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran “.php”.

## III. PEMBAHASAN

### 3.1. Membangun Aplikasi Kriptografi Metode Rot3

#### 3.1.1. Flowchart

Berikut ini adalah flowchart yang akan menggambarkan proses sistem Kriptografi metode ROT3 bekerja terdapat pada Gambar 2.



Gambar 2. Flowchart Aplikasi dengan Caesar chiper Rot 3

Pada Aplikasi pengguna akan memilih antara enkripsi atau dekripsi yang akan dilakukan kemudian memasukan karakter pada bagian input masukan plain text jika akan melakukan enkripsi atau masukan chipertext jika pengguna akan memilih untuk dekripsi text. Dalam Aplikasi dibuat khusus untuk enkripsi dan deskripsi kata yang terdiri rangkaian huruf A-Z jadi number maupun spesial karakter tidak dapat dimasukan kedalam kolom input.

### 3.1.2. Source Code

Dalam Aplikasi ini terdapat 3 file utama dalam bentuk php yaitu sebagai berikut :

#### a. lib.php

```

<?php
function rot3($str){
    $d =
array("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","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");
    $a =
array("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","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");
    $i=0;
    foreach($d as $dd){
        if($dd==$str){
            break;
        }
        $i++;
    }
    return $a[$i];
}
  
```

```

}

function d_rot3($str){
    $a =
array("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","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 =
array("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","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");
    $i=0;
    foreach($d as $dd){
        if($dd==$str){
            break;
        }
        $i++;
    }
    return $a[$i];
}
?>

```

Pada file ini kunci dari sistem kriptografi yang dijalankan dimana fungsi rot3 sebagai fungsi yang melakukan proses enkripsi dan fungsi d-rot3 sebagai fungsi yang memproses membaca cipher text untuk dilakukannya enkripsi. Dalam fungsi ini memasukan array yang akan membaca sebagai plain text yaitu a-z dan A-Z yang kemudian array untuk ciphertext yang menerapkan caesar cipher rot3 dengan menggeser 3 huruf A=D.

## b. rot.php

```

<div class="row bg-form">
    <div class="col-md-6">
        <form action="" method="post" class="form">
            <h2>ENKRIPSI ROT3</h2>
            <p>Plaintext<br /><input class="form-control" rows="3" cols="50"
                name="plain" onKeyPress="return
goodchars(event,'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ',this)"
required="" /></p>
            <p><button type="submit" class="btn btn-success" name="Enkripsi">Enkripsi</button></p>
        </form>
    </div>
    <div class="col-md-6">
        <div class="">
            <h2>HASIL ENKRIPSI ROT3</h2>
            <?php
                if ($_SERVER['REQUEST_METHOD'] == 'POST')
                {
                    if(!empty($_POST[plain]))
                    {
                        $kata = $_POST[plain];
                        $data = str_split($kata);

                        $i=0;

```

```

        foreach($data as $aaa){
            $see = rot3($aaa);
            $r[$i]=$see;
            $i++;
        }
        $rot3 = implode($r);
        echo"<h3>Plaintext</h3>";
        <p>$kata</p>
        <h3>Chipertext</h3>";
        <p>$rot3</p>
    ";
    }
    else
    {
        echo"<span style='color:#f00;'>Masukkan
        Plaintext...</span>";
    }
    }
    ?>
</div>
</div>
</div>

```

File ini bertugas menampilkan antarmuka untuk pengguna melakukan kegiatan enkripsi dengan metode rot3. Pengguna akan dihadapkan sebuah kolom input Plaintext dan tombol enkripsi yang kemudian akan menampilkan hasil dari enkripsi terdiri dari Plain Text dan Chipertext.

### c. d\_rot3.php

```

<div class="row bg-form">
    <div class="col-md-6">
        <form action="" method="post" class="form">
            <h2>DEKRIPSI ROT3</h2>
            <p>Chipertext<br /><input class="form-control" rows="3" cols="50" name="chiper"
onKeyPress="return
goodchars(event,'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ',this)"
required="" /></p>
            <p><button type="submit" class="btn btn-success" name="Dekripsi">Dekripsi</button></p>
        </form>
    </div>
    <div class="col-md-6">
        <div class="">
            <h2>HASIL DEKRIPSI ROT3</h2>
            <?php
            if ($_SERVER['REQUEST_METHOD'] == 'POST')
            {
                if(!empty($_POST[chiper]))
                {
                    $kata = $_POST[chiper];
                    $data = str_split($kata);

                    $i=0;

```



```

        foreach($data as $aaa){
            $ee = d_rot3($aaa);
            $r[$i]=$ee;
            $i++;
        }

        $dekrip = implode($r);

        echo"
        <h3>Chipertext</h3>
        <p>$_POST[chiper]</p>

        <h3>Plaintext</h3>
        <p>$dekrip</p>";

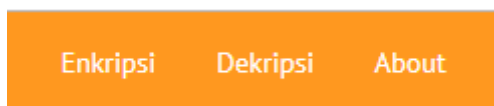
    }
    else
    {
        echo"<span style=\"color:#f00;\">Masukkan Chipertext...</span>";
    }
}
?>
</div>
</div>
</div>

```

File ini bertugas menampilkan antarmuka untuk pengguna melakukann kegiatan dekripsi dengan metode rot3. Pengguna akan dihadapkan sebuah kolom input Chipertext dan tombol dekripsi yang kemudian akan menampilkan hasil dari enkripsi terdiri dari Plain Text dan Chipertext.

### 3.1.3. Aplikasi

#### a. Navigasi



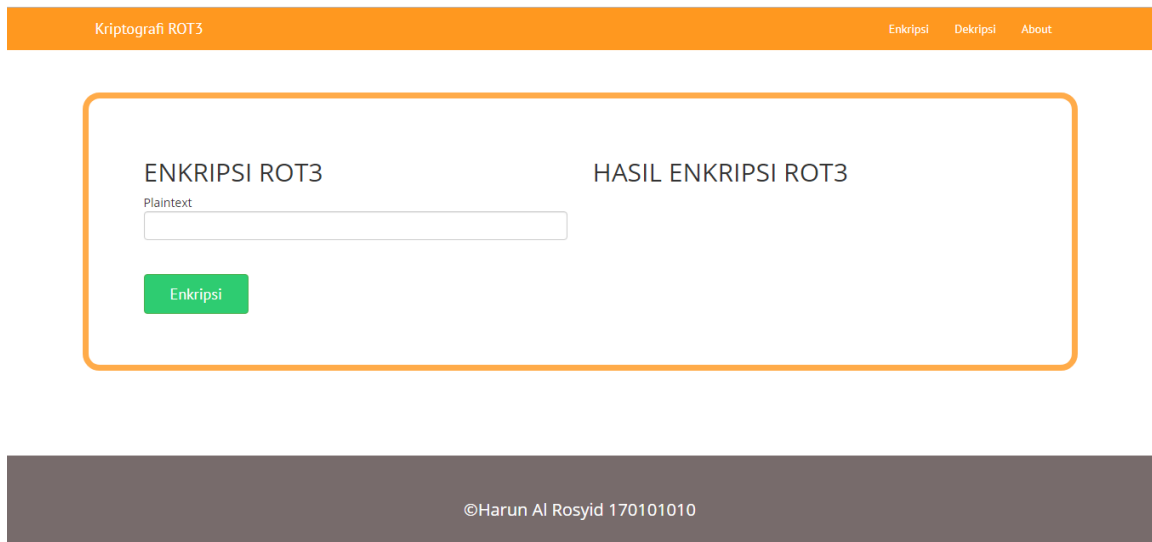
Gambar 3. Navigasi pada header

Pada bagian navigasi yang terdiri dari:

1. Menu Enkripsi sebagai navigasi untuk halaman Enkripsi
2. Menu Dekripsi sebagai navigasi untuk halaman dekripsi
3. About : merupakan penjelasan singkat dari metode enkripsi yang dipakai yaitu Caesar Chiper rot3

## b. Proses Enkripsi

Pada halaman ini akan memunculkan antarmuka yang menampilkan program enkripsi dari aplikasi Kriptografi ROT3. Pengguna akan ditampilkan sebuah kolom input untuk memasukan Plaintext dan tombol enkripsi untuk submit plaintext kemudian mendapatkan hasil dari enkripsi plaintext yaitu chipertext.



The screenshot shows the main interface of the 'Kriptografi ROT3' application. At the top, there is an orange header bar with the title 'Kriptografi ROT3' on the left and navigation links 'Enkripsi', 'Dekripsi', and 'About' on the right. The main content area is white and contains a large orange-bordered box. Inside this box, on the left, is the section 'ENKRIPSI ROT3' with a 'Plaintext' label above a text input field. Below the input field is a green button labeled 'Enkripsi'. On the right side of the box is the section 'HASIL ENKRIPSI ROT3'. At the bottom of the page, there is a dark gray footer bar with the copyright notice '©Harun Al Rosyid 170101010'.

Gambar 4. Halaman Enkripsi

## ENKRIPSI ROT3

Plaintext

Enkripsi

Gambar 5. Kolom input plaintext dan tombol submit enkripsi dimasukan plaintext.

## HASIL ENKRIPSI ROT3

Plaintext

jakarta

Chipertext

mdnduwd

Gambar 6. Hasil dari proses enkripsi menampilkan plaintext dan hasil enkripsi berupa chipertext

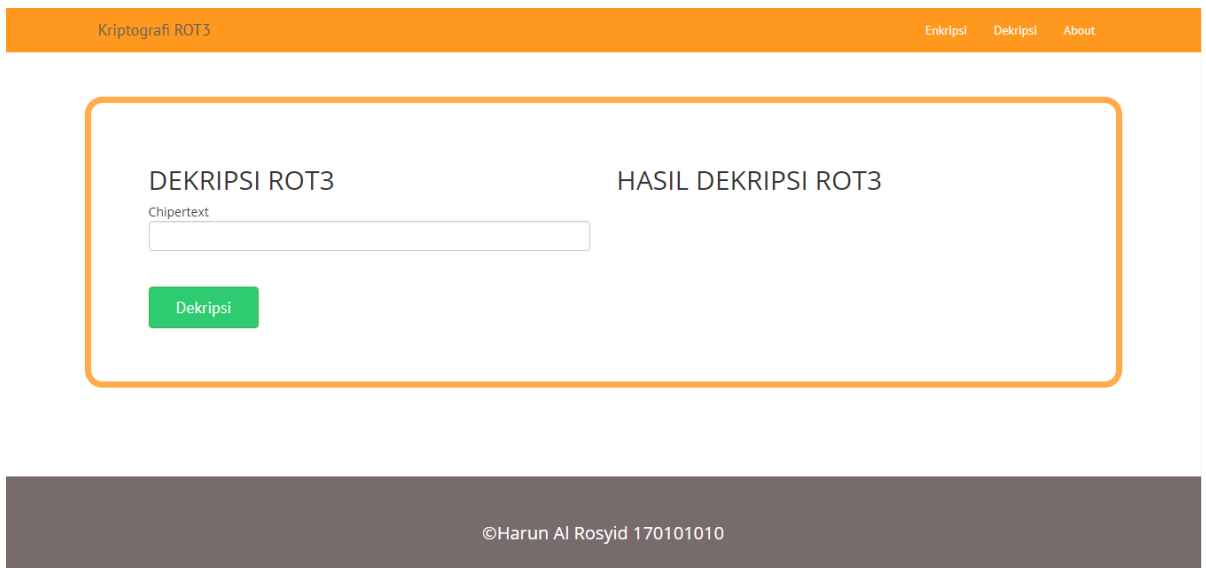
Hasil Enkripsi disini sesuai dengan algoritama yang diterapkan pada caesar chper rot 3

Plain text : 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  
Chipertext : 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

Plain text : Jakarta  
Chipertext : mdnduwd

### c. Halaman Dekripsi

Pada halaman ini akan memunculkan antarmuka yang menampilkan program Dekripsi dari aplikasi Kriptografi ROT3. Pengguna akan ditampilkan sebuah kolom input untuk memasukan Chipertext dan tombol Dekripsi untuk submit ciphertext kemudian mendapatkan hasil dari Dekripsi chipertext yaitu plaintext.



The screenshot shows a web application titled "Kriptografi ROT3". It has a navigation bar with links for "Enkripsi", "Dekripsi", and "About". The main content area is titled "DEKRIPSI ROT3" and contains a form with a label "Chipertext" above a text input field. Below the input field is a green button labeled "Dekripsi". To the right of the input field, the text "HASIL DEKRIPSI ROT3" is displayed. The footer of the application shows the copyright notice "©Harun Al Rosyid 170101010".

Gambar 7. Halaman Dekripsi

## DEKRIPSI ROT3

Chipertext

Dekripsi

Gambar 5. Kolom input chipertext dan tombol submit dekripsi dimasukan chipertext.

# HASIL DEKRIPSI ROT3

Chipertext

mdnduwd

Plaintext

jakarta

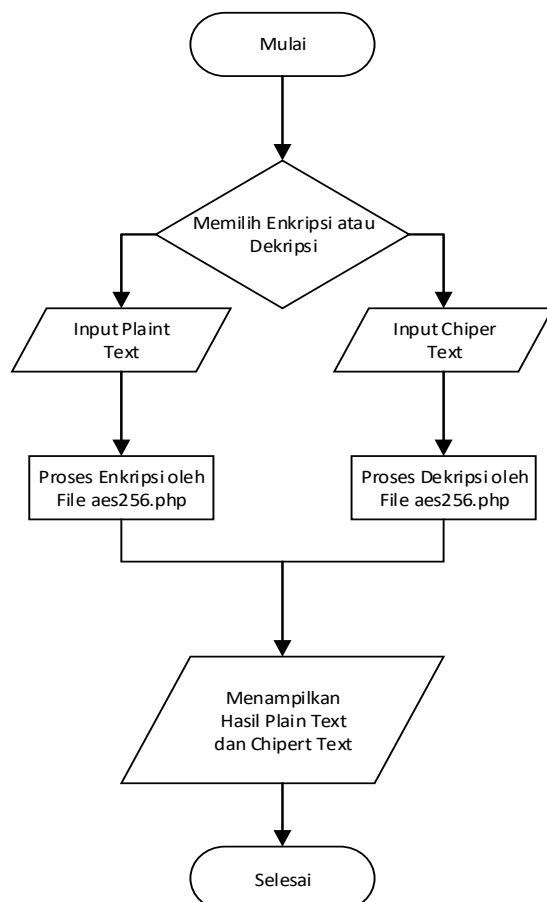
Gambar 6. Hasil dari proses dekripsi menampilkan chipertext dan hasil dekripsi berupa plaintext.

Seperti yang dijelaskan sebelumnya pada enkripsi . Maka pada proses dekripsi hanya kebalikan dari algoritma sebelumnya sehingga pada plain text **jakarta** menjadi **mdnduwd**. Caesar cipher rot3 sendiri merupakan algoritma yang klasik dalam proses kriptografi karena tingkat kerumitan yang sangat rendah maka sangat tidak disarankan untuk memakai cara ini dalam sistem yang kita buat.

## 3.2. Modifikasi Aplikasi menggunakan Algoritma AES 256

### 3.2.1. Flowchart

Berikut ini adalah flowchart yang akan menggambarkan proses sistem Kriptografi metode AES 256 bekerja terdapat pada Gambar 7.



Gambar 7. Flowchart Aplikasi dengan Aes 256

Pada Aplikasi pengguna akan memilih antara enkripsi atau dekripsi yang akan dilakukan kemudian memasukan karakter pada bagian input masukan plain text jika akan melakukan enkripsi atau masukan cipertext jika pengguna akan memilih untuk dekripsi text. Untuk versi menggunakan AES 256 input karakter plaintext telah diubah agar support seluruh jenis karakter.

### 3.1.2. Source Code

Modifikasi sourcode dilakukan untuk mendukung algoritma AES 256 . Berberapa perubahan antara lain merubah file lib.php dengan algoritma Caesar chiper ROT3 dengan file aes256.php telah mendukung algoritma AES 256. File rot3.php dan d\_rot3.php sebagai halaman enkripsi dan deskripsi diganti dengan file enkripsi.php dan dekripsi.php agar mampu menjalankan algoritma yang telah ditanam dalam file aes256.php. Berikut ini adalah source code yang telah dirubah.

#### a. aes256.php

```
<?php

function enkripsi( $string )
{
    $output = false;

    $encrypt_method = "AES-256-CBC";
    $secret_key = '';
    $secret_iv = '';

    $key = hash('sha256', $secret_key);

    $iv = substr(hash('sha256', $secret_iv), 0, 16);

    $output = openssl_encrypt($string, $encrypt_method, $key, 0, $iv);
    $output = base64_encode($output);

    return $output;
}

function dekripsi($string)
{
    $output = false;

    $encrypt_method = "AES-256-CBC";
    $secret_key = '';
    $secret_iv = '';

    $key = hash('sha256', $secret_key);

    $iv = substr(hash('sha256', $secret_iv), 0, 16);

    $output = openssl_decrypt(base64_decode($string), $encrypt_method, $key, 0, $iv);

    return $output;
}
```

?>

Penerapan fungsi enkripsi dan deskripsi dengan metode algoritma AES 256. Pada bagian ini tidak menggunakan array yang terdiri dari a-z dan A-Z seperti pada file lib.php sebelumnya di metode rot3. Dalam file aes256.php ini secret key dan secret iv tidak diisi agar bisa dibandingkan dengan metode caesar chipper rot3 yang tidak menggunakan kunci khusus. Untuk hasil output dipadukan dengan base 64.

## b. enkripsi.php

```
<div class="row bg-form">
  <div class="col-md-6">
    <form action="" method="post" class="form">
      <h2>ENKRIPSI AES 256</h2>
      <p> Plaintext<br /><input class="form-control" rows="3" cols="50" type="text" name="enkrip"
required=""> </p>
      <p> <input type="submit" class="btn btn-success" value="Enkripsi"> </p>
    </form>
  </div>
  <div class="col-md-6">
    <div class="">
      <h2>HASIL ENRIPSI AES 256</h2>
      <?php

      if ( isset( $_POST['enkrip'] ) ) {

        if(!empty($_POST[enkrip]))
        {
          $kata = $_POST[enkrip];
          $data = str_split($kata);

        }

        echo "<h3>Plaintext </h3> <p>$kata</p>";
        echo "<h3>Chipertext</h3> <p>" . enkripsi( $_POST['enkrip'] ) . "</p>";

      }

    </div>
  </div>
</div>
```

Sourcode ini meupakan pengganti dari rot3.php yang sebelumnya digunakan untuk enkripsi dengan metode caesar chipper rot 3 masih memiliki fungsi yang sama namun fungsi yang dipanggil dari file aes256.php .

## c. dekripsi.php

```
<div class="row bg-form">
  <div class="col-md-6">
    <form action="" method="post" class="form">
```

```

<h2>ENKRIPSI AES 256</h2>
<p> Plaintext<br /><input class="form-control" rows="3" cols="50" type="text" name="enkrip"
required=""> </p>
<p> <input type="submit" class="btn btn-success" value="Enkripsi"> </p>
</form>
</div>
<div class="col-md-6 ">
<div class="">
<h2>HASIL ENRIPSI AES 256</h2>
<?php

    if ( isset( $_POST['enkrip'] ) ) {

        if(!empty($_POST[enkrip]))
        {
            $kata = $_POST[enkrip];
            $data = str_split($kata);

        }

        echo "<h3>Plaintext </h3> <p>$kata</p>";
        echo "<h3>Chipertext</h3> <p>" . enkripsi( $_POST['enkrip'] ) . "</p>";

    }

    ?>

<?php } ?>
</div>
</div>
</div>

```

### 3.2.3. Aplikasi versi AES 256

#### a. Navigasi



Gambar 8. Navigasi pada header

Pada bagian navigasi yang terdiri dari:

1. Menu Enkripsi sebagai navigasi untuk halaman Enkripsi
2. Menu Dekripsi sebagai navigasi untuk halaman dekripsi
3. About : merupakan penjelasan singkat dari metode enkripsi yang dipakai yaitu Aes 256

#### b. Proses Enkripsi

Pada halaman ini akan memunculkan antarmuka yang menampilkan program enkripsi dari aplikasi Kriptografi yang sudah menggunakan metode Aes 256 . Pengguna akan ditampilkan sebuah kolom input untuk memasukan Plaintext dan tombol enkripsi untuk submit plaintext kemudian mendapatkan hasil dari enkripsi plaintext yaitu chipertext.

Kriptografi Algoritma AES 256

EnkripsiDekripsiAbout

ENKRIPSI AES 256

Plaintext

harun

Enkripsi

HASIL ENRIPSI AES 256

Plaintext

harun

Chipertext

cG5DZ1ZhMTN1cWtCVzEyeGRHam5XZz09

Gambar 9. Halaman Enkripsi

## ENKRIPSI AES 256

Plaintext

jakarta

Enkripsi

Gambar 10. Kolom input plaintext dan tombol submit enkripsi dimasukan plaintext.

## HASIL ENRIPSI AES 256

Plaintext

jakarta

Chipertext

SVhZcklkcvNnSW8wT0ppT2JMWkNvQT09

Gambar 11. Hasil dari proses enkripsi menampilkan plaintext dan hasil enkripsi berupa chipertext

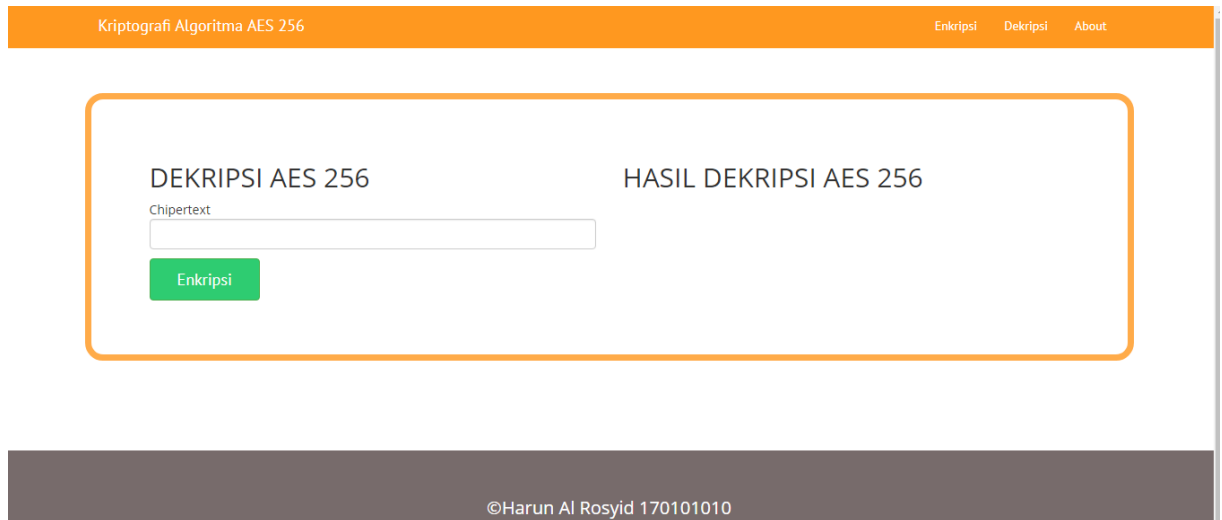
Dengan menerapkan metode AES 256 yang memiliki blok kunci 256 bit hasil chipertext yang didapatkan lebih rumit dan lebih aman digunakan dibanding metode sebelumnya. Pada metode ini hasil dari proses :



Plain text : jakarta  
Chipert text : SVhZcklkcVNnSW8wT0ppT2JMWkNvQT09

### c. Halaman Dekripsi

Pada halaman ini akan memunculkan antarmuka yang menampilkan program Dekripsi dari aplikasi Kriptografi yang sudah menggunakan Aes 256. Pengguna akan ditampilkan sebuah kolom input untuk memasukan Chipertext dan tombol Dekripsi untuk submit ciphertext kemudian mendapatkan hasil dari Dekripsi chipertext yaitu plaintext.



Gambar 12. Halaman Dekripsi

## DEKRIPSI AES 256

Chipertext

SVhZcklkcVNnSW8wT0ppT2JMWkNvQT09

Enkripsi

Gambar 13. Kolom input chipertext dan tombol submit dekripsi dimasukan chipertext.

## HASIL DEKRIPSI AES 256

Chipertext

SVhZcklkcVNnSW8wT0ppT2JMWkNvQT09

Plaintext

jakarta

Gambar 14. Hasil dari proses dekripsi menampilkan chipertext dan hasil dekripsi berupa plaintext.

Deskripsi yang dilakukan aplikasi sebagai berikut :

Chiptext : SVhZcklkcVNnSW8wT0ppT2JMWkNvQT09

Plain text : jakarta

### 3.3. Perbandingan hasil Metode Caesar Chiper ROT3 dan AES 256

Metode	Plain Text	Chiper Text
Caesar Chiper ROT3	jakarta	mdnduwd
AES 256	jakarta	SVhZcklkcVNnSW8wT0ppT2JMWkNvQT09

Pada metode Caesar chiper ROT3 diatas hasil enkripsi menunjukan keaman yang lebih rendah. Hal ini terjadi karena tidak diterapkannya chiper key dan blok yang hanya dengan mengeser 3 huruf pada 26 kunci yang ada. Tanpa ada tambahan kemanan lain hasil enkripsi dari metode Caesar chiper ROT3 sangat mudah untuk dipecahkan. Sedangkan pada metode AES 256 menampilkan hasil enkripsi yang lebih rumit berkat penerapan AES 256 bisa di pastikan tingkat keamanan dan proses kriptografi lebih baik dari metode Caesar chiper ROT3.

## IV. KESIMPULAN

Berdasarkan hasil yang telah didapat dari prose kriptografi yang menampilkan plain text dan chiper text menggunakan pemrograman PHP tingkat keamanan penyamaran sandi yang dilakukan dengan metode Caesar chiper ROT3 dan AES 256 Menunjukan bahwa AES 256 yang memiliki tingkat keamanan dan kerumitan penamaran sandi sehingga lebih layak untuk diimplementasikan pada suatu aplikasi. AES 256 memiliki blok masukan dan keluaran serta kunci 256 bit. Untuk tingkat keamanan yang lebih tinggi. Setiap masukan 256 bit plaintext dimasukkan ke dalam state yang berbentuk bujursangkar berukuran  $4 \times 4$  byte. Kelebihan AES antara lain:

- AES terbukti kebal menghadapi serangan konvensional (linear dan diferensial attack) yang menggunakan statistik untuk memecahkan sandi.
- Kesederhanaan AES memberikan keuntungan berupa kepercayaan bahwa AES tidak ditanami trapdoor
- Bila persamaan AES dapat dipecahkan dengan sedikit pasangan plaintext/ciphertext, maka riwayat AES akan berakhir
- AES didesain dengan sangat hati-hati dan baik sehingga setiap komponennya memiliki tugas yang jelas
- AES memiliki sifat cipher yang diharapkan yaitu : tahan menghadapi analisis sandi yang diketahui, fleksibel digunakan dalam berbagai perangkat keras dan lunak, baik digunakan untuk fungsi hash karena tidak memiliki weak (semi weak) key, cocok untuk perangkat yang membutuhkan key agility yang cepat, dan cocok untuk stream cipher.

## V. DAFTAR PUSTAKA

- Aji Fitrah Marisman, Anita Hidayati “PEMBANGUNAN APLIKASI PEMBANDING KRIPTOGRAFI DENGAN CAESAR CIPHER DAN ADVANCE ENCRYPTION STANDARD (AES) UNTUK FILE TEKS” Oktober 2015.
- Triswan Yuliono “Pengenalan PHP” Ilmukomputer.com 2005-2007.
- Haryanto, T, M Apriani, and T Sefyanto. “Peran Algoritma Caesar Cipher dalam Membangun Karakter Akan Kesadaran Keamanan Informasi.” Yogyakarta, November 2012.
- Lusiana, Veronica. “Implementasi Kriptografi pada File Dokumen Menggunakan Algoritma AES-128.” *Jurnal Dinamika Informatika* Vol, no. No 2 (2011).
- Manan, S, and A Subari. “Implementasi AES CIPHER CLASS Untuk Enkripsi URL Di Informasi Akademik Fakultas Teknik Universitas Diponegoro.” *Jurnal Sistem Komputer Universitas Diponegoro*, 2014.