IMPLEMENTASI KRIPTOGRAFI UNTUK KEAMANAN DATA FILE MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) DENGAN PENERAPAN TEKNIK SELEKTIF

Skripsi

untuk memenuhi sebagian persyaratan

meraih S-1

Program Studi Teknik Informatika

Disusun Oleh :

Didik Eko Pramono

15650051

Kepada

TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA

YOGYAKARTA

2019

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1972/Un.02/DST/PP.00.9/05/2019

Tugas Akhir dengan judul : IMPLEMENTASI KRIPTOGRAFI UNTUK KEAMANAN DATA FILE MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) DENGAN PENERAPAN TEKNIK SELEKTIF

yang dipersiapkan dan disusun oleh:

Nama	: DIDIK EKO PRAMONO
Nomor Induk Mahasiswa	: 15650051
Telah diujikan pada	: Senin, 06 Mei 2019
Nilai ujian Tugas Akhir	: A-

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR

Ketua Sidang

Dr. Bambang Sugiantoro, S.Si., M.T.
NIP. 19751024 200912 1 002

Penguji I

Agus Mulyanto, S.Si., M.Kom.
NIP. 19710823 199903 1 003

Penguji II

Nurochman, S.Kom., M.Kom
NIP. 19801223 200901 1 007

Yogyakarta, 06 Mei 2019
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
DEKAN

Dr. Murtono, M.Si
NIP. 19691212 200003 1 001

Universitas Islam Negeri Sunan Kalijaga FM-UINSK-BM-05-03/R0

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi
Lamp :

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Didik Eko Pramono
NIM : 15650051
Judul Skripsi : "Implementasi Kriptografi Untuk Keamanan Data File Menggunakan Algoritma Advanced Encryption Standard (AES) dengan Penerapan Teknik Selektif"

sudah dapat diajukan kembali kepada Program Studi Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Teknik Informatika

Dengan ini kami mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 2 Mei 2019
Pembimbing

Dr. Bambang Sugiantoro M.T.
NIP. 19711024 200912 1 002

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Didik Eko Pramono

NIM : 15650051

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Menyatakan bahwa skripsi saya yang berjudul **"Implementasi Kriptografi Untuk Keamanan Data File Menggunakan Algoritma Advanced Encryption Standard (AES) dengan Penerapan Teknik Selektif"** merupakan hasil penelitian saya sendiri, tidak terdapat pada karya yang pernah di ajukan untuk memperoleh gelar kesarjana di suatu perguruan tinggi, dan bukan plagiasi karya orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 2 Mei 2019

Didik Eko Pramono

NIM.15650051

KATA PENGANTAR

Alhamdulillah, segala puji bagi Allah Subhanahu wa ta'ala atas limpahan rahmatnyaNya. Shalawat serta salam semoga tercurah kepada Nabi Muhammad SAW. Akhirnya penulis dapat menyelesaikan penelitian Tugas Akhir yang berjudul **Implementasi Kriptografi Untuk Keamanan Data File Menggunakan Algoritma Advanced Encryption Standard (AES) Dengan Penerapan Teknik Selektif**. Penulis menyadari dalam menyelesaikan skripsi ini tidak akan berjalan lancar tanpa dukungan dari berbagai pihak. Oleh karena itu, kesempatan ini penulis mengucapkan banyak terima kasih kepada:

1. Prof Drs.Yudian Wahyudi, MA,Ph.D, selaku Rektor Universitas Islam Negeri Sunan Kalijaga Yogyakarta
2. Bapak Dr. Murtono M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
3. Bapak Sumarsono M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga.
4. Bapak Dr. Bambang Sugiantoro M.T, selaku dosen pembimbing yang selalu sabar membimbing, mengarahkan, memberikan nasehat dan saran selama penyusunan skripsi.
5. Ibu, Ayah, dan adekku tercinta yang senantiasa mendoa'akan dan memberikan dukungan penuh bagi penulis.
6. Seluruh teman-teman keluarga besar Program Studi Teknik Informatika, khususnya angkatan 2015 yang telah banyak sekali memberi masukan, saran dan diskusi yang begitu berharga.

Yogyakarta, 01 Mei 2019

Didik Eko Pramono

15650051

HALAMAN MOTTO

Jika engkau menolong agama Allah , niscaya Allah akan menolongmu

Sesungguhnya setiap kesulitan slalu beserta kemudahan

Sesungguhnya kebutuhan manusia tehadap ilmu melebihi kebutuhan pada makanan dan minuman, karena seorang membutuhkan makanan dan minuman sehari cukup sekali atau dua kali, sedangkan terhadap ilmu manusia membutuhkannya dalam setiap hitungan nafasnya

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR MODUL

IMPLEMENTASI KRIPTOGRAFI UNTUK KEAMANAN DATA FILE MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) DENGAN PENERAPAN TEKNIK SELEKTIF

DIDIK EKO PRAMONO

15650051

INTISARI

Kemajuan teknologi komputer dan telekomunikasi telah memberikan banyak manfaat bagi peningkatan kualitas kehidupan sosial ekonomi. Disisi lain memunculkan ancaman baru, diantaranya adalah penyadapan, pembajakan dan penyebarluasan data oleh orang lain untuk penyalahgunaan. Dalam dunia informasi terdapat data-data penting dan bersifat rahasia yang tidak boleh diketahui oleh umum.

Berdasarkan hal tersebut peneliti akan mengembangkan suatu aplikasi kriptografi Advanced Enryption Stamdard (AES) untuk mengamankan dan menjaga kerahasiaan data file , serta melihat kinerja algoritma kriptografi tersebut dari segi waktu prosesnya. Data yang digunakan adalah data file dokumen dan file multimedia yang telah melalui ekstraksi secara selektif dengan hanya memilih 99999 byte awal file untuk mereduksi ukuran objek bistream file guna meningkatkan efisiensi algoritma tanpa mengurangi resiko sisi keamanan.

Hasil penelitian menunjukkan bahwa algoritma AES dengan panjang kunci 256 bit dapat menyandikan isi suatu file sehingga dapat mengamankan file tersebut, Adapun hasil pengujian menunjukkan bahwa penerapan selektif dapat mereduksi volume objek bitsreaam sehingga mempu meningkatkan kecepatan proses enkripsi dan dekripsi dengan hasil : file uji yang berukuran >= 99999 byte dapat di enkripsi/dekripsi dengan rentang waktu 2,6 – 3,0 detik sedangkan file uji berukuran <= 99999 byte dapat di enkripsi/dekripsi dengan lama waktu < 2,7 detik.

Kata Kunci: Kriptografi Aes 256-bit, Enkripsi Selektif,, keamanan informasi

CRYPTOGRAPHIC IMPLEMENTATION FOR SECURITY DATA FILES USING ADVANCED ENCRYPTION STANDARD (AES) ALGORITHM WITH THE PRACTICE OF SELECTIVE TECHNIQUES

DIDIK EKO PRAMONO

15650051

ABSTRACT

The progress of computer and telecommunications technology has provided many benefits for improving the quality of socio-economic life. On the other hand raises new threats, including tapping, piracy and dissemination of data by others for abuse. In the world of information there are important and confidential data that should not be known by the public.

Based on this, the researcher will develop an cryptographic application Advanced Enryption Stamdard (AES) to secure and maintain the confidentiality of data files, and see the performance of the cryptographic algorithm in terms of processing time. The data used are document file data and multimedia files that have been selectively extracted by only selecting the 99999 initial byte file to reduce the size of the bistream file object to increase the efficiency of the algorithm without reducing the risk of security.

The results showed that the AES algorithm with 256-bit key length can encode the contents of a file so that it can secure the file. The test results show that selective application can reduce the volume of bitsreaam objects so that it can increase the speed of the encryption and decryption process with results : all file with size >= 99999 bytes can be encrypted / decrypted with speed of 2.6 - 3.0 seconds while the test file measuring <= 99999 bytes can be encrypted / decrypted with a speed of time <2.7 seconds.

Keywords: Aes 256-bit cryptography, Selective Encryption, information security

BAB 1
PENDAHULUAN

1.2 LATAR BELAKANG

Perpaduan teknologi telekomunikasi, internet, dan penyiaran, telah mendorong munculnya infrastruktur ekonomi baru. Tentunya memberikan manfaat bagi peningkatan kualitas kehidupan sosial dan ekonomi dengan globalisasi ekonomi digital. Disi lain keterhubungan dengan jaringan pita lebar, memunculkan ancaman pada seluruh aset nasional. Keterhubungan dalam jaringan global ini membentuk jaringan siber yang memberi banyak kemudahan dan juga memberikan dampak kerentanan dan ancaman baru. Perkembangan teknologi yang pesat membuat teknik perang siber menjadi lebih kompleks dan lebih canggih. Kemampuan cyber intelegence negara Indonesia bukan saja dibutuhkan karena ancaman perang siber dari hari ke hari semakin besar, namun menjadi pertaruhan besar kemajuan bangsa ke depan.

Dari laporan tahunan honeynet project Badan Siber dan Sandi Negara (BSSN) dan Indonesia Honeynet Project (IHP), negara Indonesia telah menduduki peringkat kedua tertinggi perang siber pada bulan November 2018 (Dwi Amanda, Lukas and Adi Putra, 2018). Sehingga keamanan dan kerahasian data adalah sesuatu yang sangat penting untuk kemajuan bangsa ini. Tentunya hal ini membutuhkan banyak peran para peneliti dalam hal keamanan data untuk dijadikan acuan bagi para pengembang dalam memilih setrategi implementasi peningkatan aspek keamanan data atau informasi.

Kriptografi merupakan salah satu teknik untuk meningkatkan mekanisme keamanan data dengan cara melakukan penyandian terhadap data asli sehingga tidak dimengerti lagi maknanya. Saat ini banyak bermunculan algoritma kriptografi yang terus dianalisis, dicoba dan disempurnakan untuk mencari algoritma yang dianggap memenuhi standar keamanan. Beberapa algoritma

kriptografi yang dikenal antara lain AES, Blowfish, RC4, Vigenere Cipher, Enigma, IDEA dan lainnya.

Advanced Encryption Standard (AES) merupakan salah satu algoritma yang dipandang cukup kuat. Algoritma ini memiliki kemampuan bertahan dari serangan. Secara keseluruhan**,** AES terdiri dari tiga penyandian blok yaitu AES-128, AES-192 dan AES-256**.** Masing-masing penyandian AES mengenkripsi dan mendekripsi data dalam blok 128 bit menggunakan kunci kriptografi untuk 128, 192 dan 256-bit dengan 256-bit merupakan yang paling aman. Beberapa organisasi bidang teknologi komputer seperti Apple, Microsoft hingga National Security Agency (NSA) menggunakan algoritma ini sebagai standar keamanan untuk mengenkripsi data penting (Taylor, 2018).

Dalam dunia informasi terdapat data yang bersifat rahasia yang beragam jenisnya, diantara jenis data-data rahasia tersebut adalah data yang bertipe file. Data berjenis file ini memiliki tingkat sensitif tinggi. Terbukti dengan perubahan terhadap sebagian bitstream, berakibat pada kerusakan bitream seluruh file secara beruntun. Hal inilah yang mendorong peneliti menggunakan strategi implementasi penerapan teknik selektif sebagai pertimbangan yang tepat dalam menerapkan algoritma AES pada data bertipe file. Strategi ini diusulkan oleh peneliti karena strategi ini melakukan enkripsi hanya terhadap sebagian bitsream saja ,tetapi mampu berdampak pada keseluruhan bitstream. Memilih strategi yang tepat tersebut, dalam strategi implementasi dapat mereduksi ukuran volume objek bitsream yang akan dienkripsi sehingga membuat algoritma AES bekerja lebih optimal, dapat meringankan beban kerja komputasi enkripsi dan dekripsi.

Berdasar uraian diatas , peneliti tertarik membangun Aplikasi mobile untuk mengamankan data file menggunakan algoritma kriptografi AES 256-bit dengan penerapan teknik selektif. Selain itu diharapkan pula aplikasi yang dibangun ini dapat melihat kinerja algoritma tersebut dari segi waktu prosesnya.

1.2 RUMUSAN MASALAH

Berdasarkan latar belakang masalah tersebut maka dapat dirumuskan permasalahan yang dapat diselsaikan dalam penelitian ini adalah Bagaimana membuat aplikasi Mobile untuk mengamankan data file menggunakan algoritma kriptografi Advanced Encryption Standard (AES) dengan penerapan teknik selektif dan memperlihatkan kinerja algoritma tersebut dari segi waktu prosesnya.

1.3 BATASAN MASALAH

Adapun batasan masalah dalam penelitian ini adalah :

1. Data file yang digunakan adalah data file dokumen office dan file multimedia audio, gambar dan video.
2. Aplikasi mobile hanya dapat dijalankan pada platform android
3. Algoritma kriptografi yang digunakan adalah AES dengan panjang kunci 256 bit.

1.4 TUJUAN PENELITIAN

Sesuai dengan latar belakang dan batasan masalah di atas, maka tujuan dari penelitian ini adalah :

1. Mampu membuat aplikasi mobile untuk mengamankan dan merahasiakan data berupa file dokumen dan file multimedia.
2. Mampu memperlihatkan kinerja algoritma AES dengan penerapan selektif dari segi waktu prosesnya.

1.5 MANFAAT PENELITIAN

Berdasarkan Rumusan Masalah diatas , maka tujuan penelitian ini adalah :

1. Diharapakan hasil dari penelitian ini dapat dapat menjaga kerahasiaan data baik yang terhubung dengan jaringan atau tidak.
2. Meningkatkan keamanan data menggunakan kriptografi AES dengan penerapan teknik selektif.

3. Aplikasi dapat memberikan keamanan data sehingga dapat menjamin kerahasiaan data dan melindungi dari pemalsuan atau pengubahan informasi yang tidak diinginkan.
4. Diharapkan dapat menjadi bahan pertimbangan para developer dalam memilih strategi implementasi keamanan dan kerahasian data file pada komputer server menggunakan algoritma AES dengan penerapan teknik selektif.

1.6 KEBARUAN PENELITIAN

Penelitian yang berhubungan dengan algoritma kriptografi sudah pernah dilakukan. Berdasarkan studi pustaka yang dilakukan penulis, belum ada penelitian tentang "*Implementasi Kriptografi Untuk Keamanan Data File Menggunakan Algoritma Advanced Encryption Standard (AES) Dengan Penerapan Teknik Selektif*" khususnya di lingkungan UIN Sunan Kalijaga Yogyakarta.

BAB V
PENUTUP

5.1 Kesimpulan

Dari penelitian yang dilakukan, ada beberapa kesimpulan yang di dapatkan, yaitu :

1. Aplikasi mobile kriptografi Advanced Encryption Standard dengan penerapan Teknik selektif untuk mengamankan dan merahasiakan file dokumen office dan file multimedia meliputi gambar, audio, video telah berhasil dibuat.
2. Aplikiasi juga dapat digunakan untuk melihat kinerja algoritma Advanced Encryption Standard dengan penerapan Teknik selektif dari segi waktu prosesnya/kecepatannya yang di kaitkan dengan ukuran sebuah file.
3. Hasil pengujian menunjukkan bahwa kecepatan enkripsi dan dekripsi dipengaruhi besarnya file, sehingga reduksi volume bitstream karena pengambilan bitsream dengan penerapan selektif, menunjukkan dapat meningkatkan kecepatan proses enkripsi dan dekripsi tanpa mengurangi resiko keamanan, dengan hasil a.)Semua File uji berukuran >= 99999 byte dapat di enkripsi/dekripsi dengan rentang waktu 2,6 – 3,0 detik b.) Semua file uji berukuran <= 99999 byte dapat di enkripsi/dekripsi dengan lama waktu < 2,7 detik.
4. Enkripsi dengan teknik penerapan selektif dengan memilih hanya 99999 byte pertama, memproleh tingkat keberhasilan hasil pengujian sebesar 100%. Hal yang mempengaruhi keberhasilan adalah a.) Pemilihan sebanyak 99999 byte untuk mengambil bitream selektif sudah cukup dapat mengakibatkan confuse bitsream sehinga file tidak dapat terbaca b.) Posisi 0 sampai 99999 byte mencangkup struktur header dan sebagian konten file yang merupakan struktur yang sensitif untuk membuat confuse bitsream file.

5.2 Saran

Dari penelitian ini, penulis memberikan beberapa saran untuk penelirian selanjutnya yang berkaitan dengan penelitian ini, yaitu :

1. Penerapan Teknik selektif diperlukan metode ekstraksi bistream Least significant bit (LSB) agar pengambilan bitsream yang dilakukan lebih terstruktur dan memiliki kepastian parameter atau acuan.
2. Perlu ditambahkan algoritma kompresi file agar menambah efisiensi waktu saat proses upload dan download file terhadap server saat keperluan enkripsi dan dekripsi.

DAFTAR PUSTAKA

Ariyus, D. (2006) *Kriptografi, Keamanan Data, dan Komunikasi*. Yogyakarta: Graha Ilmu.

Dwi Amanda, C., Lukas and Adi Putra, I. (2018) *Laporan Tahunan Honeynet Project BSSN IHP*. Available at: https://bssn.go.id/wp-content/uploads/2019/02/Laporan-Tahunan-Honeynet-Project-BSSN_IHP-2018.pdf.

Junaedi, M. (2003) *Pengantar XML, kuliah umum ilmu komputer*. Available at: www.ilmukomputer.com.

Menezes and Dkk (1996) *Handbook of Applied Cryptography*. CRC Press.

Munir, R. (2006) *Kriptografi*. Bandung: Penerbit Informatika.

Munir, R. (2012) 'DIGITAL MENGGUNAKAN KOMBINASI DUA CHAOS MAP DAN PENERAPAN TEKNIK SELEKTIF', 95. Available at: http://informatika.stei.itb.ac.id/~rinaldi.munir/Penelitian/Makalah_SNETE_2011.pdf.

Priyanta, F. (2011) *Pemrograman Android Untuk Pemula*. Jakarta: Penerbit Cerdas Pustaka.

Safaat, N. (2012) 'Android : Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android (Edisi Revisi)', in *ANDROID*.

Schneier, B. (1996) *Applied Cryptography 2nd*. Ltd: John Wiley & Sons.

Syarifuddin and Dkk (2013) 'Implementasi Algoritma AES Rijndael pada

Proses Enkripsi Hemat Energi Untuk Video Streaming dalam Jaringan Sensor Nirkabel', pp. 14–20.

Taylor, S. (2018) *Panduan Terbaik untuk Advanced Encryption Standard (AES)*. Available at: https://id.wizcase.com/blog/panduan-terbaik-untuk-advanced-encryption-standard-aes/.

Yuniati, V., Indriyanta, G. and Rachmat C., A. (2018) 'Enkripsi Dan Dekripsi Dengan Algoritma Aes 256 Untuk Semua Jenis File', *Jurnal Informatika*. doi: 10.21460/inf.2009.51.69.

LAMPIRAN

SOURCE CODE

Class Aes.php

```
Class Aes
{
   public static function cipher($input, $w)
   {
      $Nb = 4;
      $Nr = count($w) / $Nb - 1;
      $state = array();
      for ($i = 0; $i < 4 * $Nb; $i++) $state[$i % 4][floor($i / 4)] =
$input[$i];
      $state = self::addRoundKey($state, $w, 0, $Nb);
      for ($round = 1; $round < $Nr; $round++) {
         $state = self::subBytes($state, $Nb);
         $state = self::shiftRows($state, $Nb);
         $state = self::mixColumns($state, $Nb);
         $state = self::addRoundKey($state, $w, $round, $Nb);
      }
      $state = self::subBytes($state, $Nb);
      $state = self::shiftRows($state, $Nb);
      $state = self::addRoundKey($state, $w, $Nr, $Nb);
      $output = array(4 * $Nb);
      for ($i = 0; $i < 4 * $Nb; $i++) $output[$i] = $state[$i % 4][floor($i
/ 4)];
      return $output;
   }
   private static function addRoundKey($state, $w, $rnd, $Nb)
```

```
  {
    for ($r = 0; $r < 4; $r++) {
      for ($c = 0; $c < $Nb; $c++) $state[$r][$c] ^= $w[$rnd * 4 +
$c][$r];
    }
    return $state;
  }
  private static function subBytes($s, $Nb)
  {
    for ($r = 0; $r < 4; $r++) {
      for ($c = 0; $c < $Nb; $c++) $s[$r][$c] = self::$sBox[$s[$r][$c]];
    }
    return $s;
  }
  private static function shiftRows($s, $Nb)
  {
    $t = array(4);
    for ($r = 1; $r < 4; $r++) {
      for ($c = 0; $c < 4; $c++) $t[$c] = $s[$r][($c + $r) % $Nb];
      for ($c = 0; $c < 4; $c++) $s[$r][$c] = $t[$c];
    }
    return $s;
  }
  private static function mixColumns($s, $Nb)
  {
    for ($c = 0; $c < 4; $c++) {
      $a = array(4);
      $b = array(4);
      for ($i = 0; $i < 4; $i++) {
        $a[$i] = $s[$i][$c];
```

```
            $b[$i] = $s[$i][$c] & 0x80 ? $s[$i][$c] << 1 ^ 0x011b :
$s[$i][$c] << 1;
        }
        $s[0][$c] = $b[0] ^ $a[1] ^ $b[1] ^ $a[2] ^ $a[3];
        $s[1][$c] = $a[0] ^ $b[1] ^ $a[2] ^ $b[2] ^ $a[3];
        $s[2][$c] = $a[0] ^ $a[1] ^ $b[2] ^ $a[3] ^ $b[3];
        $s[3][$c] = $a[0] ^ $b[0] ^ $a[1] ^ $a[2] ^ $b[3];
      }
      return $s;
    }
    public static function keyExpansion($key)
    {
      $Nb = 4;
      $Nk = count($key) / 4;
      $Nr = $Nk + 6;
      $w = array();
      $temp = array();
      for ($i = 0; $i < $Nk; $i++) {
        $r = array($key[4 * $i], $key[4 * $i + 1], $key[4 * $i + 2], $key[4
* $i + 3]);
        $w[$i] = $r;
      }
      for ($i = $Nk; $i < ($Nb * ($Nr + 1)); $i++) {
        $w[$i] = array();
        for ($t = 0; $t < 4; $t++) $temp[$t] = $w[$i - 1][$t];
        if ($i % $Nk == 0) {
          $temp = self::subWord(self::rotWord($temp));
          for ($t = 0; $t < 4; $t++) $temp[$t] ^= self::$rCon[$i / $Nk][$t];
        } else if ($Nk > 6 && $i % $Nk == 4) {
          $temp = self::subWord($temp);
        }
```

```
            for ($t = 0; $t < 4; $t++) $w[$i][$t] = $w[$i - $Nk][$t] ^ $temp[$t];
        }
        return $w;
    }
    private static function subWord($w)
    {
        for ($i = 0; $i < 4; $i++) $w[$i] = self::$sBox[$w[$i]];
        return $w;
    }
    private static function rotWord($w)
    {
        $tmp = $w[0];
        for ($i = 0; $i < 3; $i++) $w[$i] = $w[$i + 1];
        $w[3] = $tmp;
        return $w;
    }
    private static $sBox = array(
        0x63, 0x7c, 0x77, 0x7b, 0xf2, 0x6b, 0x6f, 0xc5, 0x30, 0x01, 0x67,
0x2b, 0xfe, 0xd7, 0xab, 0x76,
        0xca, 0x82, 0xc9, 0x7d, 0xfa, 0x59, 0x47, 0xf0, 0xad, 0xd4, 0xa2,
0xaf, 0x9c, 0xa4, 0x72, 0xc0,
        0xb7, 0xfd, 0x93, 0x26, 0x36, 0x3f, 0xf7, 0xcc, 0x34, 0xa5, 0xe5,
0xf1, 0x71, 0xd8, 0x31, 0x15,
        0x04, 0xc7, 0x23, 0xc3, 0x18, 0x96, 0x05, 0x9a, 0x07, 0x12, 0x80,
0xe2, 0xeb, 0x27, 0xb2, 0x75,
        0x09, 0x83, 0x2c, 0x1a, 0x1b, 0x6e, 0x5a, 0xa0, 0x52, 0x3b, 0xd6,
0xb3, 0x29, 0xe3, 0x2f, 0x84,
        0x53, 0xd1, 0x00, 0xed, 0x20, 0xfc, 0xb1, 0x5b, 0x6a, 0xcb, 0xbe,
0x39, 0x4a, 0x4c, 0x58, 0xcf,
        0xd0, 0xef, 0xaa, 0xfb, 0x43, 0x4d, 0x33, 0x85, 0x45, 0xf9, 0x02,
0x7f, 0x50, 0x3c, 0x9f, 0xa8,
```

```
    0x51, 0xa3, 0x40, 0x8f, 0x92, 0x9d, 0x38, 0xf5, 0xbc, 0xb6, 0xda,
0x21, 0x10, 0xff, 0xf3, 0xd2,
    0xcd, 0x0c, 0x13, 0xec, 0x5f, 0x97, 0x44, 0x17, 0xc4, 0xa7, 0x7e,
0x3d, 0x64, 0x5d, 0x19, 0x73,
    0x60, 0x81, 0x4f, 0xdc, 0x22, 0x2a, 0x90, 0x88, 0x46, 0xee, 0xb8,
0x14, 0xde, 0x5e, 0x0b, 0xdb,
    0xe0, 0x32, 0x3a, 0x0a, 0x49, 0x06, 0x24, 0x5c, 0xc2, 0xd3, 0xac,
0x62, 0x91, 0x95, 0xe4, 0x79,
    0xe7, 0xc8, 0x37, 0x6d, 0x8d, 0xd5, 0x4e, 0xa9, 0x6c, 0x56, 0xf4,
0xea, 0x65, 0x7a, 0xae, 0x08,
    0xba, 0x78, 0x25, 0x2e, 0x1c, 0xa6, 0xb4, 0xc6, 0xe8, 0xdd, 0x74,
0x1f, 0x4b, 0xbd, 0x8b, 0x8a,
    0x70, 0x3e, 0xb5, 0x66, 0x48, 0x03, 0xf6, 0x0e, 0x61, 0x35, 0x57,
0xb9, 0x86, 0xc1, 0x1d, 0x9e,
    0xe1, 0xf8, 0x98, 0x11, 0x69, 0xd9, 0x8e, 0x94, 0x9b, 0x1e, 0x87,
0xe9, 0xce, 0x55, 0x28, 0xdf,
    0x8c, 0xa1, 0x89, 0x0d, 0xbf, 0xe6, 0x42, 0x68, 0x41, 0x99, 0x2d,
0x0f, 0xb0, 0x54, 0xbb, 0x16);
  private static $rCon = array(
    array(0x00, 0x00, 0x00, 0x00),
    array(0x01, 0x00, 0x00, 0x00),
    array(0x02, 0x00, 0x00, 0x00),
    array(0x04, 0x00, 0x00, 0x00),
    array(0x08, 0x00, 0x00, 0x00),
    array(0x10, 0x00, 0x00, 0x00),
    array(0x20, 0x00, 0x00, 0x00),
    array(0x40, 0x00, 0x00, 0x00),
    array(0x80, 0x00, 0x00, 0x00),
    array(0x1b, 0x00, 0x00, 0x00),
    array(0x36, 0x00, 0x00, 0x00));
}
```

AesCtr.php

```
Class AesCtr extends Aes
{

   public static function encrypt($plaintext, $password, $nBits)
   {
      $blockSize = 16;
      if (!($nBits == 128 || $nBits == 192 || $nBits == 256)) return '';
      $nBytes = $nBits / 8;
      $pwBytes = array();
      for ($i = 0; $i < $nBytes; $i++) $pwBytes[$i] =
ord(substr($password, $i, 1)) & 0xff;
      $key = Aes::cipher($pwBytes, Aes::keyExpansion($pwBytes));
      $key = array_merge($key, array_slice($key, 0, $nBytes - 16));
      $counterBlock = array();
      $nonce = floor(microtime(true) * 1000);
      $nonceMs = $nonce % 1000;
      $nonceSec = floor($nonce / 1000);
      $nonceRnd = floor(rand(0, 0xffff));

      for ($i = 0; $i < 2; $i++) $counterBlock[$i] = self::urs($nonceMs, $i
* 8) & 0xff;
      for ($i = 0; $i < 2; $i++) $counterBlock[$i + 2] =
self::urs($nonceRnd, $i * 8) & 0xff;
```

```
      for ($i = 0; $i < 4; $i++) $counterBlock[$i + 4] =
self::urs($nonceSec, $i * 8) & 0xff;
      $ctrTxt = '';
      for ($i = 0; $i < 8; $i++) $ctrTxt .= chr($counterBlock[$i]);
      $keySchedule = Aes::keyExpansion($key);

      $blockCount = ceil(strlen($plaintext) / $blockSize);
      $ciphertxt = array();

      for ($b = 0; $b < $blockCount; $b++) {
         for ($c = 0; $c < 4; $c++) $counterBlock[15 - $c] = self::urs($b,
$c * 8) & 0xff;
         for ($c = 0; $c < 4; $c++) $counterBlock[15 - $c - 4] =
self::urs($b / 0x100000000, $c * 8);

         $cipherCntr = Aes::cipher($counterBlock, $keySchedule);
         $blockLength = $b < $blockCount - 1 ? $blockSize :
(strlen($plaintext) - 1) % $blockSize + 1;
         $cipherByte = array();

         for ($i = 0; $i < $blockLength; $i++) {
            $cipherByte[$i] = $cipherCntr[$i] ^ ord(substr($plaintext, $b
* $blockSize + $i, 1));
            $cipherByte[$i] = chr($cipherByte[$i]);
         }
         $ciphertxt[$b] = implode('', $cipherByte);
      }
      $ciphertext = $ctrTxt . implode('', $ciphertxt);
      $ciphertext = base64_encode($ciphertext);
      return $ciphertext;
   }
```

```
   public static function decrypt($ciphertext, $password, $nBits)
   {
      $blockSize = 16;
      if (!($nBits == 128 || $nBits == 192 || $nBits == 256)) return '';
      $ciphertext = base64_decode($ciphertext);
      $nBytes = $nBits / 8;
      $pwBytes = array();
      for ($i = 0; $i < $nBytes; $i++) $pwBytes[$i] =
ord(substr($password, $i, 1)) & 0xff;
      $key = Aes::cipher($pwBytes, Aes::keyExpansion($pwBytes));
      $key = array_merge($key, array_slice($key, 0, $nBytes - 16));
      $counterBlock = array();
      $ctrTxt = substr($ciphertext, 0, 8);
      for ($i = 0; $i < 8; $i++) $counterBlock[$i] = ord(substr($ctrTxt, $i,
1));
      $keySchedule = Aes::keyExpansion($key);
      $nBlocks = ceil((strlen($ciphertext) - 8) / $blockSize);
      $ct = array();
      for ($b = 0; $b < $nBlocks; $b++) $ct[$b] = substr($ciphertext, 8 +
$b * $blockSize, 16);
      $ciphertext = $ct;
      $plaintxt = array();
      for ($b = 0; $b < $nBlocks; $b++) {
         for ($c = 0; $c < 4; $c++) $counterBlock[15 - $c] = self::urs($b,
$c * 8) & 0xff;
         for ($c = 0; $c < 4; $c++) $counterBlock[15 - $c - 4] =
self::urs(($b + 1) / 0x100000000 - 1, $c * 8) & 0xff;
         $cipherCntr = Aes::cipher($counterBlock, $keySchedule);
         $plaintxtByte = array();
         for ($i = 0; $i < strlen($ciphertext[$b]); $i++) {
```

```
            $plaintxtByte[$i] = $cipherCntr[$i] ^
ord(substr($ciphertext[$b], $i, 1));
            $plaintxtByte[$i] = chr($plaintxtByte[$i]);
         }
         $plaintxt[$b] = implode('', $plaintxtByte);
      }
      $plaintext = implode('', $plaintxt);
      return $plaintext;
   }
   private static function urs($a, $b)
   {
      $a &= 0xffffffff;
      $b &= 0x1f; // (bounds check)
      if ($a & 0x80000000 && $b > 0) {
         $a = ($a >> 1) & 0x7fffffff;
         $a = $a >> ($b - 1);
      } else {
         $a = ($a >> $b);
      }
      return $a;
   }
}
```

Class Server.php

```
<?php
require 'aes.php';
require 'aesctr.php';
class Constants
{
   static $DB_SERVER="localhost";
   static $DB_NAME  ="crypto_dbs";
```

```
   static $USERNAME ="root";
   static $PASSWORD ="";
   static $SQL_SELECT_ALL="SELECT * FROM tb_file";
   static $SELECT_ON_USER = "SELECT
id_file,id_user,nama,type,size,kunci,file,length,bit_file FROM tb_file
WHERE id_user =";
}

class Server
{
   public function connect()
   {
      $con=new
mysqli(Constants::$DB_SERVER,Constants::$USERNAME,Constants::$PA
SSWORD,Constants::$DB_NAME);
      if($con->connect_error)
      {
         return null;
      }else
      {
         return $con;

      }
   }
   public function login(){

      $con=$this->connect();
      if($con != null)
      {
         $username =
mysqli_real_escape_string($con,$_POST['username']);
```

```
        $password =
mysqli_real_escape_string($con,$_POST['password']);
        $sql = "SELECT id,nama,username,password FROM tb_user
WHERE username = '$username'";
        $result=$con->query($sql);
        if ($result->num_rows > 0) {
           $row = mysqli_fetch_array($result);
           if (password_verify($password,$row["password"])) {

print(json_encode(array("id"=>$row["id"],"nama"=>$row["nama"],"use
rname"=>$row["username"],"succes"=> 1, "message" => "login
Succes")));
           }else{
              print(json_encode(array("succes" => 0,"message" =>
"wrong password")));
           }
        }else{
           print(json_encode(array("succes" => 0,"message" => "wrong
password")));
        }
     }else{
        print(json_encode(array("message"=>"ERROR PHP EXCEPTION :
CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
     }

  }
  public function register(){
     $con=$this->connect();
     if($con != null)
     {
        $options = [
```

```
            'cost' => 10
            ];
        $username =
mysqli_real_escape_string($con,$_POST['username']);
        $password =
mysqli_real_escape_string($con,$_POST['password']);
        $nama = mysqli_real_escape_string($con,$_POST['nama']);
        // echo $nama;
        // die();

        $passwordhash =
password_hash($password,PASSWORD_DEFAULT,$options);


        $sql = "INSERT INTO tb_user (id,nama, username, password)
VALUES ('','$nama','$username','$passwordhash')";
        try {
            $register= $con->query($sql);
            if ($register) {
                print(json_encode(array("succes"=>
"1","message"=>"Register Success, Silahkan Login")));
            }else{
                print(json_encode(array("succes"=>
"0","message"=>"Unsuccessful. Connection was successful but data
could not be register to register.")));
            }

        } catch (Exception $e) {
            print(json_encode(array("message"=>"ERROR PHP
EXCEPTION : CAN'T SAVE TO MYSQL. ".$e->getMessage())));
            $con->close();
```

```
        }
    }else{
        print(json_encode(array("message"=>"ERROR PHP EXCEPTION :
CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
    }
  }
  public function cekFile($path){
      $val=0;
      if (file_exists($path)) {
         $fileName    =pathinfo($path, PATHINFO_FILENAME);
         $fileExtension=pathinfo($path, PATHINFO_EXTENSION);
         $fileDir     =pathinfo($path, PATHINFO_DIRNAME);
         $val=$val + 1;
         $fileName= $fileName."-".$val;
         $fileCompleteName = $fileName.".".$fileExtension;

         $path =$fileDir.'/'.$fileCompleteName;
         if (file_exists($path)) {
            cekFile($path,$val);
         }
      }

          return $path;

  }
  public function InsertToEncrypt()
  {
    $con=$this->connect();
    if($con != null)
    {
```

```
        $password      =
mysqli_real_escape_string($con,$_POST['password']);
        $plaintextkunci =
mysqli_real_escape_string($con,$_POST['kunci']);
        $kunci        = AesCtr::encrypt($plaintextkunci,$password,256);
        $bit          = 256;
        $file_name     = $_FILES['file']['name'];
        $file_type     = $_FILES['file']['type'];
        $file_size     = $_FILES['file']['size'];
        $id_user       = (int)$_POST['id_user'];
        $isifile       =file_get_contents($_FILES['file']['tmp_name']);
        $bit_selektif   =substr($isifile, 0,99999);
        $yangdiencrypt  =
AesCtr::encrypt($bit_selektif,$plaintextkunci,256);
        $length = strlen($yangdiencrypt);
        $allbitEncrypt = str_replace($bit_selektif, $yangdiencrypt,
$isifile);
        $bitEncryptSlektif= substr($allbitEncrypt, 0,1500);
        if (!is_dir('encrypted/'.$id_user)) {
           mkdir('encrypted/'.$id_user, 0777, true);
        }
        $path=$this-
>cekFile("encrypted/".$id_user."/".basename($_FILES['file']['name']));
        $file_name=basename($path);
        $sql2 = "INSERT INTO
                tb_file (
                 id_file,
                 id_user,
                 nama,
                 type,
                 size,
```

```
            file,
            kunci,
            length,
            bit_file
            )VALUES (
                    '',
                    $id_user,
                    '$file_name',
                    '$file_type',
                    '$file_size',
                    '$path',
                    '$kunci',
                    '$length',
                     $bit)
                     ";
try
{
   $result=$con->query($sql2);
   if($result)
   {


          $Newpath = fopen($path,"w");
          fwrite($Newpath,$allbitEncrypt);
          fclose($Newpath);
          header('Content-Type: application/json');
          print(json_encode(array(
             'error'  => "0",
             'message'=> 'Success insert to table file',
             'isi'    => $bitEncryptSlektif),JSON_PRETTY_PRINT));
```

```
            }else
            {
               // echo "error";

print(json_encode(array("error"=>1,"message"=>"Unsuccessful.
Connection was successful but data could not be Inserted.")));
            }
            $con->close();
         }catch (Exception $e)
         {
            print(json_encode(array("message"=>"ERROR PHP
EXCEPTION : CAN'T SAVE TO MYSQL. ".$e->getMessage())));
            $con->close();
         }
      }else{
         print(json_encode(array("message"=>"ERROR PHP EXCEPTION :
CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
      }
   }
   public function InsertFile()
   {
      $con=$this->connect();
      if($con != null)
      {
         $password     =
mysqli_real_escape_string($con,$_POST['password']);
         $encryptedkunci=
mysqli_real_escape_string($con,$_POST['kuncifile']);
         $kunci     =AesCtr::encrypt($encryptedkunci,$password,256);
         $bit      = 256;
         $file_name = $_FILES['file']['name'];
```

```
        $file_type = $_FILES['file']['type'];
        $file_size = $_FILES['file']['size'];
        $id_user= (int)$_POST['id_user'];

        $isifile =file_get_contents($_FILES['file']['tmp_name']);
        $bit_selektif=substr($isifile, 0,999999);
        $length = 133344; //panjang hasil encryp
        if (!is_dir('encrypted/'.$id_user)) {
           mkdir('encrypted/'.$id_user, 0777, true);
        }
        $path= $this-
>cekFile("encrypted/".$id_user."/".basename($_FILES['file']['name']));
        $file_name = basename($path);



        $sql2 = "INSERT INTO
                tb_file (
                 id_file,
                 id_user,
                 nama,
                 type,
                 size,
                 file,
                 kunci,
                 length,
                 bit_file
                 )VALUES (
                        '',
                        $id_user,
                        '$file_name',
```

```
                    '$file_type',
                    '$file_size',
                    '$path',
                    '$kunci',
                    '$length',
                     $bit)
                     ";
        try
        {
           $result=$con->query($sql2);
           if($result)
           {

                $Newpath = fopen($path,"w");
                fwrite($Newpath,$isifile);
                fclose($Newpath);
                header('Content-Type: application/json');
                print(json_encode(array(
                   'error'  => "0",
                   'message'=> 'Success insert to table
file'),JSON_PRETTY_PRINT));


           }else
           {
              // echo "error";

print(json_encode(array("error"=>1,"message"=>"Unsuccessful.
Connection was successful but data could not be Inserted.")));
           }
           $con->close();
```

```
        }catch (Exception $e)
        {
           print(json_encode(array("error"=>1,"message"=>"ERROR PHP
EXCEPTION : CAN'T SAVE TO MYSQL. ".$e->getMessage())));
           $con->close();
        }
     }else{
        print(json_encode(array("error"=>1,"message"=>"ERROR PHP
EXCEPTION : CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
     }
  }
  public function select_all()
  {
     $con=$this->connect();
     if($con != null)
     {
        $id = (int) $_GET['id_user'];
        $password= $_GET['passwordsl'];
        $result=$con->query(Constants::$SELECT_ON_USER.$id);
        if($result->num_rows > 0)
        {
           $spiritual_teachers['fileku']=array();
           while($row=$result->fetch_array())
           {
              $kunci = AesCtr::decrypt($row['kunci'],$password,256);
              array_push($spiritual_teachers['fileku'], array(
                 "id_file"   =>$row['id_file'],
                 "id_user"   =>$row['id_user'],
                 "nama"     =>$row['nama'],
                 "size"     =>$this->formatBytes($row['size']),
                 "file"     =>$row['file'],
```

```
                "length"    =>$row['length'],
                "kunci"     =>$kunci,
                "bit_file"  =>$row['bit_file'])
              );
            }
            header('Content-Type: application/json');

print(json_encode($spiritual_teachers,JSON_PRETTY_PRINT));
        }else{
        }
        $con->close();
      }else{
        print(json_encode(array("PHP EXCEPTION : CAN'T CONNECT TO
MYSQL. NULL CONNECTION.")));
      }
   }
   public function getfileDecryptbyID()
   {
      $con = $this->connect();
      if ($con != null) {
        $id = $_GET['did'];
        $id_user = $_GET['id_user'];
        $password = $_GET['password'];
          $result = $con->query("SELECT *FROM tb_file WHERE id_file
= $id");
          if ($result->num_rows > 0) {
          $row = mysqli_fetch_array($result);
          $fileku =$row["file"];
          $kunciku= AesCtr::decrypt($row["kunci"],$password,256);
          $path   =$row["file"];
          $length =$row["length"];
```

```
            $isifile=file_get_contents($path);
            $findbitEncryptSelectif = substr($isifile, 0,$length);
            $bitDecryptSlektif=
AesCtr::decrypt($findbitEncryptSelectif,$kunciku,256);
            $bitDecryptAll = str_replace($findbitEncryptSelectif,
$bitDecryptSlektif, $isifile);
             if (!is_dir('decrypted/'.$id_user)) {
             mkdir('decrypted/'.$id_user, 0777, true);
            }
                $file = fopen('decrypted/'.$id_user.'/'.$row["nama"],'w');
               fwrite($file,$bitDecryptAll);
               fclose($file);

               $file = 'decrypted/'.$id_user.'/'.$row["nama"];

               $cek = file_exists($file);
               if ($cek)
               {
                  header('Content-Description: File Transfer');
                  header('Content-Type: application/octet-stream');
                  header('Content-Disposition: attachment;
filename='.basename($file));
                  header('Expires: 0');
                  header('Cache-Control: must-revalidate');
                  header('Pragma: public');
                  header('Content-Length: ' . filesize($file));

                  ob_clean();
                  flush();
                  readfile($file);
```

```
            }

        }else{
            print(json_encode(array("cannot execute query sql")));
        }
    }else{
        print(json_encode(array("PHP EXCEPTION : CAN'T CONNECT TO
MYSQL. NULL CONNECTION.")));
    }
  }

  public function getfileEncryptbyID()
  {
    $con = $this->connect();
    if ($con != null) {
        $id = $_GET['eidfile'];
        $password = $_GET['password'];
        $result = $con->query("SELECT *FROM tb_file WHERE id_file =
$id");

        if ($result->num_rows > 0) {

          $row = mysqli_fetch_array($result);
          $fileku =$row["file"];
          $kunciku=AesCtr::decrypt($row['kunci'],$password,256);
          $path   =$row["file"];
          $length =$row["length"];
          $isifile=file_get_contents($path);
              $cek = file_exists($path);
```

```
            if ($cek)
            {
               header('Location: '.$path);
            }

        }else{
            print(json_encode(array("cannot execute query sql")));
        }
      }else{
        print(json_encode(array("PHP EXCEPTION : CAN'T CONNECT TO
MYSQL. NULL CONNECTION.")));
      }
   }
   public function formatBytes($bytes)
   {
       $bytes = number_format($bytes / 1048576, 2);
       return $bytes;
   }
   public function handleRequest() {
      if (isset($_POST['username']) && isset($_POST['password']) &&
isset($_POST['nama']) ){
      $this->register();
      }
      elseif (isset($_POST['username']) && isset($_POST['password'])) {
        $this->login();
      }
      elseif (isset($_GET['did']) && isset($_GET['id_user']) &&
isset($_GET['password']) ) {
        // header('Location: encrypted/1/ab.mp4');
        $this->getfileDecryptbyID();
```

```
      }
      elseif (isset($_GET['eidfile']) && isset($_GET['id_user']) &&
isset($_GET['password']) ) {
         // header('Location: encrypted/1/ab.mp4');
        $this->getfileEncryptbyID();
      }
      elseif (isset($_POST['id_user']) && isset($_POST['kunci'])&&
isset($_POST['password']) ) {
         $this->InsertToEncrypt();
      }
      elseif( isset($_POST['id_user']) && isset($_POST['kuncifile'])&&
isset($_POST['password']) ){
         $this->InsertFile();
      }
      elseif (isset($_GET['id_user']) && isset($_GET['passwordsl'])) {
         $this->select_all();
      }
   }
}
$server=new Server();
$server->handleRequest();
```

Server.php

```
<?php
require 'aes.php';
require 'aesctr.php';
class Constants
{
   static $DB_SERVER="localhost";
   static $DB_NAME  ="crypto_dbs";
   static $USERNAME ="root";
   static $PASSWORD ="";
   static $SQL_SELECT_ALL="SELECT * FROM tb_file";
   static $SELECT_ON_USER = "SELECT
id_file,id_user,nama,type,size,kunci,file,length,bit_file FROM tb_file
WHERE id_user =";
}

class Server
{
   public function connect()
   {
      $con=new
mysqli(Constants::$DB_SERVER,Constants::$USERNAME,Constants::$PA
SSWORD,Constants::$DB_NAME);
      if($con->connect_error)
      {
         return null;
      }else
      {
         return $con;

      }
```

```
    }
    public function login(){

        $con=$this->connect();
        if($con != null)
        {
            $username =
mysqli_real_escape_string($con,$_POST['username']);
            $password =
mysqli_real_escape_string($con,$_POST['password']);
            $sql = "SELECT id,nama,username,password FROM tb_user
WHERE username = '$username'";
            $result=$con->query($sql);
            if ($result->num_rows > 0) {
                $row = mysqli_fetch_array($result);
                if (password_verify($password,$row["password"])) {

print(json_encode(array("id"=>$row["id"],"nama"=>$row["nama"],"use
rname"=>$row["username"],"succes"=> 1, "message" => "login
Succes")));
                }else{
                    print(json_encode(array("succes" => 0,"message" =>
"wrong password")));
                }
            }else{
                print(json_encode(array("succes" => 0,"message" => "wrong
password")));
            }
        }else{
            print(json_encode(array("message"=>"ERROR PHP EXCEPTION :
CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
```

```
    }

  }
  public function register(){
    $con=$this->connect();
    if($con != null)
    {
      $options = [
        'cost' => 10
        ];
      $username =
mysqli_real_escape_string($con,$_POST['username']);
      $password =
mysqli_real_escape_string($con,$_POST['password']);
      $nama = mysqli_real_escape_string($con,$_POST['nama']);
      // echo $nama;
      // die();

      $passwordhash =
password_hash($password,PASSWORD_DEFAULT,$options);


      $sql = "INSERT INTO tb_user (id,nama, username, password)
VALUES ('','$nama','$username','$passwordhash')";
      try {
        $register= $con->query($sql);
        if ($register) {
          print(json_encode(array("succes"=>
"1","message"=>"Register Success, Silahkan Login")));
        }else{
```

```
            print(json_encode(array("succes"=>
"0","message"=>"Unsuccessful. Connection was successful but data
could not be register to register.")));
          }

        } catch (Exception $e) {
          print(json_encode(array("message"=>"ERROR PHP
EXCEPTION : CAN'T SAVE TO MYSQL. ".$e->getMessage())));
          $con->close();
        }
      }else{
        print(json_encode(array("message"=>"ERROR PHP EXCEPTION :
CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
      }
   }
   public function cekFile($path){
        $val=0;
        if (file_exists($path)) {
          $fileName    =pathinfo($path, PATHINFO_FILENAME);
          $fileExtension=pathinfo($path, PATHINFO_EXTENSION);
          $fileDir     =pathinfo($path, PATHINFO_DIRNAME);
          $val=$val + 1;
          $fileName= $fileName."-".$val;
          $fileCompleteName = $fileName.".".$fileExtension;

          $path =$fileDir.'/'.$fileCompleteName;
          if (file_exists($path)) {
            cekFile($path,$val);
          }
        }
```

```
            return $path;

    }
    public function InsertToEncrypt()
    {
        $con=$this->connect();
        if($con != null)
        {
            $password      =
mysqli_real_escape_string($con,$_POST['password']);
            $plaintextkunci =
mysqli_real_escape_string($con,$_POST['kunci']);
            $kunci        = AesCtr::encrypt($plaintextkunci,$password,256);
            $bit          = 256;
            $file_name     = $_FILES['file']['name'];
            $file_type     = $_FILES['file']['type'];
            $file_size     = $_FILES['file']['size'];
            $id_user       = (int)$_POST['id_user'];
            $isifile       =file_get_contents($_FILES['file']['tmp_name']);
            $bit_selektif   =substr($isifile, 0,99999);
            $yangdiencrypt  =
AesCtr::encrypt($bit_selektif,$plaintextkunci,256);
            $length = strlen($yangdiencrypt);
            $allbitEncrypt = str_replace($bit_selektif, $yangdiencrypt,
$isifile);
            $bitEncryptSlektif= substr($allbitEncrypt, 0,1500);
            if (!is_dir('encrypted/'.$id_user)) {
                mkdir('encrypted/'.$id_user, 0777, true);
            }
            $path=$this-
>cekFile("encrypted/".$id_user."/".basename($_FILES['file']['name']));
```

```
$file_name=basename($path);
$sql2 = "INSERT INTO
        tb_file (
         id_file,
         id_user,
         nama,
         type,
         size,
         file,
         kunci,
         length,
         bit_file
         )VALUES (
                 '',
                 $id_user,
                 '$file_name',
                 '$file_type',
                 '$file_size',
                 '$path',
                 '$kunci',
                 '$length',
                  $bit)
                  ";
try
{
   $result=$con->query($sql2);
   if($result)
   {


         $Newpath = fopen($path,"w");
```

```
                fwrite($Newpath,$allbitEncrypt);
                fclose($Newpath);
                header('Content-Type: application/json');
                print(json_encode(array(
                   'error'  => "0",
                   'message'=> 'Success insert to table file',
                   'isi'    => $bitEncryptSlektif),JSON_PRETTY_PRINT));

            }else
            {
               // echo "error";

print(json_encode(array("error"=>1,"message"=>"Unsuccessful.
Connection was successful but data could not be Inserted.")));
            }
            $con->close();
        }catch (Exception $e)
        {
            print(json_encode(array("message"=>"ERROR PHP
EXCEPTION : CAN'T SAVE TO MYSQL. ".$e->getMessage())));
            $con->close();
        }
     }else{
        print(json_encode(array("message"=>"ERROR PHP EXCEPTION :
CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
     }
  }
  public function InsertFile()
  {
     $con=$this->connect();
     if($con != null)
```

```
    {
        $password     =
mysqli_real_escape_string($con,$_POST['password']);
        $encryptedkunci=
mysqli_real_escape_string($con,$_POST['kuncifile']);
        $kunci    =AesCtr::encrypt($encryptedkunci,$password,256);
        $bit      = 256;
        $file_name = $_FILES['file']['name'];
        $file_type = $_FILES['file']['type'];
        $file_size = $_FILES['file']['size'];
        $id_user= (int)$_POST['id_user'];

        $isifile =file_get_contents($_FILES['file']['tmp_name']);
        $bit_selektif=substr($isifile, 0,999999);
        $length = 133344; //panjang hasil encryp
        if (!is_dir('encrypted/'.$id_user)) {
           mkdir('encrypted/'.$id_user, 0777, true);
        }
        $path= $this-
>cekFile("encrypted/".$id_user."/".basename($_FILES['file']['name']));
        $file_name = basename($path);



        $sql2 = "INSERT INTO
                tb_file (
                 id_file,
                 id_user,
                 nama,
                 type,
                 size,
```

```
                file,
                kunci,
                length,
                bit_file
                )VALUES (
                        '',
                        $id_user,
                        '$file_name',
                        '$file_type',
                        '$file_size',
                        '$path',
                        '$kunci',
                        '$length',
                         $bit)
                         ";
        try
        {
           $result=$con->query($sql2);
           if($result)
           {

                $Newpath = fopen($path,"w");
                fwrite($Newpath,$isifile);
                fclose($Newpath);
                header('Content-Type: application/json');
                print(json_encode(array(
                   'error'  => "0",
                   'message'=> 'Success insert to table
file'),JSON_PRETTY_PRINT));
```

```
            }else
            {
               // echo "error";

print(json_encode(array("error"=>1,"message"=>"Unsuccessful.
Connection was successful but data could not be Inserted.")));
            }
            $con->close();
         }catch (Exception $e)
         {
            print(json_encode(array("error"=>1,"message"=>"ERROR PHP
EXCEPTION : CAN'T SAVE TO MYSQL. ".$e->getMessage())));
            $con->close();
         }
      }else{
         print(json_encode(array("error"=>1,"message"=>"ERROR PHP
EXCEPTION : CAN'T CONNECT TO MYSQL. NULL CONNECTION.")));
      }
   }
   public function select_all()
   {
      $con=$this->connect();
      if($con != null)
      {
         $id = (int) $_GET['id_user'];
         $password= $_GET['passwordsl'];
         $result=$con->query(Constants::$SELECT_ON_USER.$id);
         if($result->num_rows > 0)
         {
            $spiritual_teachers['fileku']=array();
            while($row=$result->fetch_array())
```

```
            {
               $kunci = AesCtr::decrypt($row['kunci'],$password,256);
               array_push($spiritual_teachers['fileku'], array(
                  "id_file"   =>$row['id_file'],
                  "id_user"   =>$row['id_user'],
                  "nama"      =>$row['nama'],
                  "size"      =>$this->formatBytes($row['size']),
                  "file"      =>$row['file'],
                  "length"    =>$row['length'],
                  "kunci"     =>$kunci,
                  "bit_file"  =>$row['bit_file'])
               );
            }
            header('Content-Type: application/json');

print(json_encode($spiritual_teachers,JSON_PRETTY_PRINT));
         }else{
         }
         $con->close();
      }else{
         print(json_encode(array("PHP EXCEPTION : CAN'T CONNECT TO
MYSQL. NULL CONNECTION.")));
      }
   }
   public function getfileDecryptbyID()
   {
      $con = $this->connect();
      if ($con != null) {
         $id = $_GET['did'];
         $id_user = $_GET['id_user'];
         $password = $_GET['password'];
```

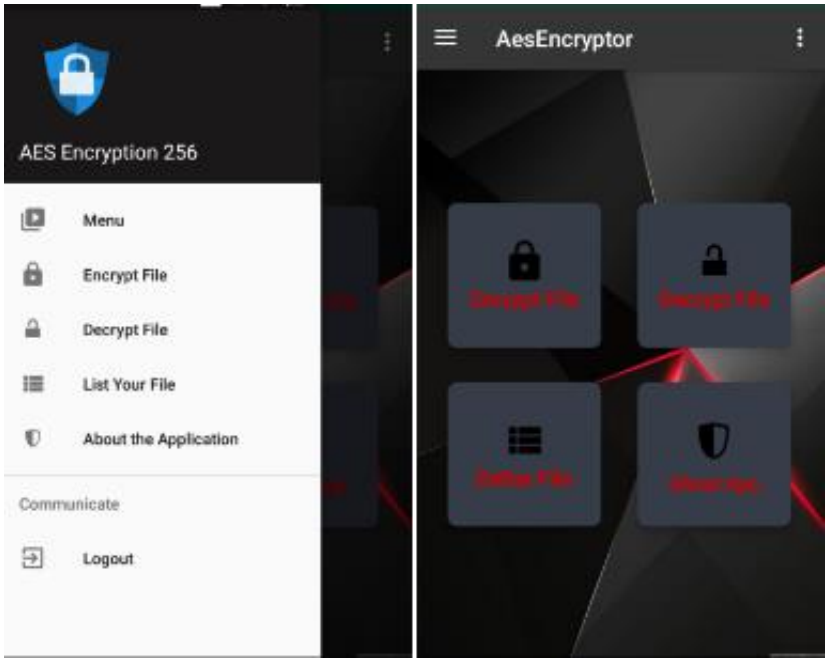
```
          $result = $con->query("SELECT *FROM tb_file WHERE id_file
= $id");
          if ($result->num_rows > 0) {
          $row = mysqli_fetch_array($result);
          $fileku =$row["file"];
          $kunciku= AesCtr::decrypt($row["kunci"],$password,256);
          $path   =$row["file"];
          $length =$row["length"];
          $isifile=file_get_contents($path);
          $findbitEncryptSelectif = substr($isifile, 0,$length);
          $bitDecryptSlektif=
AesCtr::decrypt($findbitEncryptSelectif,$kunciku,256);
          $bitDecryptAll = str_replace($findbitEncryptSelectif,
$bitDecryptSlektif, $isifile);
           if (!is_dir('decrypted/'.$id_user)) {
           mkdir('decrypted/'.$id_user, 0777, true);
          }
              $file = fopen('decrypted/'.$id_user.'/'.$row["nama"],'w');
             fwrite($file,$bitDecryptAll);
             fclose($file);

             $file = 'decrypted/'.$id_user.'/'.$row["nama"];

             $cek = file_exists($file);
             if ($cek)
             {
                header('Content-Description: File Transfer');
                header('Content-Type: application/octet-stream');
                header('Content-Disposition: attachment;
filename='.basename($file));
                header('Expires: 0');
```

```
                header('Cache-Control: must-revalidate');
                header('Pragma: public');
                header('Content-Length: ' . filesize($file));

                ob_clean();
                flush();
                readfile($file);



            }

        }else{
            print(json_encode(array("cannot execute query sql")));
        }
    }else{
        print(json_encode(array("PHP EXCEPTION : CAN'T CONNECT TO
MYSQL. NULL CONNECTION.")));
    }
  }

  public function getfileEncryptbyID()
  {
    $con = $this->connect();
    if ($con != null) {
        $id = $_GET['eidfile'];
        $password = $_GET['password'];
        $result = $con->query("SELECT *FROM tb_file WHERE id_file =
$id");

        if ($result->num_rows > 0) {
```

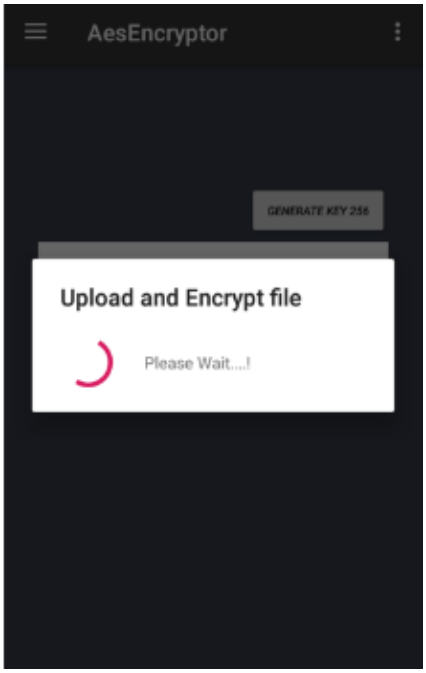
```
            $row = mysqli_fetch_array($result);
            $fileku =$row["file"];
            $kunciku=AesCtr::decrypt($row['kunci'],$password,256);
            $path   =$row["file"];
            $length =$row["length"];
            $isifile=file_get_contents($path);
                $cek = file_exists($path);

                if ($cek)
                {
                   header('Location: '.$path);
                }

          }else{
              print(json_encode(array("cannot execute query sql")));
          }
       }else{
          print(json_encode(array("PHP EXCEPTION : CAN'T CONNECT TO
MYSQL. NULL CONNECTION.")));
       }
    }
    public function formatBytes($bytes)
    {
        $bytes = number_format($bytes / 1048576, 2);
        return $bytes;
    }
    public function handleRequest() {
       if (isset($_POST['username']) && isset($_POST['password']) &&
isset($_POST['nama']) ){
       $this->register();
```

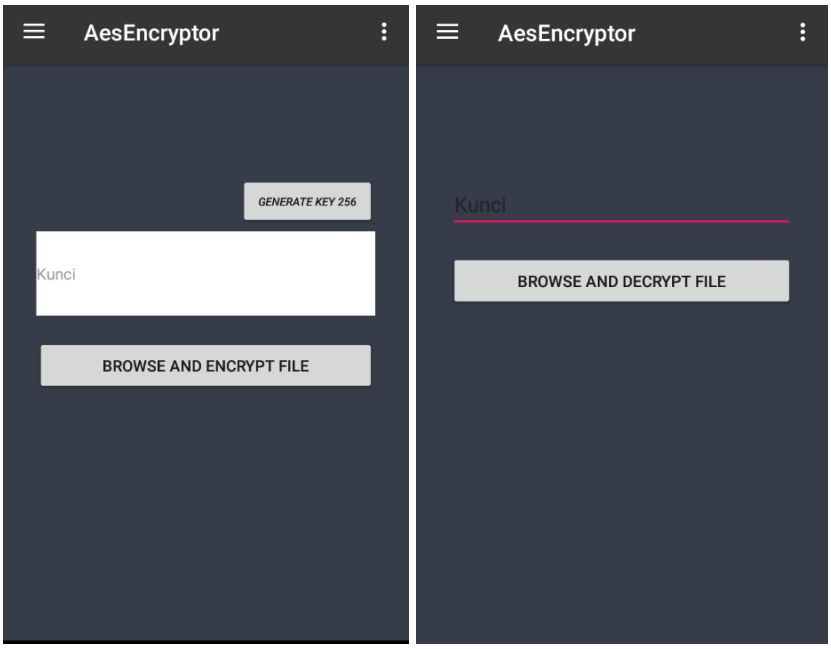
```
        }
        elseif (isset($_POST['username']) && isset($_POST['password'])) {
            $this->login();
        }
        elseif (isset($_GET['did']) && isset($_GET['id_user']) &&
isset($_GET['password']) ) {
            // header('Location: encrypted/1/ab.mp4');
            $this->getfileDecryptbyID();
        }
        elseif (isset($_GET['eidfile']) && isset($_GET['id_user']) &&
isset($_GET['password']) ) {
            // header('Location: encrypted/1/ab.mp4');
           $this->getfileEncryptbyID();
        }
        elseif (isset($_POST['id_user']) && isset($_POST['kunci'])&&
isset($_POST['password']) ) {
            $this->InsertToEncrypt();
        }
        elseif( isset($_POST['id_user']) && isset($_POST['kuncifile'])&&
isset($_POST['password']) ){
            $this->InsertFile();
        }
        elseif (isset($_GET['id_user']) && isset($_GET['passwordsl'])) {
            $this->select_all();
        }
    }
}
$server=new Server();
$server->handleRequest();
```

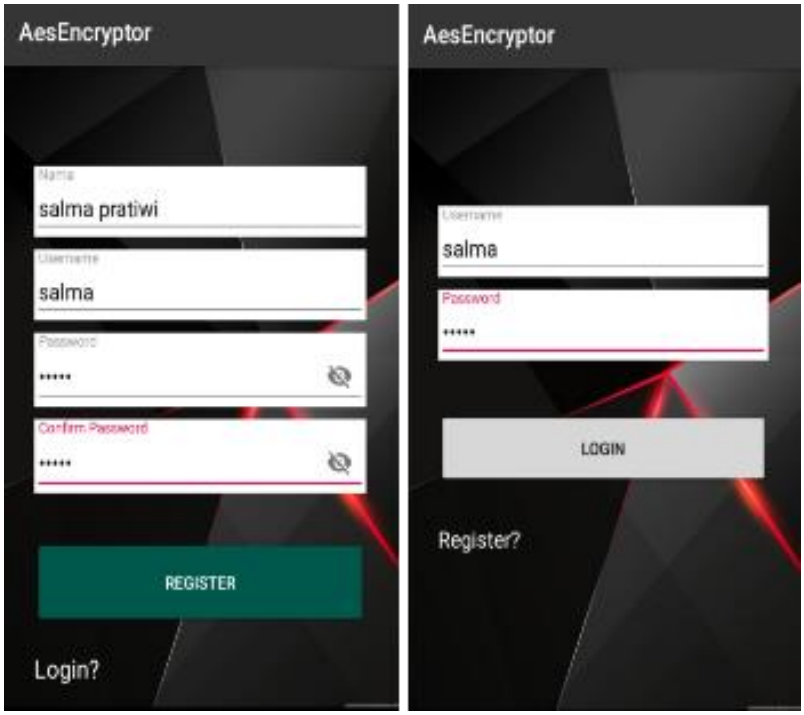

AesEncryptor
dudu.jpg
size original = 256154 Byte
size Encrypted = 289499 Byte
size Decrypted = 256154 Byte
Size Reduction = 156155 Byte
Bitstream Selektif= 99999 Byte
Speed Encryption = 2.780159 Second
Speed Decryption = 2.693815 Second
kunci= TUEBRO98R0BOKGKMTE58VB5MR9VT3BD7
Recording_3.mp3
size original = 1074718 Byte
size Encrypted = 1108063 Byte
size Decrypted = 1074718 Byte
Size Reduction = 974719 Byte
Bitstream Selektif= 99999 Byte
Speed Encryption = 2.682311 Second
Speed Decryption = 2.697951 Second
kunci= UMDLGHA9BF3DCPCR0VK6M4XCO6H5CVQC
AUD-20180831-WA0009.mp3
size original = 807407 Byte
size Encrypted = 840752 Byte
size Decrypted = 807407 Byte

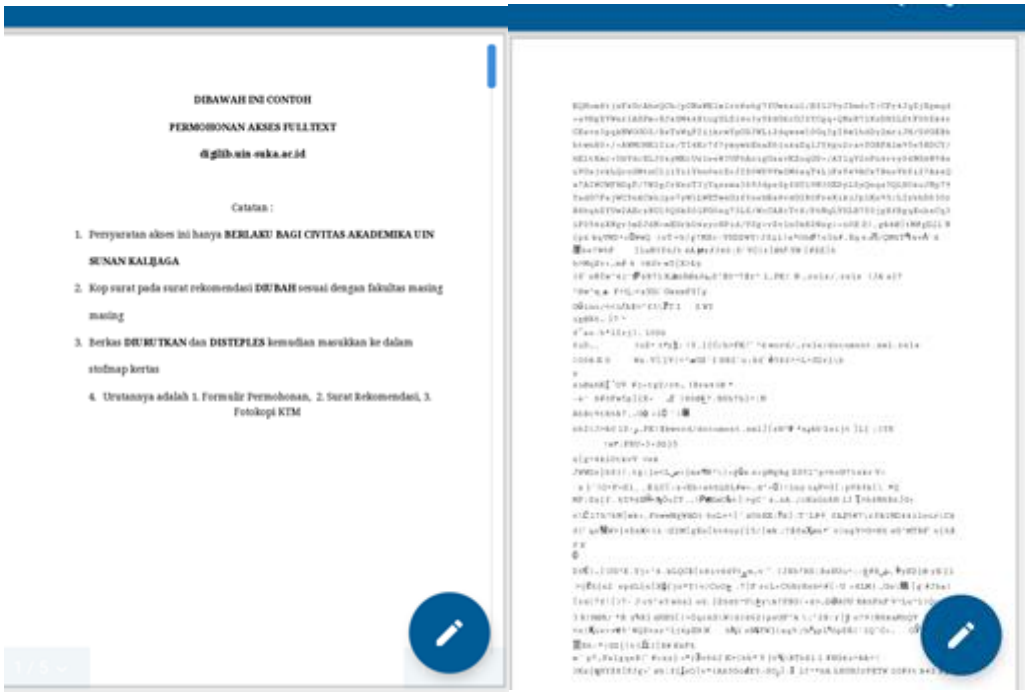

AES Encryption 256
Menu
Encrypt File
Decrypt File
List Your File
About the Application
Communicate
Logout
AesEncryptor


AesEncryptor
GENERATE KEY 256
6FM7RJ85K2Q0FYFI909HNCDQR9I16OM5
BROWSE AND ENCRYPT FILE


AesEncryptor
GENERATE KEY 256
Upload and Encrypt file
Please Wait....!


AesEncryptor
GENERATE KEY 256
Kunci
BROWSE AND ENCRYPT FILE
AesEncryptor
Kunci
BROWSE AND DECRYPT FILE


AesEncryptor
salma pratiwi
salma
REGISTER
Login?
AesEncryptor
salma
LOGIN
Register?



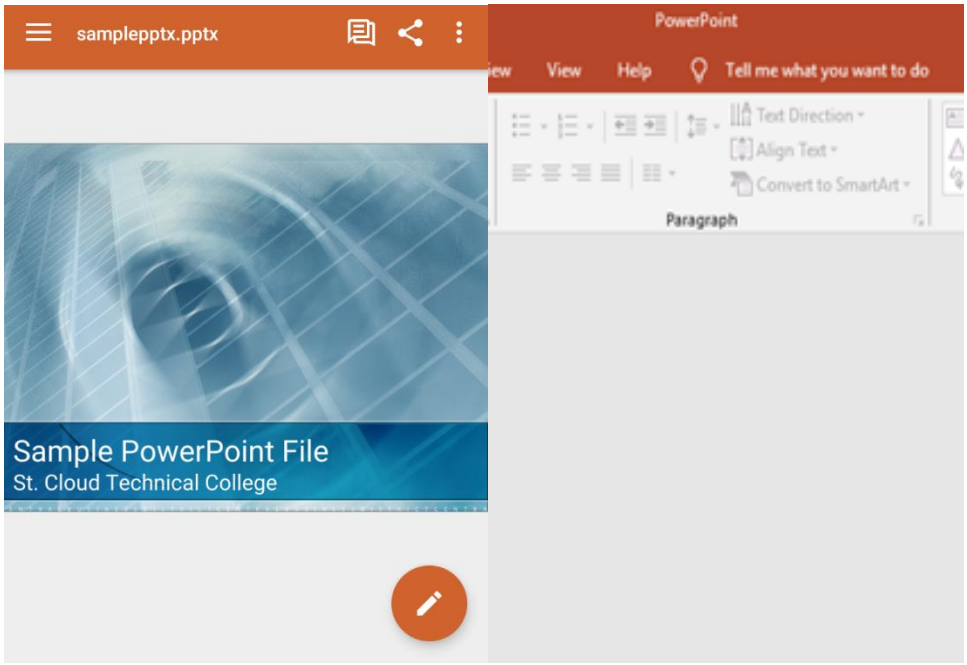
DIBAWAH INI CONTOH

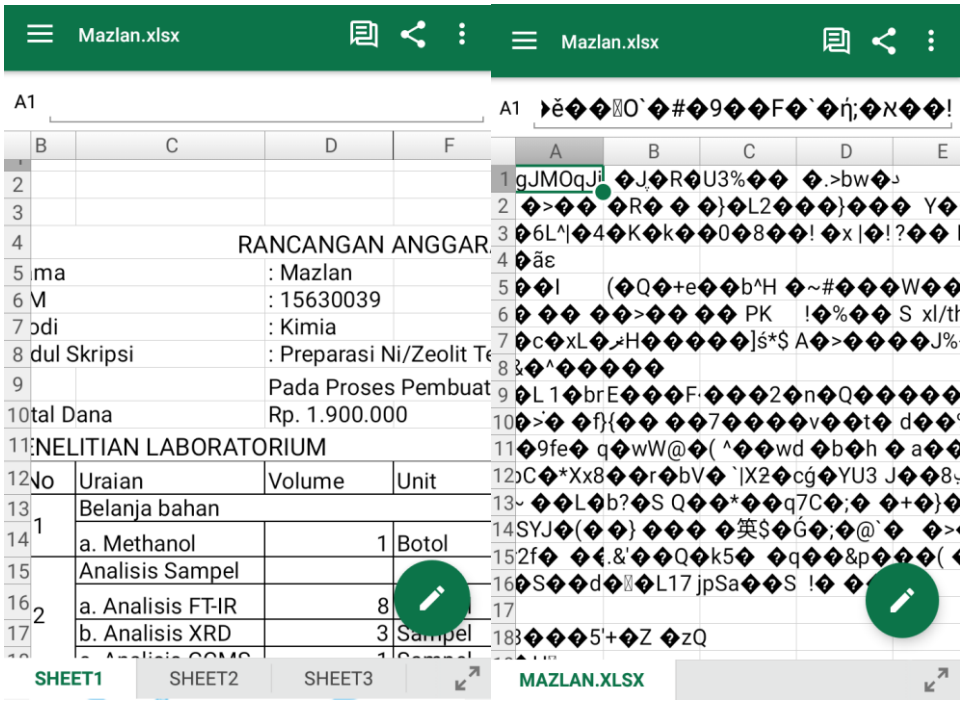
PERMOHONAN AKSES FULLTEXT

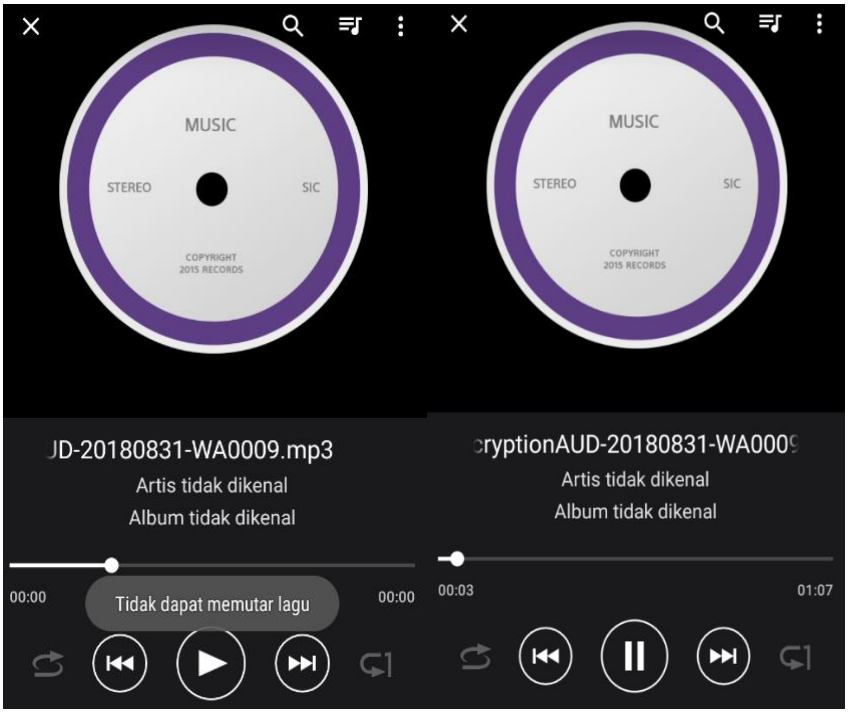
diglib.uin-suka.ac.id

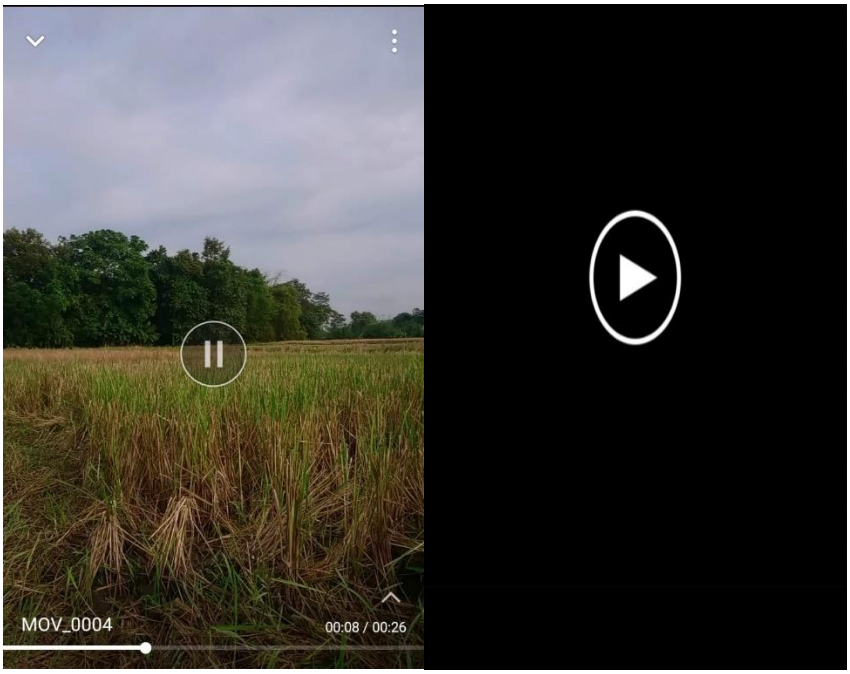
Catatan :

1. Persyaratan akses ini hanya **BERLAKU BAGI CIVITAS AKADEMIKA UIN SUNAN KALIJAGA**
2. Kop surat pada surat rekomendasi **DIUBAH** sesuai dengan fakultas masing masing
3. Berkas **DIURUTKAN** dan **DISTEPLES** kemudian masukkan ke dalam stofmap kertas
4. Urutannya adalah 1. Formulir Permohonan, 2. Surat Rekomendasi, 3. Fotokopi KTM


sampleppx.pptx
Sample PowerPoint File
St. Cloud Technical College
PowerPoint
View
Help
Tell me what you want to do
Text Direction
Align Text
Convert to SmartArt
Paragraph


Mazlan.xlsx
A1
RANCANGAN ANGGAR
: Mazlan
: 15630039
: Kimia
: Preparasi Ni/Zeolit Te
Pada Proses Pembuat
Rp. 1.900.000
NELITIAN LABORATORIUM
No
Uraian
Volume
Unit
Belanja bahan
a. Methanol
1
Botol
Analisis Sampel
a. Analisis FT-IR
8
b. Analisis XRD
3
SHEET1
SHEET2
SHEET3
Mazlan.xlsx
A1
MAZLAN.XLSX


MUSIC
STEREO
SIC
COPYRIGHT
2015 RECORDS
JD-20180831-WA0009.mp3
Artis tidak dikenal
Album tidak dikenal
00:00
Tidak dapat memutar lagu
00:00
MUSIC
STEREO
SIC
COPYRIGHT
2015 RECORDS
cryptionAUD-20180831-WA000
Artis tidak dikenal
Album tidak dikenal
00:03
01:07


MOV_0004
00:08 / 00:26

DAFTAR RIWAYAT HIDUP

Nama	: Didik Eko Pramono
Tempat,Tanggal Lahir	: Wonogiri, 28 Oktober 1997
Alamat Asal	: Kutukan, RT 23,RW 08, Bubakan,Kec.Girimarto, Kab.Wonogiri, Prov. Jawa Tengah
No HP	: +858 5819 1591
Email	: didikeko1997@gmail.com
Orang Tua	
Ayah	: Suripto
Ibu	: Katinem

Pendidikan Formal :

- SD Negeri 2 Bubakan (2006 - 2011)
- MTs Negeri 1 Wonogiri (20011 - 2013)
- SMK Ibu S.Soemoharmanto Jatipurno (2013 - 2015)
- UIN Sunan Kalijaga Yogyakarta (Angkatan 2015)