" **Implementasi Enkripsi AES dan Steganografi pada Aplikasi Windows**"

# **Disusun guna memenuhi UTS mata kuliah Kriptografi**

Dosen Pengampu: **Karsito, S.Kom., M.Kom.**

****

Oleh:

|  |  |
| --- | --- |
| Roana | **312210027** |

**KELAS TI.22.C.1**

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK**

**UNIVERSITAS PELITA BANGSA CIKARANG TAHUN 2024/2025**

**Implementasi Enkripsi dan Steganografi untuk Keamanan Data Digital**

**Abstrak** Makalah ini menjelaskan integrasi algoritma kriptografi modern, yaitu Advanced Encryption Standard (AES), dengan teknik steganografi digital dalam sebuah aplikasi desktop berbasis Windows Forms. Penggunaan kombinasi kedua metode ini bertujuan untuk meningkatkan keamanan data melalui enkripsi data yang kuat dan penyembunyian pesan pada media gambar. Aplikasi ini memungkinkan pengguna untuk melakukan enkripsi, dekripsi, serta menyembunyikan dan mengekstrak pesan secara efisien. Penelitian ini memberikan kontribusi signifikan pada bidang keamanan informasi dengan menawarkan pendekatan multi-layered untuk perlindungan data.

**1. Pendahuluan**

Dalam era digital yang semakin maju, ancaman terhadap keamanan informasi menjadi perhatian utama. Berbagai metode telah dikembangkan untuk melindungi data, salah satunya adalah melalui penggunaan kriptografi dan steganografi. Kriptografi berfungsi mengamankan data dengan mengubahnya menjadi format yang sulit dipahami tanpa kunci yang benar. Di sisi lain, steganografi menyembunyikan keberadaan data dengan menyisipkannya ke dalam media digital seperti gambar atau audio.

Makalah ini membahas implementasi algoritma AES untuk enkripsi dan dekripsi data serta teknik Least Significant Bit (LSB) steganografi untuk menyembunyikan pesan pada gambar. Kombinasi ini memberikan tingkat perlindungan yang lebih tinggi terhadap data sensitif.

**2. Advanced Encryption Standard (AES)**

AES adalah algoritma enkripsi simetris yang diadopsi sebagai standar oleh National Institute of Standards and Technology (NIST). AES dikenal dengan kecepatan dan keamanan yang tinggi, sehingga banyak digunakan untuk berbagai aplikasi, mulai dari komunikasi hingga penyimpanan data.

Algoritma ini bekerja dengan panjang blok data 128-bit dan mendukung panjang kunci 128, 192, dan 256-bit. Proses enkripsi melibatkan beberapa tahapan, seperti substitusi byte, pergeseran baris, pencampuran kolom, dan penambahan kunci (key schedule). Setiap tahapan dirancang untuk memastikan bahwa data terenkripsi sulit dipecahkan tanpa mengetahui kuncinya.

**3. Steganografi Least Significant Bit (LSB)**

LSB adalah teknik steganografi yang sederhana namun efektif. Teknik ini memanfaatkan bit-bit terkecil pada setiap piksel dalam gambar untuk menyembunyikan pesan. Karena perubahan pada LSB tidak mempengaruhi tampilan visual gambar secara signifikan, pesan dapat disembunyikan tanpa terdeteksi.

Proses menyembunyikan pesan melibatkan konversi pesan menjadi representasi biner, yang kemudian disisipkan pada LSB setiap piksel. Untuk mengekstrak pesan, bit-bit ini dibaca kembali dan dikonversi ke dalam bentuk teks.

**4. Implementasi Program**

Aplikasi ini dikembangkan menggunakan bahasa pemrograman C# dengan antarmuka grafis berbasis Windows Forms. Struktur utama aplikasi terdiri dari:

* **Fungsi Enkripsi dan Dekripsi:** Memanfaatkan algoritma AES untuk mengenkripsi dan mendekripsi teks atau file.
* **Fungsi Steganografi:** Menyisipkan pesan ke dalam gambar menggunakan teknik LSB dan mengekstrak pesan dari gambar.
* **Antarmuka Pengguna:** Menyediakan tombol dan form input untuk memudahkan interaksi pengguna.

**5. Panduan Penggunaan Aplikasi**

1. **Enkripsi File:**
   * Pilih file yang akan dienkripsi.
   * Masukkan kunci enkripsi.
   * Klik tombol "Encrypt" untuk menghasilkan file terenkripsi.
2. **Dekripsi File:**
   * Pilih file terenkripsi.
   * Masukkan kunci dekripsi.
   * Klik tombol "Decrypt" untuk mendapatkan file asli.
3. **Menyembunyikan Pesan:**
   * Pilih gambar yang akan digunakan.
   * Masukkan pesan yang akan disembunyikan.
   * Klik tombol "Hide Message" untuk menyimpan gambar dengan pesan tersembunyi.
4. **Mengekstrak Pesan:**
   * Pilih gambar yang mengandung pesan tersembunyi.
   * Klik tombol "Extract Message" untuk membaca pesan tersembunyi.

**6. Keunggulan dan Batasan**

**Keunggulan:**

* Menggabungkan dua metode keamanan (kriptografi dan steganografi) dalam satu aplikasi.
* Antarmuka pengguna yang mudah digunakan.
* Dukungan untuk berbagai jenis file dan format gambar.

**Batasan:**

* Ukuran pesan yang dapat disisipkan bergantung pada resolusi gambar.
* Proses enkripsi dan dekripsi membutuhkan kunci yang benar, sehingga pengguna harus menyimpan kunci dengan aman.

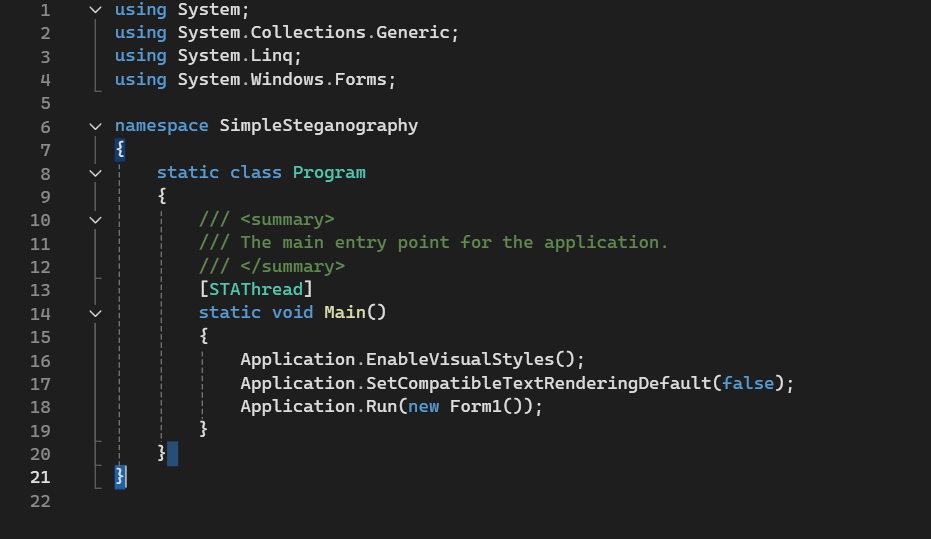
**7. Kesimpulan**

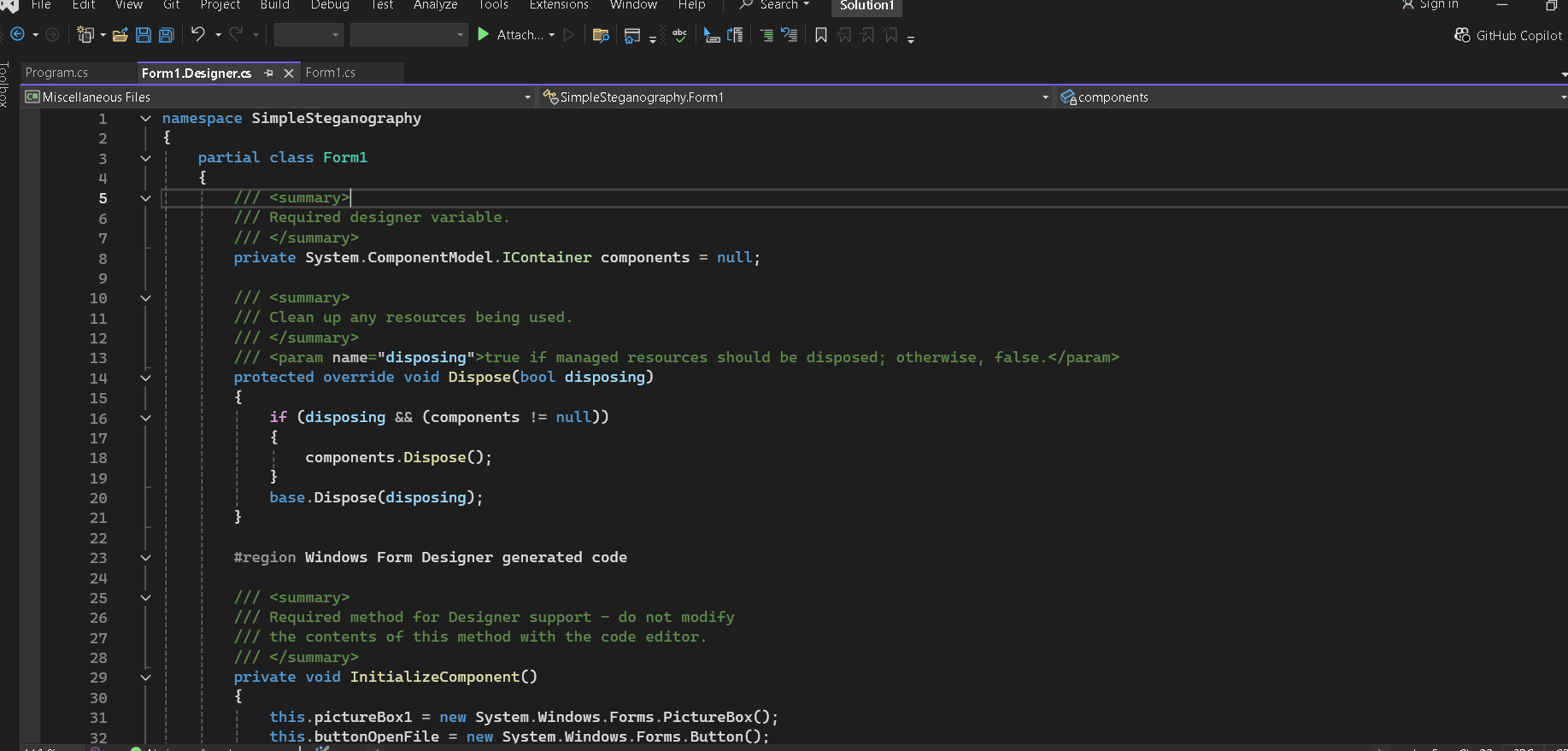
Aplikasi ini berhasil mengintegrasikan algoritma AES dan teknik steganografi LSB untuk meningkatkan keamanan data digital. Dengan menggunakan aplikasi ini, pengguna dapat mengenkripsi data dan menyembunyikan pesan dengan cara yang efisien dan aman. Pengembangan lebih lanjut dapat mencakup peningkatan algoritma steganografi untuk mendukung format gambar lainnya dan menambahkan fitur tambahan seperti pengamanan multi-layer.

**Referensi**

1. National Institute of Standards and Technology (NIST). (2001). Announcing the Advanced Encryption Standard (AES).
2. Katzenbeisser, S., & Petitcolas, F. A. P. (2000). Information Hiding Techniques for Steganography and Digital Watermarking. Artech House.
3. Roana28. (2025). UAS-kriptografi. GitHub repository.

<https://github.com/Roana28/UAS-kriptografi>





using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace SimpleSteganography

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void buttonOpenFile\_Click(object sender, EventArgs e)

{

OpenFileDialog openDialog = new OpenFileDialog();

openDialog.Filter = "Image Files (\*.png, \*.jpg) | \*.png; \*.jpg";

openDialog.InitialDirectory = @"C:\Users\metech\Desktop";

if (openDialog.ShowDialog() == DialogResult.OK)

{

textBoxFilePath.Text = openDialog.FileName.ToString();

pictureBox1.ImageLocation = textBoxFilePath.Text;

}

}

private void buttonEncode\_Click(object sender, EventArgs e)

{

Bitmap img = new Bitmap(textBoxFilePath.Text);

for (int i = 0; i < img.Width; i++)

{

for (int j = 0; j < img.Height; j++)

{

Color pixel = img.GetPixel(i, j);

if (i < 1 && j < textBoxMessage.TextLength)

{

Console.WriteLine("R = [" + i + "][" + j + "] = " + pixel.R);

Console.WriteLine("G = [" + i + "][" + j + "] = " + pixel.G);

Console.WriteLine("G = [" + i + "][" + j + "] = " + pixel.B);

char letter = Convert.ToChar(textBoxMessage.Text.Substring(j, 1));

int value = Convert.ToInt32(letter);

Console.WriteLine("letter : " + letter + " value : " + value);

img.SetPixel(i, j, Color.FromArgb(pixel.R, pixel.G, value));

}

if (i == img.Width - 1 && j == img.Height - 1)

{

img.SetPixel(i, j, Color.FromArgb(pixel.R, pixel.G, textBoxMessage.TextLength));

}

}

}

SaveFileDialog saveFile = new SaveFileDialog();

saveFile.Filter = "Image Files (\*.png, \*.jpg) | \*.png; \*.jpg";

saveFile.InitialDirectory = @"C:\Users\metech\Desktop";

if (saveFile.ShowDialog() == DialogResult.OK)

{

textBoxFilePath.Text = saveFile.FileName.ToString();

pictureBox1.ImageLocation = textBoxFilePath.Text;

img.Save(textBoxFilePath.Text);

}

}

private void buttonDecode\_Click(object sender, EventArgs e)

{

Bitmap img = new Bitmap(textBoxFilePath.Text);

string message = "";

Color lastpixel = img.GetPixel(img.Width - 1, img.Height - 1);

int msgLength = lastpixel.B;

for (int i = 0; i < img.Width; i++)

{

for (int j = 0; j < img.Height; j++)

{

Color pixel = img.GetPixel(i, j);

if (i < 1 && j < msgLength)

{

int value = pixel.B;

char c = Convert.ToChar(value);

string letter = System.Text.Encoding.ASCII.GetString(new byte[] { Convert.ToByte(c) });

message = message + letter;

}

}

}

textBoxMessage.Text = message;

}

}

}