

**SISTEM KEAMANAN CITRA DIGITAL MENGGUNAKAN
ALGORITMA *SPRITZ***

SKRIPSI

IMAM ZAKI HUSEIN NASUTION

71153010



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

**SISTEM KEAMANAN CITRA DIGITAL MENGGUNAKAN
ALGORITMA *SPRITZ***

SKRIPSI

Diajukan untuk Memenuhi syarat Mencapai Gelar Sarjana Komputer

IMAM ZAKI HUSEIN NASUTION

71153010



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth,

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Warahmatullahi Wabarakathu

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, kami selaku pembimbing berpendapat skripsi saudara,

Nama : Imam Zaki Husein Nasution

NIM : 71153010

Program Studi : Ilmu Komputer

Judul : Sistem Keamanan Citra Digital Menggunakan Algoritma *Spritz*

Dapat disetujui untuk segera *dimunaqasyakan*. Atas perhatiannya kami ucapkan terima kasih.

Medan, 23 Juli 2020 M

02 Dzulhijah 1441 H

Komisi Pembimbing,

Pembimbing I,

Pembimbing II,

Dr. Mhd. Furqan, S.Si., M.Comp.sc.
NIP. 198008062006041003

Rakhmat Kurniawan R, S.T., M.Kom
NIP. 198503162015031003

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Imam Zaki HuseinNasution
Nim : 71153010
Program Studi : Ilmu Komputer
Judul : Sistem Keamanan Citra Digital Menggunakan
Algoritma *Spritz*

Menyatakan dengan sebenarnya bahwa skripsi yang saya serahkan ini benar-benar merupakan hasil karya saya sendiri, kecuali kutipan-kutipan dari ringkasan- ringkasan yang semuanya saya jelaskan sumbernya.

Apabila kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil ciplakan, maka gelar dan ijazah yang diberikan oleh institut batal saya terima.

Medan, 23 Juli 2020

Materai

6000

Imam Zaki Husein Nasution
NIM. 71153010



KEMENTERIAN AGAMA REPUBLIK INDONESIA UNIVERSITAS
ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl. IAIN No. 1 Medan 20235
Telp. (061) 6615683-6622925, Fax. (061) 6615683
Url: <https://saintek.unsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor: 114/ST/ST.V.2/PP.01.1/01/2021

Judul : Sistem Keamanan Citra Digital
Menggunakan Algoritma *Spritz*
Nama : Imam Zaki Husein Nasution
Nomor Induk Mahasiswa : 71153010
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan di hadapan Dewan Penguji Skripsi Program Studi Ilmu
Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan
dinyatakan **LULUS**.

Pada hari/ Tanggal : Senin, 23 Juli 2020
Tempat : Via Zoom Meeting

Tim Ujian Munaqasyah,
Ketua,

Dr. Mhd. Furqan, S.Si, M.Comp.Sc.
NIP. 198008062006041003

Dewan Penguji,

Penguji I,

Penguji II,

Dr. Mhd. Furqan, S.Si, M.Comp.Sc.
NIP. 198008062006041003

Rakhmat Kurniawan R, S.T., M.Kom
NIP. 198503162015031003

Penguji III,

Penguji IV,

Sriani, M.Kom
NIB. 1100000108

Abdul Halim Hasugian, M.Kom
NIB. 1100000113

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan,

Dr. H.M Jamil, M.A.
NIP. 196609101999031002

ABSTRAK

Citra digital yang bersifat pribadi dan rahasia sangat rentan terhadap penyadapan oleh pihak-pihak yang tidak bertanggung jawab. Terutama bila didistribusikan melalui jaringan internet seperti pada aplikasi berbasis *chatting facebook, whatsapp* dan media *e-mail*. Citra yang dikirim kadang sering merupakan citra yang bersifat rahasia dan harus dijaga keamanannya. Demi menjaga keamanan citra digital dapat dilakukan dengan pemanfaatan teknik kriptografi. Teknik kriptografi dapat mengamankan citra digital dengan merubah nilai-nilai *pixel* dari citra digital sehingga menghasilkan nilai *pixel* yang berbeda dari citra asli yang akan diamankan. Penelitian ini akan menggunakan algoritma *Spritz* untuk enkripsi dan dekripsi citra digital pada aplikasi keamanan citra. Hasil dari enkripsi Algoritma *Spritz* menghasilkan nilai-nilai *pixel* yang jauh berbeda dengan nilai-nilai *pixel* aslinya. Sehingga citra hasil enkripsi lebih aman untuk didistribusikan melalui media perantara.

Kata Kunci: *kriptografi, citra, Spritz*

ABSTRACT

Personal and confidential digital images are very vulnerable to eavesdropping by irresponsible parties. Especially when distributed through internet networks such as Facebook-based chat applications, WhatsApp and e-mail media. The image that is sent sometimes is often an image that is confidential and must be kept safe. For the sake of maintaining digital image security can be done by using cryptographic techniques. Cryptography techniques can secure digital images by changing the pixel values of digital images so as to produce pixel values that are different from the original image to be secured. This study will use the Spritz algorithm for encryption and decryption of digital images in image security applications. The results of the Spritz Algorithm encryption produce pixel values that are far different from the original pixel values. So that the encrypted image is safer to be distributed through intermediary media.

Keywords: cryptography, image, Spritz

KATA PENGANTAR

Tiada puja dan puji syukur yang pantas dilantunkan oleh penulis selain kepada Allah SWT yang tidak pernah berhenti memberikan segala nikmat dan hidaya sehingga dengan ridho-Nya penulis dapat menyelesaikan tugas akhir ini dengan judul “Sistem Keamanan Citra Digital Menggunakan Algoritma Spritz”. Shalawat serta salam tidak lupa safaat ialah Nabi Muhammad SAW beserta keluarga, sahabat dan para umatnya.

Pada kesempatan ini, dengan segala kerendahan hati penulis juga menyampaikan ucapan terimakasih kepada :

1. Bapak Prof. Dr. Saidurrahman, M.Ag, selaku Rektor Universitas Islam Negeri Sumatera Utara Medan.
2. Bapak H.M Jamil, M.A, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
3. Bapak Dr. Mhd. Furqan S.Si., M.Comp., Sc, selaku Ketua Program Studi Ilmu Komputer dan Pembimbing Skripsi I. Semoga selalu memberikan yang terbaik untuk generasi ilmu komputer.
4. Bapak Rakhmat Kurniawan R, ST, M.Kom, selaku pembimbing II yang telah membimbing penulis selama menyelesaikan skripsi ini. Semoga selalu memberikan yang terbaik untuk generasi ilmu komputer.
5. Bapak dan Ibu dosen yang telah mendidik dan membimbing penulis dalam mengerjakan skripsi ini.
6. Kepada kedua orangtua saya Ayah dan Ibu serta keluarga dan seluruh teman-teman saya yang telah memberikan dukungan, motivasi dan semangat dalam penyelesaian skripsi ini.
7. Teman-teman seperjuangan yang telah membantu dalam menyelesaikan tugas akhir ini. Semoga kita selalu di rakhmati Allah SWT.
8. Serta semua pihak memberikan bantuan tulus dan dukungan dalam penyusunan tugas akhir ini yang tidak tersebutkan satu persatu.

Penulis menyadari bahwa skripsi ini belum sempurna dan masih banyak kekurangan. Semoga hasil dari skripsi ini ada manfaatnya bagi pihak yang berkepentingan dan kesalahan dalam penulisan skripsi, sepenuhnya berasal dari penulis sendiri.

Medan, Februari 2020
Penulis,

Imam Zaki Husein Nasution
NIM. 71153010

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
KATA PENGANTAR	iii
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Perumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA	4
2.1. Keamanan	4
2.2. Kriptografi	4
2.2.1. Aspek Keamanan Kriptografi.....	5
2.2.2. Serangan Terhadap Kriptografi	6
2.2.3. Pengelompokkan Algoritma Kriptografi	6
2.3. Algoritma <i>Spritz</i>	8
2.4. Citra Digital	10
2.4.1. Pengertian Citra Digital.....	10
2.4.2. Format File Citra	12
2.5. Flowchart	12
2.6. Bahasa Pemrograman Visual Basic 2010	14
BAB III METODE PENELITIAN	17

3.1. Tahapan Penelitian	17
3.2. Metode Pengumpulan Data.....	18
3.3. Analisa Sistem	18
3.3.1. Analisa Kebutuhan Sistem.....	21
3.4. Penerapan	22
BAB IV HASIL DAN PEMBAHASAN.....	23
4.1. Proses Enkripsi Manual Algoritma Spritz	23
4.2. Proses Dekripsi Manual Algoritma Spritz	34
4.3. Perancangan Interface	36
4.4. Flowchart Sistem	40
4.5. Tampilan Program Aplikasi	46
4.5.1. Tampilan Input Program Aplikasi	46
4.5.2. Proses Enkripsi Pada Aplikasi	49
4.5.3. Proses Dekripsi Pada Aplikasi	54
4.6. Hasil Pengujian Program Aplikasi	58
BAB V KESIMPULAN DAN SARAN	60
5.1. Kesimpulan	60
5.2. Saran	60
DAFTAR PUSTAKA	61

DAFTAR TABEL

Tabel 2.1 Simbol Penghubung/Alur (Flow Direction System)	13
Tabel 2.2 Simbol Proses (Processing Symbols).....	13
Tabel 2.3 Simbol Input-Output.....	14
Tabel 3.1 Kebutuhan Hardware.....	22
Tabel 4.1 Nilai RGB Pixel Sampel.....	23
Tabel 4.2 Larik State Awal.....	24
Tabel 4.3 Kode ASCII.....	25
Tabel 4.4 Hasil Key Scheduling Algorithm.....	27
Tabel 4.5 Nilai RGB Pixel	28
Tabel 4.6 Proses Nilai Pixel Cipherimage	33
Tabel 4.7 Pengujian Enkripsi Citra Plainimage.....	36
Tabel 4.8 Pengujian Dekripsi Citra Cipherimage.....	58
Tabel 4.9 Pengujian Dekripsi Citra <i>Cipherimage</i>	59

DAFTAR GAMBAR

Gambar 2.1 Diagram enkripsi dan dekripsi kunci simetri	7
Gambar 2.2 Diagram enkripsi dan dekripsi kunci simetri	8
Gambar 2.3 Fungsi Hash.....	8
Gambar 2.4 Ilustrasi Citra digital	11
Gambar 2.5 Posisi Koordinat Citra Digital.....	11
Gambar 2.6 Tampilan IDE Visual Basic 2010	15
Gambar 3.1 Tahapan Penelitian	17
Gambar 3.2 Diagram Proses Pengamanan Citra	19
Gambar 3.3 Diagram Enkripsi Spritz.....	20
Gambar 3.4 Diagram Dekripsi Spritz.....	21
Gambar 4.1 Plainimage Sampel 2 x 1 Pixel	23
Gambar 4.2 Kode ASCII.....	36
Gambar 4.3 Rancangan Menu Utama	37
Gambar 4.4 Rancangan Form Enkripsi.....	38
Gambar 4.5 Rancangan Form Dekripsi	39
Gambar 4.6 Rancangan Form Cara Pemakaian.....	40
Gambar 4.7 Rancangan Form Tentang Penulis	41
Gambar 4.8 Flowchart Sistem	43
Gambar 4.9 Flowchart Enkripsi	44
Gambar 4.10 Flowchart Dekripsi	45
Gambar 4.11 Flowchart Cara Pemakaian	46
Gambar 4.12 Flowchart About Me.....	47
Gambar 4.13 Tampilan Menu Utama.....	47
Gambar 4.14 Tampilan Form Enkripsi	48
Gambar 4.15 Tampilan Form Dekripsi	49
Gambar 4.16 Tampilan Cara Pemakaian	49
Gambar 4.17 Tampilan About Me	50
Gambar 4.18 Tampilan Pemilihan Citra Plainimage	50
Gambar 4.19 Tampilan Citra Plainimage	51

Gambar 4.20 Tampilan proses memasukan kunci.....	52
Gambar 4.21 Tampilan Proses Enkripsi Spritz	52
Gambar 4.22 Tampilan Proses Enkripsi Spritz	53
Gambar 4.23 Tampilan Proses Enkripsi Spritz Selesai.....	53
Gambar 4.24 Tampilan Proses Simpan Cipherimage	54
Gambar 4.25 Tampilan Pemilihan Citra cipherimage.....	55
Gambar 4.26 Tampilan Citra cipherimage	55
Gambar 4.27 Tampilan proses memasukan kunci.....	56
Gambar 4.28 Tampilan Proses Dekripsi Spritz	56
Gambar 4.29 Tampilan Proses Dekripsi Spritz Selesai	57
Gambar 4.30 Tampilan Proses Simpan Plainimage	57
Gambar 4.31 Hasil Enkripsi dan Dekripsi Citra Digital	57

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Listing Program
2.	Daftar Riwayat Hidup
3.	Kartu Bimbingan Skripsi

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Citra digital yang bersifat pribadi dan rahasia sangat rentan terhadap peyadapan oleh pihak-pihak yang tidak bertanggung jawab. Terutama bila didistribusikan melalui jaringan internet seperti pada aplikasi berbasis *chatting facebook, whatsapp* dan media *e-mail*. Citra yang dikirim melalui jaringan internet rawan terhadap penyerangan dan penyadapan, serta penyimpanan yang dilakukan didalam media *stroge* rawan terhadap pengaksesan oleh orang-orang yang tidak memiliki wewenang (Munir, 2010). Karena citra digital masih berupa citra yang dapat dikenali dan dapat dimanfaatkan oleh pihak penyadap untuk keuntungan pribadi sehingga merugikan pihak yang memiliki akses terhadap data citra. Tindakan penyadapan data citra tersebut dapat diminalisir dengan aplikasi teknik enkripsi kriptografi.

Didalam Al-qur'an terdapat ayat yang membahas tentang mencuri hak orang lain.

وَالسَّارِقُ وَالسَّارِقَةُ فَاقْطَعُوا أَيْدِيَهُمَا جَزَاءً بِمَا كَسَبَا نَكَالًا مِنَ اللَّهِ وَاللَّهُ عَزِيزٌ حَكِيمٌ

“Laki-laki yang mencuri dan perempuan yang mencuri, potonglah tangan keduanya (sebagai) pembalasan bagi apa yang mereka kerjakan dan sebagai siksaan dari Allah. Dan Allah Maha Perkasa lagi Maha Bijaksana” (QS. Al Maidah: 38).

Algoritma *spritz* merupakan varian dari algoritma RC4 yang dilakukan oleh Ron Rivest dan Jacob Schultz pada tahun 2014. Algoritma ini menghasilkan *sponge-base construction* dalam menghasilkan kunci dalam proses enkripsi dan dekripsi. Algoritma ini bekerja berdasarkan konsep *stream cipher* yaitu enkripsi satu per satu. Salah satu kelebihan dari algoritma ini adalah proses pembangkitan kunci yang digunakan dalam proses enkripsi maupun dekripsi (Zebua, 2018)

Penambahan elemen w yang relatif prima dengan nilai N pada proses *pseudo-random generation algorithm* adalah perbedaan algoritma ini dengan RC4. Selain sebagai *stream cipher*, algoritma *spritz* dapat juga digunakan sebagai *hash function* (fungsi *hash*) dan *Message Authentication Code* (MAC) dengan menggunakan *sponge function* dalam mengamankan data. Kunci yang dibangkitkan berikutnya selalu bergantung pada aliran kunci sebelumnya. Tingginya kompleksitas dari performansi algoritma *spritz* menyebabkan rumitnya para kriptanalis untuk menemukan kunci dan memecahkan algoritma ini.

Penelitian ini akan mengamankan citra digital dengan metode *Spritz* untuk enkripsi dan dekripsi citra digital. Hasil dari enkripsi algoritma *Spritz* menghasilkan *cipherimage* dengan nilai-nilai *pixel* yang jauh berbeda dengan nilai-nilai *pixel plainimage*. Format citra yang digunakan berupa .jpg dan .bmp untuk proses *input* dan format .bmp berupa hasil *output*.

Berdasarkan uraian latar belakang di atas, maka penulis mengambil judul dalam penelitian ini adalah “**Sistem Keamanan Citra Digital Menggunakan Algoritma Spritz**”.

1.2. Perumusan Masalah

Berdasarkan uraian dalam latar belakang, maka rumusan masalah dalam penelitian ini adalah :

1. Bagaimana menerapkan algoritma *Spritz* dalam mengamankan citra digital ?
2. Bagaimana merancang dan membangun aplikasi yang dapat menjaga kerahasiaan informasi yang terkandung pada citra digital menggunakan algoritma *Spritz* ?

1.3. Batasan Masalah

Berdasarkan dari paparan rumusan masalah di atas, maka batasan masalah dalam penelitian ini adalah :

1. Citra digital yang akan dienkripsi berupa gambar (*image*) berwarna dengan format *jpg*, dan *bmp*.
2. Pengamanan citra digital dienkripsi dengan menentukan nilai warna RGB dari

citra.

3. *Software* yang digunakan dalam merancang dan membangun aplikasi adalah *software* bahasa pemrograman *Microsoft VB Net 2010*.
4. Kunci enkripsi dan dekripsi yang digunakan dengan panjang 4 karakter (*4 byte*).
5. Hasil enkripsi akan berubah menjadi *.bmp* dengan format citra masukan *.jpg* ataupun *.bmp*.
6. Citra yang digunakan adalah citra 24 bit.

1.4. Tujuan Penelitian

Berdasarkan paparan dari perumusan dan batasan masalah di atas, adapun tujuan dari penelitian ini antara lain :

1. Untuk menerapkan algoritma *Spritz* dalam mengamankan citra digital.
2. Untuk merancang dan membangun aplikasi yang dapat menjaga kerahasiaan informasi yang terkandung pada citra digital menggunakan algoritma *Spritz*.

1.5. Manfaat Penelitian

Berdasarkan paparan dari tujuan di atas, adapun manfaat yang diberikan dalam penelitian ini adalah :

1. Mengurangi terjadinya manipulasi pada informasi yang terkandung pada citra digital.
2. Memberikan informasi bagaimana cara kerja dari algoritma *Spritz* dalam menenkripsi citra digital.
3. Untuk menjaga keaslian dan keterjaminan sebuah citra digital dari ancaman yang ada.

BAB II

TINJAUAN PUSTAKA

2.1. Keamanan

Masalah keamanan merupakan salah satu aspek terpenting pada sebuah sistem informasi. Masalah keamanan sering kali kurang mendapatkan perhatian dari para perancang dan pengolahan sistem informasi serta berada di urutan setelah tampilan, atau bahkan di urutan terakhir dalam daftar yang dianggap penting. Apabila mengganggu performa sistem, sering kali masalah keamanan tidak begitu dipedulikan bahkan ditiadakan. Keamanan data adalah kebebasan dari bahaya atau sebagai kondisi keselamatan. Keamanan secara rinci adalah perlindungan data di dalam sesuatu sistem melawan terhadap otorisasi tidak sah, modifikasi, atau perusakan dan perlindungan sistem komputer terhadap pengguna yang tidak bertanggung jawab (Agustina, Kurniati, 2009) .

2.2. Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Kriptografi adalah sebuah teknik penyandian pesan yang dilakukan agar pesan dapat dikirim dan diterima dengan aman. Kriptografi bertujuan untuk menjaga kerahasiaan data dan informasi agar tidak disalah gunakan oleh pihak yang tidak sah (Setyaningsih, 2015). Agar kriptografi dapat berjalan dengan baik haruslah terdapat empat elemen utama didalamnya, yang paling berkait satu sama lain (Ibrahim, 2017), yaitu:

1. Plainimage

Merupakan sebagai pesan gambar awal atau pesan asli yang dikirim pada proses komunikasi. *Plainimage* inilah yang kemudian dienkripsi dan didekripsi.

2. Cipherimage

Merupakan pesan gambar yang tersembunyi, yaitu pesan gambar asli (*plainimage*) yang telah dienkripsi pada proses kriptografi. *Cipherimage* ini dapat diubah kembali ke bentuk aslinya (*plainimage*) memanfaatkan *key* yang telah disediakan.

3. *Cryptography Key*

Merupakan kunci yang digunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (*key*) yang sama maka proses enkripsi dan deskripsi tidak dapat dilakukan dengan baik. Kunci (*key*) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya kriptografi.

4. *Encryption Decryption Algorithm*

Komponen terakhir yang juga sama pentingnya dalam proses kriptografi adalah algoritma yang di gunakan untuk enkripsi dan dekripsi.

2.2.1. Aspek Keamanan Kriptografi

Keamanan telah menjadi aspek yang sangat penting dari suatu sistem informasi yang umumnya hanya ditunjukkan bagi segolongan tertentu, karena itu penting untuk melindungi sistem informasi tersebut demi mencegahnya jatuh kepada pihak-pihak lain yang tidak berkepentingan. Salah satu upaya pengaman sistem informasi yang dapat dilakukan adalah kriptografi dengan beberapa aspek keamanan informasi (Agustina dan Kurniati, 2017), yaitu :

1. Kerahasiaan (*confidentiality*)

Merupakan layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.

2. Integritas Data (*integrity*)

Merupakan layanan yang menjamin bahwa pesan masih utuh/asli atau belum pernah dimanipulasi selama pengiriman.

3. Otentikasi (*authentication*)

Merupakan layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (*data origin authentication*).

4. Nir penyangkalan (*non repudiation*)

Merupakan layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

2.2.2. Serangan Terhadap Kriptografi

Setelah data dienkripsi bukan berarti tidak ada pihak yang ingin mengetahui dan memanfaatkan hasil *cipher* yang sudah terenkripsi, ilmu ini disebut dengan kriptanalisis. Adapun jenis serangan (*attack*) terhadap pesan yang sudah dienkripsi sebagai berikut (Prabowo, 2018) :

1. *Ciphertext only attack* merupakan penyerangan yang hanya mendapatkan pesanyang sudah disandikan.
2. *Known plaintext attack* merupakan serangan yang sudah mendapatkan sandi dan pesan asli.
3. *Chosen plaintext attack* adalah model serangan untuk pembacaan sandi yang menganggap bahwa penyerang dapat memperoleh *ciphertext* untuk *plaintext* sewenang-wenang. Tujuan dari serangan ini adalah untuk mendapatkan informasi yang mengurangi keamanan skema enkripsi.

2.2.3. Pengelompokan Algoritma Kriptografi

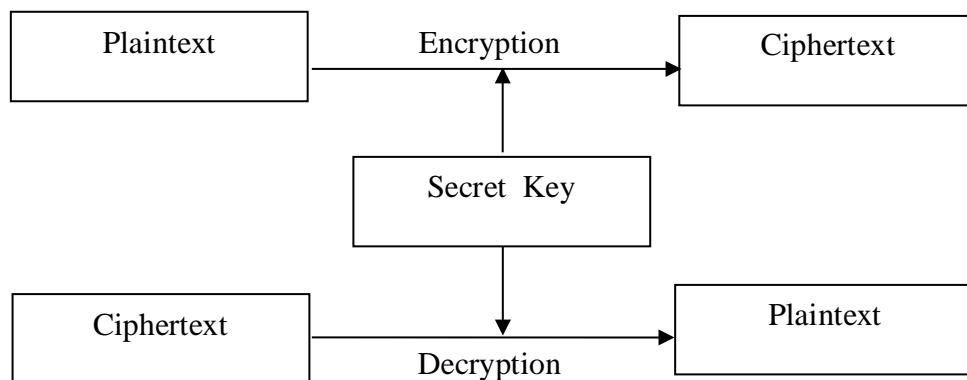
Berdasarkan jenis kunci yang digunakan dalam proses enkripsi dan dekripsi, algoritma kriptografi dibagi menjadi dua macam, yaitu kriptografi simetrik dan kriptografi asimetrik. Perbedaan utama di antara keduanya terletak pada sama dan tidaknya kunci yang digunakan dalam proses enkripsi dengan kunci yang digunakan pada proses dekripsi (Aleisa, 2015).

1. Algoritma Kriptografi Simetrik

Enkripsi kunci simetris atau tunggal adalah salah satu metode enkripsi tertua dan biasanya sesederhana menggeser huruf teks dengan nomor tertentu. Algoritma kunci simetris hanya menyediakan kunci tunggal untuk proses enkripsi dan dekripsi yang bisa menjadi angka, kata, atau huruf acak, dan siapa pun yang memiliki kunci akan memiliki kemampuan untuk dekripsi *ciphertext*. Kesulitan dalam proses ini adalah memberikan kunci dari pengirim kepada penerima dan memastikan bahwa penerima telah menerima kunci. Jika kuncinya hilang atau diperoleh oleh pihak ketiga, maka data yang dienkripsi menjadi tidak tersedia. Salah satu yang hebat manfaat dari sistem kunci rahasia adalah kemampuan untuk menerjemahkan kata sandi dengan mudah dengan kunci dan kecepatan enkripsi. Bila dibandingkan dengan sistem kunci asimetris, sistem kunci simetrik adalah

metode yang menarik karena aplikasinya tidak memerlukan keterlibatan pengguna eksternal (Aleisa, 2015).

Contoh dari algoritma ini adalah *Data Encryption Standard (DES)*, *Triple Data Encryption Standard (3DES)*, *International Data Encryption Algorithm (IDEA)*, *Advanced Encryption Standard (AES)*, dan sebagainya. Adapun diagram enkripsi kunci simetri dapat dilihat pada gambar 2.1 berikut :

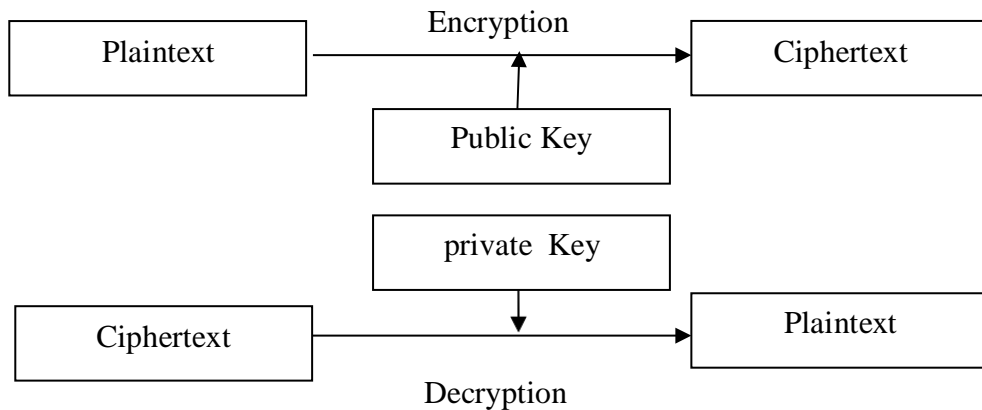


Gambar 2.1 Diagram enkripsi dan dekripsi kunci simetri
 Sumber : Thakur dan Khumar, 2011

2. Algoritma Kriptografi Asimetrik

Sistem algoritma kunci asimetris, juga dikenal sebagai enkripsi kunci publik, lebih aman karena perlu menggunakan dua kunci, kunci publik dan pribadi, untuk mengenkripsi dan mendekripsi teks. Kunci publik dapat diketahui oleh siapa saja, dan penerima akan memberikan kunci publiknya kepada pengirim dan pengirim akan menggunakannya untuk mengenkripsi teks yang akan dikirim. Penerima akan menerima *ciphertext* dan mendekripsi menggunakan kunci pribadinya. Kunci pribadi tidak pernah didistribusikan, itulah sebabnya ancaman dari pihak ketiga sangat berkurang karena tanpa kunci pribadi, teks tidak dapat didekripsi. Meskipun kunci publik dan pribadi dalam kriptografi asimetris telah memecahkan masalah keamanan pengiriman kunci, masih memiliki kelemahan di bidang keamanan. Pertama, enkripsi kunci publik jauh lebih lambat daripada enkripsi kunci rahasia tunggal. Ke dua, hanya efisien untuk sejumlah data kecil seperti email, tetapi tidak untuk enkripsi massal (Aleisa, 2015). Algoritma yang

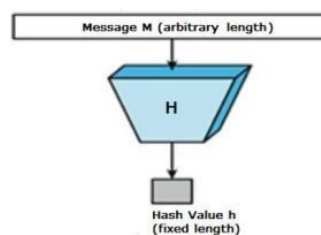
menggunakan kunci umum dan public ini antara lain *Digital Signature Algoritma* (DSA) , *Rivest Shamit-Adleman* (RSA), *Diffie- hellman* (DH), dan sebagainya. Proses enkripsi dekripsi kunci asimetri dapat dilihat pada gambar 2.2 berikut :



Gambar 2.2 Diagram enkripsi dan dekripsi kunci simetri
 Sumber : Thakur dan Khumar, 2011

3. Fungsi Hash

Fungsi *hash* adalah fungsi matematika yang mengubah nilai *input* numerik menjadi nilai numerik terkompresi lainnya. *Input* ke fungsi *hash* adalah panjang yang tidak ditentukan tetapi *output* selalu memiliki panjang tetap. Nilai yang dikembalikan oleh fungsi *hash* disebut pesan digest atau hanya nilai *hash* (Mohtashim, 2015). Gambar berikut ini menggambarkan fungsi *hash* :



Gambar 2.3 Fungsi Hash
 Sumber : Mohtashim, 2015

2.3. Algoritma Spritz

Algoritma *Spritz* adalah sebuah algoritma yang di perbaharui dari algoritma RC4 yang dilakukan oleh *Ron Rivest* dan *Jacob Schultz* pada 2014. Penamambahan elemen A yang relatif prima terhadap nilai N menjadi perbedaan antara algoritma *Spritz* dan RC4. Proses enkripsi algoritma *Spritz* adalah varian dari enkripsi

algoritma RC4, dimana data atau pesan yang dienkripsi akan diproses dengan konsep *stream cipher* yaitu enkripsi satu persatu. Selain *stream cipher*, algoritma *Spritz* juga bias digunakan sebagai fungsi *hash* dan *Code Authentication Message* (MAC) dengan menggunakan *spons* berfungsi dalam mengamankan data. Algoritma *Spritz* memiliki prosedur utama yang terdiri dari tiga proses (Zebua, 2018), yaitu :

1. *Key Scheduling Algorithm* (KSA)

Key Scheduling Algorithm atau algoritma penjadwalan kunci digunakan untuk membuat tabel S-Box (*array S*) dan permutasi tabel dalam *array S*. Panjang *array* yang diperlukan adalah 256 yang mulai dari indeks 0 hingga 255. Tujuandari KSA adalah memiliki proses *array* permutasi dengan nilai sebanyak 256 kali yang diinisialisasi dengan variabel *i* dan *j* dengan tipe integer.

Rumus KSA adalah :

untuk $i = 0$ sampai $N - 1$ $S[i] = i$

selanjutnya $i, j = 0$

untuk $i = 0$ sampai $N - 1$

$j = (j + S[i] + K[i \bmod \text{Key.length}]) \bmod N$ tukar ($S[i], S[j]$)

$j = j$ selanjutnya i

di mana N adalah ukuran *array* yang akan dimutasi, mis. 0 - 255.

2. *Pseudo-Random Generation Algoritma* (PRGA)

Pseudo-Random Generation Algoritma (PRGA) atau pengacakan dilakukan untuk mendapatkan kunci baru dengan sejumlah elemen - elemen polos. Nilai W adalah variabel baru yang ditambahkan ke algoritma *Spritz* sesuai dengan algoritma RC4. Nilai dari variabel i, j, k dan z dimulai dari 0 dan akan berubah sesuai hasil pada setiap iterasi. Proses ini melibatkan nilai-nilai *array S* yang telah diijinkan dalam proses KSA.

Rumus PRGA adalah:

untuk $i = 0$ sampai plain.length

$i = (i + w) \bmod N$

$j = (k + S[j + S[i]]) \bmod N$ $k = (i + k + S[j]) \bmod N$ tukar $S[i], S[j]$

$z = (S[j + S[i + S[z + k]]]) \bmod N$

output z selanjutnya i

di mana w adalah nilai integer yang relatif prima dengan N dan nilai i, j, k, z dimulai dari 0.

3. Enkripsi dan Dekripsi.

Proses enkripsi dan dekripsi dilakukan dengan cara XOR-biner setiap output z dengan masing-masing elemen polos dalam aliran.

Rumus enkripsi :

$$C_i = P_i \text{ XOR } z_i \dots\dots\dots(1)$$

Rumus dekripsi :

$$P_i = C_i \text{ XOR } z_i \dots\dots\dots(2)$$

Dimana :

P_i = elemen sederhana
 C_i = elemen sandi

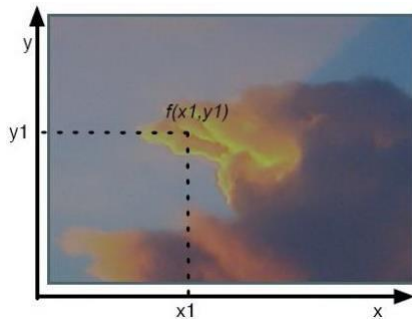
z_i = elemen kunci (hasil proses PRGA)

2.4. Citra Digital

Pengertian citra secara umum adalah suatu gambar, foto, ataupun berbagai tampilan dua dimensi yang menggambarkan suatu visualisasi objek. Citra juga dapat diartikan sebagai suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Berdasarkan bentuk sinyal penyusunnya, citra dapat digolongkan menjadi dua jenis yaitu citra analog dan citra digital. Citra analog adalah citra yang dibentuk dari sinyal analog yang bersifat kontinyu, sedangkan citra digital adalah citra yang dibentuk dari sinyal digital yang bersifat diskrit (Pamungkas, 2017).

2.4.1. Pengertian Citra Digital

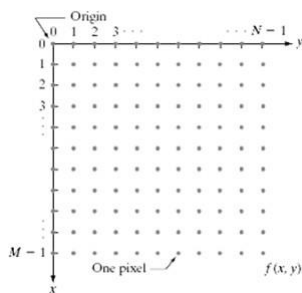
Citra digital adalah citra yang dapat diolah oleh komputer (Sutoyo, 2009). Citra digital juga dapat didefinisikan sebagai kumpulan piksel-piksel yang disusun dalam larik dua-dimensi, di mana piksel adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat (Ahmad, 2005). Citra digital diperoleh dengan cara mengukur warna pada sebuah citra pada titik-titik pada citra dan merpresentasikannya ke dalam bentuk digital atau angka bilangan bulat. Banyak teknik dalam mengolah citra, salah satunya adalah dengan mengidentifikasi tekstur pada objek citranya (Furqan, Sriani dan Harahap, 2020).



Gambar 2.4 Ilustrasi Citra digital

Sumber : Purnama, 2012

Citra digital memiliki fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut (Putra, 2010). Apabila nilai (x,y) dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital. Berikut gambar 2.5 yang menunjukkan posisi koordinat citra digital.



Gambar 2.5 Posisi Koordinat Citra Digital

Sumber : Putra, 2010

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi *geometric*), melakukan pemilihan citra ciri (*feature image*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Input dari pengolahan citra adalah citra, sedangkan outputnya adalah citra hasil

pengolahan (Sutoyo, 2009).

2.4.2. Format File Citra

Sebuah format file citra harus dapat menyatukan kualitas citra, ukuran file dan kompatibilitas dengan berbagai aplikasi. Format file citra standar yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan untuk menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Ini adalah contoh format umum yaitu, *bitmap (.bmp)*, *tagged image format (.tif, .tiff)*, *Portable Network Graphics (.png)*, *JPEG (.jpg)* (Putra, 2010).

Bahkan menurut Sutoyo *et al.* (2009), ada dua jenis format file citra yang sering digunakan dalam pengolahan citra, yaitu citra *bitmap* dan citra *vector*. Pada citra *bitmap* ini sering disebut juga citra *raster*. Citra *bitmap* ini menyimpan data kode citra secara digital dan lengkap (cara penyimpanannya adalah per piksel). Citra *bitmap* ini dipresentasikan dalam bentuk matriks atau dipetakan dengan menggunakan bilangan biner atau sistem bilangan yang lain. Citra ini memiliki kelebihan untuk memanipulasi warna, tetapi untuk mengubah objek lebih sulit. Tampilan *bitmap* mampu menunjukkan kehalusan gradasi bayangan dan warna dari sebuah gambar. Tetapi bila tampilan diperbesar maka tampilan di monitor akan tampak pecah-pecah (kualitas citra menurun). Contoh format file citra antara lain adalah *JPG, PNG, BMP, GIFF, TIF, WPG, IMG*.

2.5. Flowchart

Flowchart adalah bagian-bagian yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flowchart* merupakan suatu cara penyajian dari suatu Algoritma (Sutanta, 2005).

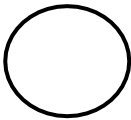
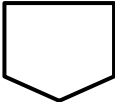
Flowchart disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses didalam program. Simbol-simbol yang digunakan dapat dibagi 3 (Tiga) kelompok (Sutanta, 2005), yakni sebagai berikut :

1. Simbol Penghubung / Alur (*Flow Direction Symbols*). Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.
2. Sistem Proses (*Procesing Symbols*). Simbol yang menunjukkan jenis operasi

pengolahan dalam suatu proses/prosedur.



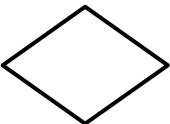



3. Simbol *input output (Input-Output)*. Simbol yang menunjukkan jenis peralatan yang digunakan sebagai *input* atau *output*.

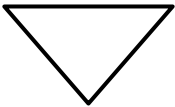

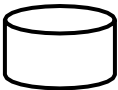
Tabel 2.1 Simbol Penghubung/Alur (*Flow Direction System*)

Simbol	Nama	Keterangan
		Simbol <i>Connector</i> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman/lembar yang sama.
	<i>Offline connector</i>	Simbol <i>Offline connector</i> Untuk menyatakan sambungan dari satu proses dalam halaman/lembar yang berbeda.

Sumber : Edhy Sutanta, 2005




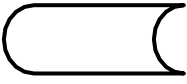

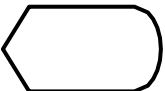
Tabel 2.2 Simbol Proses (*Processing Symbols*)

Simbol	Nama	Keterangan
	Proses	Simbol yang menunjukkan pengolahan yang dilakukan oleh computer.
	<i>Manual</i>	Simbol <i>Manual</i> Untuk menunjukan suatu tindakan (Proses) yang tidak
	<i>Decision/Logika</i>	Simbol <i>Decision/Logika</i> Untuk menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban Ya/Tidak
	<i>pendefinedProses</i>	Simbol <i>pendefined Proses</i> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
	<i>Terminal</i>	Simbol <i>Terminal</i> Untuk menyatakan permulaan dan akhirsuatu program.
	<i>Keying Operating</i>	Simbol <i>Keying Operating</i> Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard

	<i>Off-Line Storage</i>	Simbol <i>Off-Line Storage</i> Untuk menunjukkan bahwa data dalam simbol ini akan disimpan kesuatu media tertentu.
	<i>Manual Input</i>	Simbol <i>Manual Input</i> Untuk memasukkandata secara manual dengan menggunakan <i>online keyboard</i>
	<i>(Disk Storage)</i>	Data penyimpan (<i>Disk Storage</i>)

Sumber : Edhy Sutanta, 2005

Tabel 2.3 Simbol *Input-Output*

Simbol	Nama	Keterangan
	<i>Input-Output</i>	Simbol <i>Input-Output</i> Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya.
	<i>Punched Card</i>	Simbol <i>Punched Card</i> Untuk menyatakan input berasal dari kartu atau output ditulis ke kartu.
	<i>Magnetic-Tape Unit</i>	Simbol <i>Magnetic-Tape Unit</i> Untuk menyatakan input berasal dari pita magnetic atau output disimpan ke pita magnetic.
	<i>Disk Storage</i>	Simbol <i>Disk Storage</i> Untuk menyatakan input berasal dari disk atau output disimpan ke disk.
	<i>Dokument</i>	Simbol <i>Dokument</i> Untuk mencetak laporan ke Printer.
	<i>Display</i>	Simbol <i>Display</i> Untuk menyatakan peralatan output yang digunakan berupa layer (Video, Komputer).

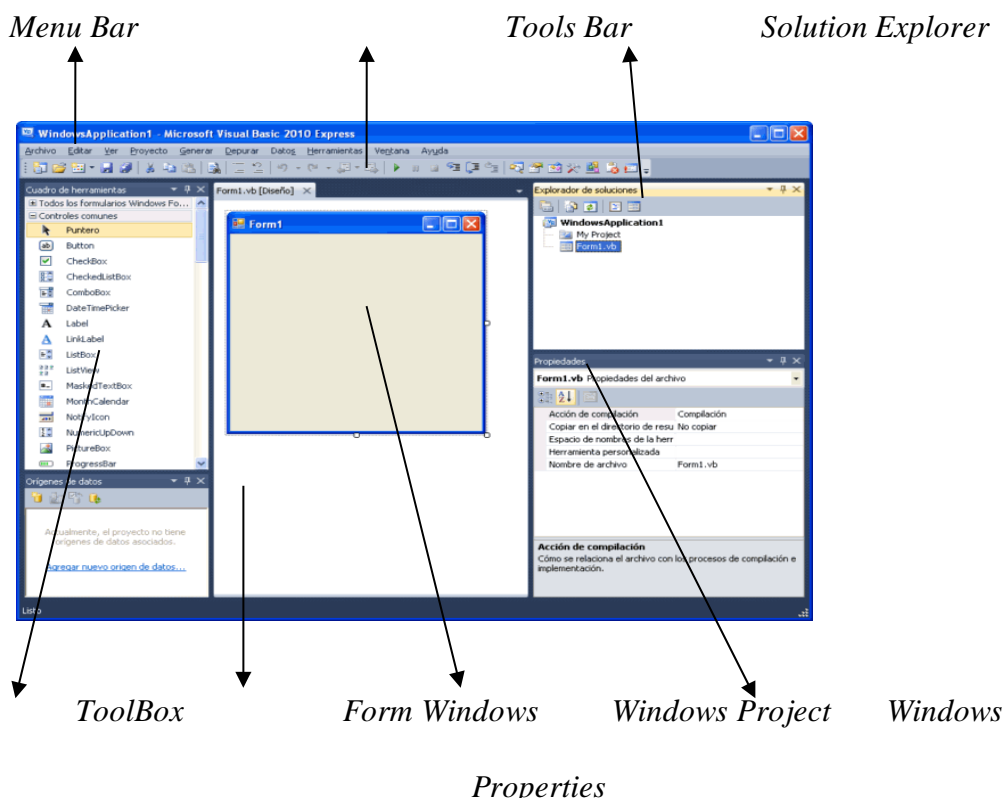
Sumber : Edhy Sutanta, 2005

2.6. Bahasa Pemrograman Visual Basic 2010

Microsoft meluncurkan *Visual Basic 2010* pada tahun 2010. *Visual Basic 2010* adalah bahasa pemrograman yang sepenuhnya berorientasi objek yang diimplementasikan pada *.NET Framework*. VB 2010 adalah versi bahasa

pemrograman *Visual Basic*.NET sudah menggunakan *Framework* berbasis .NET, sehingga anda dapat membuat *Console Application*, *Windows Application*, serta *Web Application* secara mudah dan terstruktur. Namun, vb2010 mempertahankan sebagian besar sintaks yang mirip dengan *Visual Basic* klasik 6. Serupa dengan versi bahasa VB.NET sebelumnya, *Visual Basic* dipaketkan dalam lingkungan pengembangan terpadu yang dikenal sebagai *Visual Studio* (Kiong, 2012).

Setelah mengaktifkan program *Visual Basic* 2010 dan membuka jenis aplikasi yang ada, maka bidang kerja tempat menghasilkan program aplikasi akan ditampilkan. Tempat ini disebut juga dengan *Integrated Development Integretion* (IDE). Bentuk layar *Visual Basic* 2010 hampir sama dengan layar program-program aplikasi windows pada umumnya, seperti *Microsoft C++*, *Microsoft Visual Basic*, *Visual Basic* 6.0, FoxPro dan lainnya (Kiong, 2012). Berikut tampilan dasar pada *Visual Basic* 2010, seperti yang terlihat pada gambar 2.6 di bawah ini.



Properties
Gambar 2.6 Tampilan IDE *Visual Basic* 2010
Sumber : L.V Kiong, 2012

Berikut beberapa bagian-bagian dan nama *windows* yang dapat tampil pada IDE *Visual Basic* 2010, antara lain :

1. *Menu Bar*

Menu Bar digunakan untuk melakukan proses atau perintah- perintah tertentu. *Menu bar* dibagi menjadi beberapa pilihan sesuai dengan kegunaannya, seperti *menu bar* file digunakan untuk memproses atau menjalankan perintah-perintah yang berhubungan dengan file, seperti membuka file baru, menyimpan file, selain itu juga terdapat *menu bar* lain seperti *edit*, *view*, *project*, *build*, *debug*, *data*, *format*, *tools*, *window*, dan *help*. Untuk menggunakan *menu bar*, dapat dilakukan dengan dua cara, yaitu dengan *mouse*, klik *mouse* pada *menu* dan *sub menu*, dan juga dengan *keyboard*, tekan ALT dan karakter bergaris bawah.

Contoh : ALT + E untuk menampilkan menu *edit*.

2. *Toolbars*

Toolbars merupakan sebuah komponen yang berfungsi untuk melakukan perintah khusus secara cepat.

3. *Toolbox*

Toolbox merupakan komponen lingkungan kerja VB yang berisikan *tool-tool* untuk di tempatkan di *form*. Jika kita membuat sebuah aplikasi, maka komponen-komponen tersebut akan kita tempatkan di form dan menjadi komponen jendela program.

4. *Solution Explorer*

Solution Explorer berfungsi untuk menampilkan *project* beserta file-file pendukung yang terdapat pada sebuah program aplikasi.

5. *Form Windows*

Form merupakan objek utama yang berfungsi sebagai *interface* (antar muka) dari aplikasi yang akan dibuat. Pada *Form* ini kita bisa meletakkan objek -objek yang terdapat dari *Toolbox*.

6. *Windows Project*

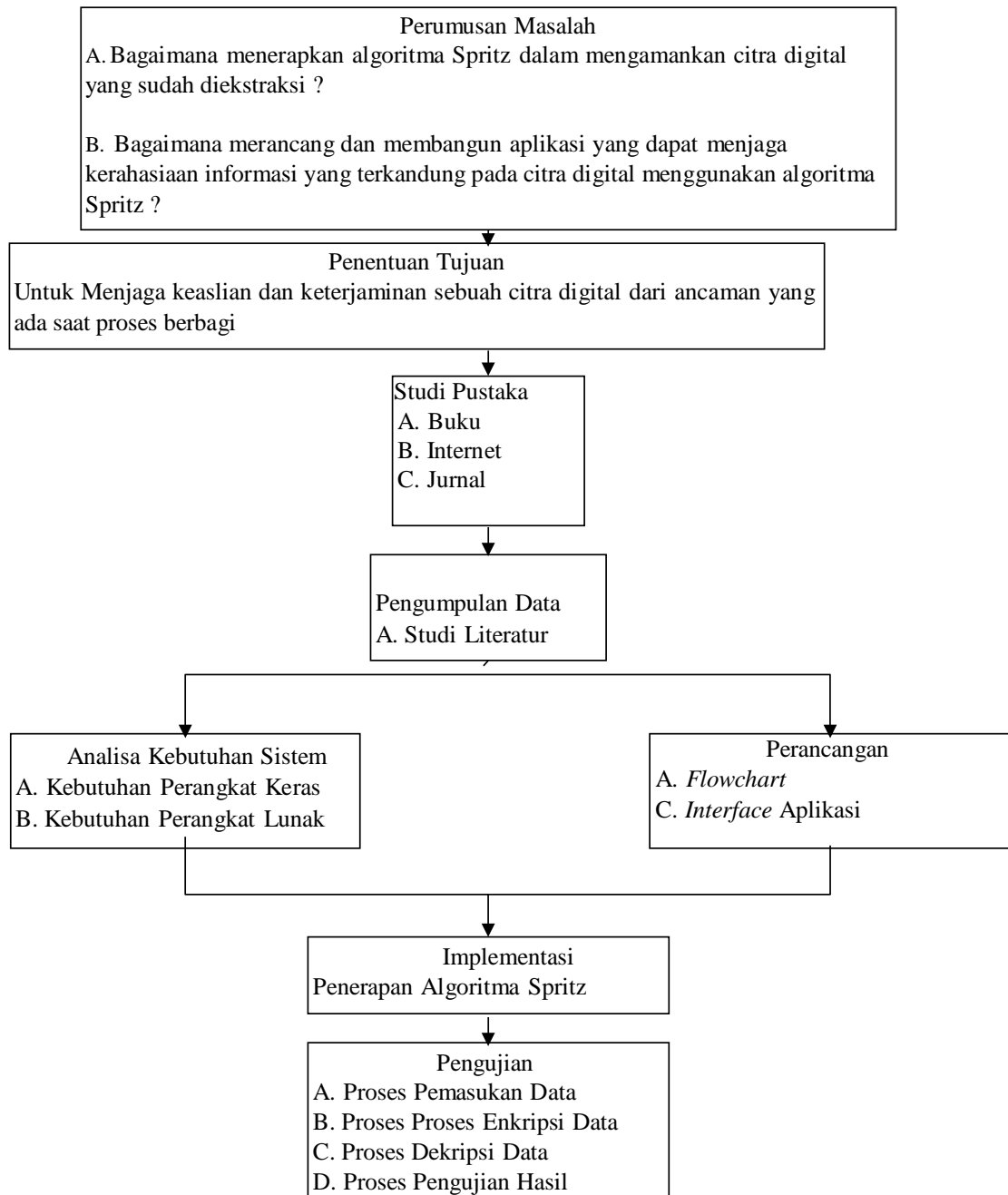
Windows Project berfungsi untuk menampilkan daftar *form* dan modul yang terdapat di *project* aplikasi yang sedang dikerjakan.

7. *Windows Properties*

Windows Properties berfungsi untuk menampilkan daftar dari sebuah komponen yang sedang aktif.

BAB III METODE PENELITIAN

3.1. Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian

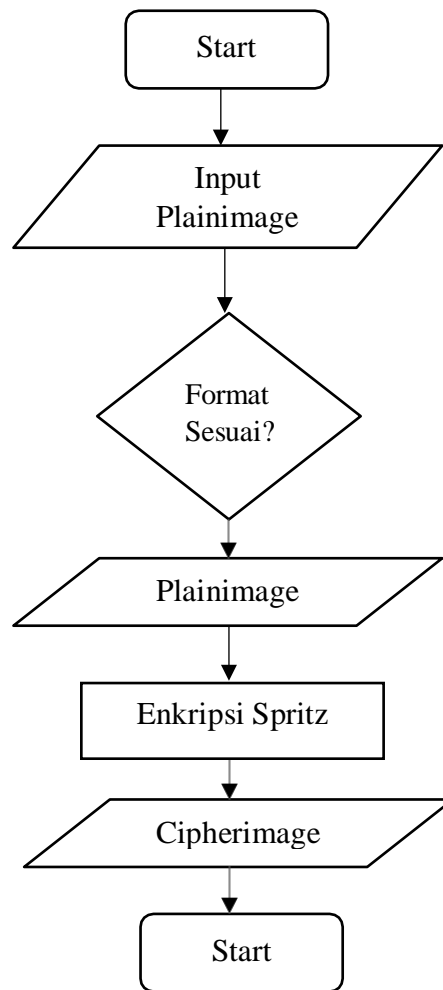
3.2. Metode Pengumpulan Data

Pengumpulan data menggunakan metode studi literatur yang merupakan tahap pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper*, dan buku-buku yang berkaitan dengan judul penelitian, serta mencari informasi dari berbagai sumber di internet untuk mengetahui perkembangan terbaru dari data yang diambil sebagai bahan dalam pembuatan tugas akhir.

3.3. Analisa Sistem

Citra digital sangat rentan terhadap penyadapan maupun pencurian oleh pihak yang tidak bertanggung jawab untuk keuntungan kelompok atau pribadi. Apalagi di masa saat ini, di mana citra digital dapat didistribusikan bebas melalui jaringan internet berbasis *chatting*. Citra digital bersifat rahasia yang belum disandikan jika berada ditangan yang salah, maka dengan sangat mudah dianalisa karena citra tersebut masih berupa data yang asli sehingga dapat merugikan salah satu pihak pengirim. Hal tersebut dapat diatasi dengan teknik kriptografi.

Berdasarkan rumusan masalah pada bab sebelumnya dan paparan di atas, masalah yang terjadi adalah bagaimana sebuah citra digital yang belum disandikan dapat diamankan dengan teknik kriptografi. Sebelum melakukan teknik enkripsi kriptografi, terlebih dahulu menentukan pola citra yang akan dienkripsi. Pola citra yang akan dienkripsi pada pembahasan skripsi ini adalah sebuah citra warna RGB dengan format yang sudah ditentukan yaitu JPG, PNG dan BMP. Adapun proses *input* dan pengamanan citra dengan teknik kriptografi dapat dilihat pada gambar berikut:



Gambar 3.2 *Flowchart* Proses Pengamanan Citra

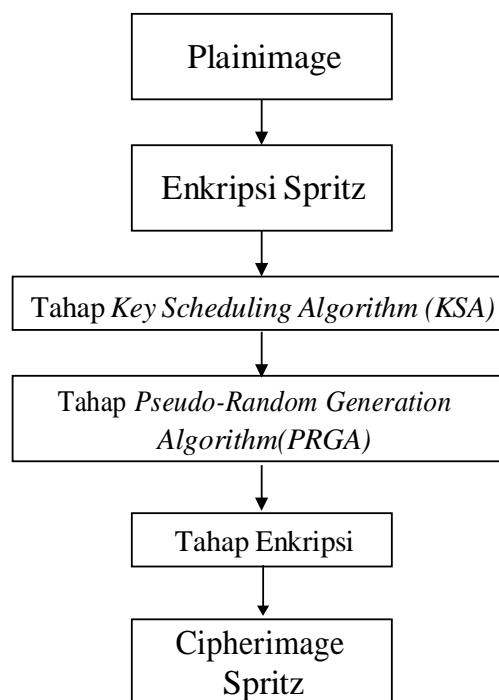
Agar algoritma dapat berjalan dengan baik terhadap enkripsi citra digital, terlebih dahulu citra diekstraksi untuk mendapatkan informasi nilai desimal pada setiap *pixel* citra. Citra yang akan diekstraksi pada kasus ini adalah citra RGB yang memiliki *bitdepth* 24 bit. Maksud dari 24 bit adalah di dalam 1 *pixel* citra digital memiliki nilai warna RGB dengan ruang sebanyak 24 bit. R (*Red*) memiliki nilai 8bit, G (*Green*) memiliki nilai 8 bit dan B (*Blue*) memiliki nilai 8 bit. Nilai RGB disetiap *pixel* inilah yang akan dienkripsi menggunakan algoritma kriptografi. Proses enkripsi algoritma kriptografi yang akan diterapkan adalah algoritma *Spritz*. Proses awal enkripsi akan mengacak tabel *Key Scheduling Algorithm* (KSA) sebanyak 256 kali dengan nilai desimal kunci menggunakan rumus *Key Scheduling Algorithm* (KSA), lalu nilai dari tabel yang sudah di acak tersebut digunakan pada

tahap *Pseudo-Random Generation Algorithm* (PRGA) untuk mendapatkan nilai z akhir yang akan dirubah kedalam bentuk biner, kemudian dilakukan operasi XOR dengan *plainimage* yang sudah dirubah kedalam bentuk biner untuk tahap enkripsi, sehingga akan menghasilkan *cipherimage*.

Berikut gambar proses enkripsi dan dekripsi algoritma *Spritz* :

1. Diagram enkripsi Algoritma *Spritz*

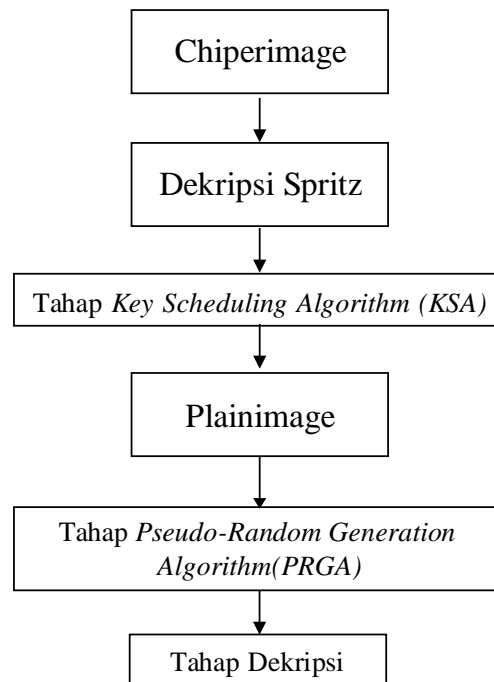
Diagram enkripsi algoritma *Spritz* pada *plainimage* sebagai berikut:



Gambar 3.3 Diagram Enkripsi *Spritz*

2. Diagram Dekripsi Algoritma *Spritz*

Proses dekripsi adalah sama dengan proses enkripsi. Diagram dekripsi algoritma *Spritz* pada *cipherimage* sebagai berikut.



Gambar 3.4 Diagram Dekripsi *Spritz*

3.3.1. Analisa Kebutuhan Sistem

Analisa Kebutuhan sistem berfungsi untuk menentukan perangkat apa saja yang dibutuhkan dalam pembuatan aplikasi keamanan citra digital yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*). Dengan menggunakan analisa kebutuhan sistem maka dapat diketahui kebutuhan minimum yang diperlukan untuk membuat aplikasi keamanan citra digital tersebut. Berikut ini adalah penjabaran tentang spesifikasi *hardware* dan *software* yang dibutuhkan dalam pembuatan aplikasi keamanan citra digital.

1. Kebutuhan Perangkat Keras (*Hardware*)

kebutuhan perangkat keras (*hardware*) yang digunakan untuk mendukung Aplikasi Keamanan Citra Digital adalah sebagai berikut :

Tabel 3.1 Kebutuhan *Hardware*

Nama Komponen	Spesifikasi
<i>Procesor</i>	Intel(R) Core(TM) i3-380M
<i>Memory</i>	4GB
<i>Harddisk</i>	100 GB

3.3.2. Kebutuhan Perangkat Lunak (*Software*)

Adapun kebutuhan perangkat lunak (*software*) penulis yang digunakan untuk perancangan Aplikasi Keamanan Citra Digital adalah sebagai berikut :

- a. *Visual Studio* 2010 Ultimate Edition
- b. *Microsoft Visio* 2010

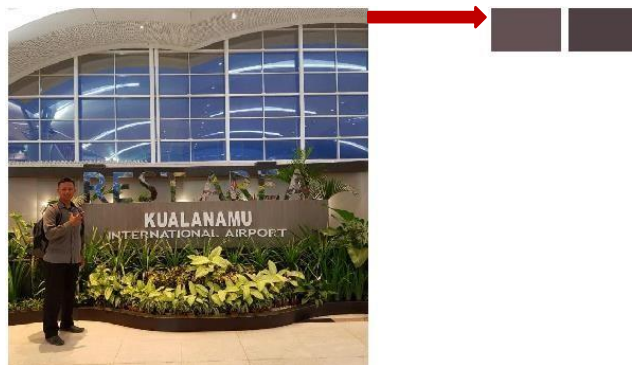
3.4. Penerapan

Penerapan/penggunaan sistem ini adalah menjelaskan sistem yang dibangun dengan menggunakan teknik kriptografi menggunakan algoritma *Spritz* untuk mengamankan citra digital berbasis dekstop.

BAB IV HASIL DAN PEMBAHASAN

4.1. Proses Enkripsi Manual Algoritma Spritz

Contoh kasus proses enkripsi manual menggunakan algoritma *Spritz* dalam menyandikan sebuah citra digital berwarna dengan ekstensi jpg resolusi 640 x 640 dan *bitdepth* 24 *bit*.



Gambar 4.1 *Plainimage* Sampel 2 x 1 *Pixel*

Pada proses hitungan manual, nilai desimal elemen warna pada setiap *pixel* diambil menggunakan *software* matlab. Adapun nilai dari setiap *pixel* sampel yang diambil dilihat pada tabel di bawah ini :

Tabel 4.1 Nilai RGB *Pixel* Sampel

Pixel	Warna	Plain	Pixel	Warna	Plain
		Desimal			Desimal
1	R	73	1	R	65
	G	81		G	76
	B	66		B	82

Berdasarkan tabel di atas, nilai desimal dari *plainimage* 2 x 1 adalah 73, 81, 66, 65, 76, 82. Kunci yang digunakan dalam enkripsi hitungan manual adalah “2019” dan panjang kunci 4 *byte*. Tahap awal proses enkripsi adalah pengacakan tabel *Key Scheduling Algorithm* (KSA) sebanyak 256 kali dengan nilai desimal kunci menggunakan rumus *Key Scheduling Algorithm* (KSA), dilanjutkan dengan tahap *Pseudo-Random Generation Algorithm* (PGRA) untuk mendapatkan nilai *z* akhir

yang akan di XOR dengan nilai dari RGB *plainimage* pada tahap enkripsi untuk mendapatkan *cipherimage*.

1. Tahap *Key Scheduling Algorithm (KSA)*

Langkah *key scheduling* dimulai dengan menginisialisasikan *state* awal berupa larik yang terdiri dari 256 elemen. Larik *state* awal dapat dilihat pada tabel 4.2 berikut ini:

Tabel 4.2 Larik *State* Awal

0	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36	37	38	39	40	41
42	43	44	45	46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93	94	95	96	97
98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139
140	141	142	143	144	145	146	147	148	149	150	151	152	153
154	155	156	157	158	159	160	161	162	163	164	165	166	167
168	169	170	171	172	173	174	175	176	177	178	179	180	181
182	183	184	185	186	187	188	189	190	191	192	193	194	195
196	197	198	199	200	201	202	203	204	205	206	207	208	209
210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237
238	239	240	241	242	243	244	245	246	247	248	249	250	251
252	253	254	255										

Selanjutnya akan dilakukan perhitungan untuk nilai j . Untuk melakukan perhitungan nilai j maka kunci yang digunakan harus diubah ke dalam kode ASCII. Kode ASCII dapat dilihat pada gambar 4.3 berikut ini :

0	A	44	.	88	X	132	E	176	°	220	U
1	ā	45	-	89	Y	133	ē	177	±	221	Y
2	Ā	46	.	90	Z	134	Ē	178	ˆ	222	Þ
3	ā	47	/	91	[135	ē	179	ˆ	223	ß
4	À	48	0	92	\	136	È	180	ˆ	224	à
5	à	49	1	93]	137	é	181	μ	225	á
6	Ç	50	2	94	^	138	É	182	¶	226	â
7	ç	51	3	95	~	139	ê	183	ˆ	227	ã
8	Ĉ	52	4	96	”	140	Ĕ	184	ˆ	228	ä
9	ĉ	53	5	97	a	141	ē	185	ˆ	229	å
10	Ċ	54	6	98	b	142	Ĝ	186	°	230	æ
11	ċ	55	7	99	c	143	ġ	187	»	231	ç
12	Ď	56	8	100	d	144	Ġ	188	¼	232	È
13	đ	57	9	101	e	145	ġ	189	½	233	E
14	İ	58	:	102	f	146	Ģ	190	¾	234	É
15	ı	59	;	103	g	147	ĵ	191	ˆ	235	Ê
16	İ	60	<	104	h	148	Ģ	192	À	236	I
17	ij	61	=	105	i	149	ĵ	193	À	237	I
18	J	62	>	106	j	150	H	194	À	238	I
19	K	63	?	107	k	151	h	195	À	239	I
20	k	64	@	108	l	152	H	196	À	240	Đ
21	κ	65	A	109	m	153	h	197	À	241	N
22	L	66	B	110	n	154	I	198	Æ	242	O
23	l	67	C	111	o	155	ı	199	Ç	243	O
24	L	68	D	112	p	156	I	200	E	244	O
25	ı	69	E	113	q	157	ı	201	E	245	Ó
26	L	70	F	114	r	158	I	202	E	246	Ó
27	ı	71	G	115	s	159	ı	203	È	247	+
28	L	72	H	116	t	160	I	204	I	248	Ø
29	ı	73	I	117	u	161	ı	205	I	249	U
30	L	74	J	118	v	162	e	206	I	250	U
31	ı	75	K	119	w	163	£	207	I	251	U
32	SPACE	76	L	120	x	164	¤	208	Đ	252	Ū
33	!	77	M	121	y	165	¥	209	N	253	Y
34	"	78	N	122	z	166	ı	210	O	254	Þ
35	#	79	O	123	{	167	š	211	O	255	Y
36	\$	80	P	124		168	-	212	O		
37	%	81	Q	125	}	169	©	213	Ó		
38	&	82	R	126	~	170	ª	214	Ó		
39	'	83	S	127	Đ	171	«	215	×		
40	(84	T	128	Ď	172	-	216	Ø		
41)	85	U	129	đ	173		217	Ū		
42	*	86	V	130	Đ	174	©	218	Ū		
43	+	87	W	131	đ	175	-	219	Ū		

Gambar 4.3 Kode ASCII

Tentukan nilai desimal kunci “2019” berdasarkan tabel ASCII. Nilai desimal kunci yaitu :

Index	0	1	2	3
Kunci	2	0	1	9
Desimal	50	48	49	57

Kemudian dilakukan perhitungan nilai j yang pertama dengan nilai awal $i = 0$ dan $j = 0$ sebagai berikut :

$$j = (j + S[i] + \text{key}[i \bmod \text{keylength}]) \bmod 256$$

$$j = (0 + S[0] + \text{key}[0 \bmod 4]) \bmod 256$$

$$j = (0 + 0 + 50) \bmod 256$$

$$j = 50 \bmod 256$$

$$j = 50$$

Nilai $S[i] = S[0]$ swap dengan nilai $S[j] = S[50]$ maka, Nilai $S[0] = 50$ dan nilai $S[50] = 0$

Selanjutnya perhitungan kembali untuk nilai j yang kedua dengan nilai $i = 1$ dan

$j = 50$ sebagai berikut :

$$j = (j + S[i] + \text{key}[i \bmod \text{keylength}]) \bmod 256$$

$$j = (50 + S[1] + \text{key}[1 \bmod 4]) \bmod 256$$

$$j = (50 + 1 + 48) \bmod 256$$

$$j = 99 \bmod 256$$

$$j = 99$$

Nilai $S[i] = S[1]$ swap dengan nilai $S[j] = S[99]$ maka, Nilai $S[1] = 99$ dan nilai $S[99] = 1$

Selanjutnya perhitungan kembali untuk nilai j yang ketiga dengan nilai $i = 2$ dan

$j = 99$ sebagai berikut :

$$j = (j + S[i] + \text{key}[i \bmod \text{keylength}]) \bmod 256$$

$$j = (99 + S[2] + \text{key}[2 \bmod 4]) \bmod 256$$

$$j = (99 + 2 + 49) \bmod 256$$

$$j = 150 \bmod 256$$

$$j = 150$$

Nilai $S[i] = S[2]$ swap dengan nilai $S[j] = S[150]$ maka, Nilai $S[2] = 150$ dan nilai $S[150] = 2$

Selanjutnya perhitungan kembali untuk nilai j yang keempat dengan nilai $i = 3$ dan

$j = 150$ sebagai berikut :

$$j = (j + S[i] + \text{key}[i \bmod \text{keylength}]) \bmod 256$$

$$j = (150 + S[3] + \text{key}[3 \bmod 4]) \bmod 256$$

$$j = (150 + 3 + 57) \bmod 256$$

$$j = 210 \bmod 256$$

$$j = 210$$

Nilai $S[i] = S[3]$ swap dengan nilai $S[j] = S[210]$ maka, Nilai $S[3] = 210$ dan nilai $S[210] = 3$

Langkah ini diulangi hingga i mencapai nilai 255 dan hasil dari tahap *key scheduling* dapat dilihat pada tabel 4.3 dimana nilai i berada pada baris yang berwarna biru.

Tabel 4.4 Hasil *Key Scheduling Algorithm(KSA)*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
207	234	45	53	150	195	74	173	124	140	64	228	2	54	206
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
141	15	252	108	40	75	136	77	152	209	244	71	0	34	144
30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
29	50	23	81	33	73	21	3	222	158	86	32	96	214	37
45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
47	58	13	175	51	99	210	35	133	80	103	49	247	30	249
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74
200	4	193	60	248	83	65	139	135	9	106	238	229	90	218
75	76	77	78	79	80	81	82	83	84	85	86	87	88	89
227	187	121	250	110	8	230	27	44	92	220	165	118	36	104
90	91	92	93	94	95	96	97	98	99	100	101	102	103	104
237	142	107	183	84	146	85	102	203	69	73	5	16	159	24
105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
128	168	41	39	52	17	20	91	59	109	82	129	26	205	208
120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
156	87	67	241	113	31	221	72	122	198	115	42	6	70	240
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
127	10	160	232	239	57	38	197	138	174	176	68	117	181	14
150	151	152	153	154	155	156	157	158	159	160	161	162	163	164
255	89	134	148	18	125	88	130	151	12	171	213	112	145	154
165	166	167	168	169	170	171	172	173	174	175	176	177	178	179
66	147	28	48	224	201	98	120	226	114	202	55	166	25	243
180	181	182	183	184	185	186	187	188	189	190	191	192	193	194
137	199	95	79	62	154	178	169	185	217	223	98	153	167	235
195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
177	7	253	225	162	155	22	192	157	180	97	179	126	136	211
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
186	191	204	170	245	251	246	11	100	119	231	76	63	123	190
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
164	184	212	182	163	172	43	94	105	161	242	219	1	19	215
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254
111	194	189	233	143	61	101	149	116	216	196	132	131	188	56
255														
46														

2. Tahap *Pseudo-Random Generation Algorithm(PRGA)*

Setelah melakukan tahap *Key Scheduling Algorithm(KSA)* maka akan dilakukan proses *Pseudo-Random Generation Algorithm(PRGA)* sebanyak panjang nilai *pixel plainimage*. Adapun panjang nilai *pixel* sampel *plainimage* sebagai berikut:

Tabel 4.5 Nilai RGB *Pixel* Sampel

Pixel	Warna	Plain
		Desimal
1	R	73
	G	81
	B	66
2	R	65
	G	76
	B	82

Pada tahap awal inisialisasikan nilai $i = 0$, $j = 0$, $k = 0$, $z = 0$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

$$\begin{aligned}
 i &= i + w &&= 0 + 221 \bmod 256 = 221 \\
 j &= k + S[j + S[i]] &&= 0 + S[0 + S[221]] \bmod 256 \\
 &&&= 0 + S[0 + 76] \bmod 256 \\
 &&&= S[76] = 187 \\
 k &= i + k + S[j] &&= 221 + 0 + S[187] \bmod 256 \\
 &&&= 221 + 0 + 169 \bmod 256 \\
 &&&= 134 \\
 S[i], S[j] &&&= S[i] = S[221] = 76, S[j] = S[187] = 169 \\
 \text{Swap } S[i] \text{ with } S[j] &&&= S[i] = S[221] = 169, S[j] = S[187] = 76 \\
 z &= S[j + S[i + S[z + k]]] &&= S[187 + S[221 + S[0 + 134]]] \bmod 256 \\
 &&&= S[187 + S[221 + 240]] \bmod 256 \\
 &&&= S[187 + S[205]] \bmod 256 \\
 &&&= S[187 + 97] \bmod 256 \\
 &&&= S[28] \bmod 256 \\
 &&&z = 34
 \end{aligned}$$

Tahap kedua inisialisasikan nilai $i = 221$, $j = 187$, $k = 134$, $z = 34$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

$$i = i + w = 221 + 221 \bmod 256 = 186$$

$$\begin{aligned}
 j &= k + S[j + S[i]] &= 134 + S[187 + S[186]] \bmod 256 \\
 & &= 134 + S[187 + 178] \bmod 256 \\
 & &= 134 + S[109] \bmod 256 \\
 & &= 134 + 52 \\
 & &= 186 \\
 k &= i + k + S[j] &= 186 + 134 + S[186] \bmod 256 \\
 & &= 64 + 178 \bmod 256 \\
 & &= 242 \\
 S[i], S[j] & &= S[i] = S[186] = 178, S[j] = S[186] = 178 \\
 \text{Swap } S[i] \text{ with } S[j] & &= S[i] = S[186] = 178, S[j] = S[186] = 178 \\
 z &= S[j + S[i + S[z + k]]] &= S[186 + S[186 + S[34 + 242]]] \bmod 256 \\
 & &= S[186 + S[186 + S[20]]] \bmod 256 \\
 & &= S[186 + S[186 + 75]] \bmod 256 \\
 & &= S[186 + S[5]] \bmod 256 \\
 & &= S[186 + 195] \bmod 256 \\
 & &= S[125] \bmod 256
 \end{aligned}$$

$$z = 31$$

Tahap ketiga inisialisasikan nilai $i = 186$, $j = 186$, $k = 242$, $z = 31$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

$$\begin{aligned}
 i &= i + w &= 186 + 221 \bmod 256 = 151 \\
 j &= k + S[j + S[i]] &= 242 + S[186 + S[152]] \bmod 256 \\
 & &= 242 + S[186 + 89] \bmod 256 \\
 & &= 242 + S[19] \bmod 256 \\
 & &= 242 + 40 \bmod 256 \\
 & &= 26 \\
 k &= i + k + S[j] &= 151 + 242 + S[26] \bmod 256 \\
 & &= 137 + 71 \bmod 256 \\
 & &= 208 \\
 S[i], S[j] & &= S[i] = S[151] = 89, S[j] = S[26] = 71 \\
 \text{Swap } S[i] \text{ with } S[j] & &= S[i] = S[151] = 71, S[j] = S[26] = 89
 \end{aligned}$$

$$\begin{aligned}
z &= S[j + S[i + S[z + k]]] &&= S[26 + S[151 + S[31 + 208]]] \bmod 256 \\
&&&= S[26 + S[151 + S[239]]] \bmod 256 \\
&&&= S[26 + S[151 + 215]] \bmod 256 \\
&&&= S[26 + S[110]] \bmod 256 \\
&&&= S[26 + 17] \bmod 256 \\
&&&= S[43] \bmod 256
\end{aligned}$$

$$z = 214$$

Tahap keempat inialisasikan nilai $i = 151$, $j = 26$, $k = 208$, $z = 214$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

$$\begin{aligned}
i &= i + w &&= 151 + 221 \bmod 256 = 116 \\
j &= k + S[j + S[i]] &&= 208 + S[26 + S[116]] \bmod 256 \\
&&&= 208 + S[26 + 129] \bmod 256 \\
&&&= 208 + S[155] \bmod 256 \\
&&&= 208 + 125 \bmod 256 \\
&&&= 107
\end{aligned}$$

$$\begin{aligned}
k &= i + k + S[j] &&= 116 + 208 + S[207] \bmod 256 \\
&&&= 68 + 41 \bmod 256 \\
&&&= 109
\end{aligned}$$

$$S[i], S[j] = S[i] = S[116] = 129, S[j] = S[107] = 41$$

$$\text{Swap } S[i] \text{ with } S[j] = S[i] = S[116] = 41, S[j] = S[107] = 129$$

$$\begin{aligned}
z &= S[j + S[i + S[z + k]]] &&= S[107 + S[116 + S[214 + 109]]] \bmod 256 \\
&&&= S[107 + S[116 + S[67]]] \bmod 256 \\
&&&= S[107 + S[116 + 139]] \bmod 256 \\
&&&= S[107 + S[255]] \bmod 256 \\
&&&= S[107 + 46] \bmod 256 \\
&&&= S[153] \bmod 256
\end{aligned}$$

$$z = 148$$

Tahap ke lima inialisasikan nilai $i = 116$, $j = 107$, $k = 109$, $z = 148$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

$$i = i + w = 116 + 221 \bmod 256 = 81$$

$$\begin{aligned}
j &= k + S[j + S[i]] &&= 109 + S[107 + S[81]] \bmod 256 \\
&&&= 109 + S[107 + 230] \bmod 256 \\
&&&= 109 + S[81] \bmod 256 \\
&&&= 109 + 81 \bmod 256 \\
z &= 190 \\
k &= i + k + S[j] &&= 81 + 109 + S[190] \bmod 256 \\
&&&= 81 + 109 + 223 \bmod 256 \\
&&&= 157 \\
S[i], S[j] &&&= S[i] = S[81] = 230, S[j] = S[190] = 223 \\
\text{Swap } S[i] \text{ with } S[j] &&&= S[i] = S[81] = 223, S[j] = S[190] = 230 \\
z &= S[j + S[i + S[z + k]]] &&= S[190 + S[81 + S[148 + 157]]] \bmod 256 \\
&&&= S[190 + S[81 + S[49]]] \bmod 256 \\
&&&= S[190 + S[81 + 51]] \bmod 256 \\
&&&= S[190 + S[132]] \bmod 256 \\
&&&= S[190 + 6] \bmod 256 \\
&&&= S[196] \bmod 256
\end{aligned}$$

$$z = 7$$

Tahap ke enam inisialisasikan nilai $i = 81$, $j = 190$, $k = 157$, $z = 7$, dan w adalah bilangan acak yang merupakan bilangan GCD atau relatif prima dengan panjang S yaitu 256. Selanjutnya lakukan proses seperti berikut:

$$\begin{aligned}
i &= i + w &&= 81 + 221 \bmod 256 = 46 \\
j &= k + S[j + S[i]] &&= 157 + S[190 + S[46]] \bmod 256 \\
&&&= 157 + S[190 + 58] \bmod 256 \\
&&&= 109 + S[248] \bmod 256 \\
&&&= 109 + 248 \bmod 256 \\
z &= 101 \\
k &= i + k + S[j] &&= 46 + 57 + S[101] \bmod 256 \\
&&&= 46 + 57 + 5 \bmod 256 \\
&&&= 118 \\
S[i], S[j] &&&= S[i] = S[46] = 58, S[j] = S[101] = 5 \\
\text{Swap } S[i] \text{ with } S[j] &&&= S[i] = S[46] = 5, S[j] = S[101] = 58
\end{aligned}$$

$$\begin{aligned}
z &= S[j + S[i + S[z + k]]] &&= S[101 + S[46 + S[7 + 118]]] \text{ mod } 256 \\
&&&= S[101 + S[46 + S[125]]] \text{ mod } 256 \\
&&&= S[101 + S[46 + 31]] \text{ mod } 256 \\
&&&= S[101 + S[77]] \text{ mod } 256 \\
&&&= S[101 + 121] \text{ mod } 256 \\
&&&= S[222] \text{ mod } 256 \\
z &= 63
\end{aligned}$$

3. Tahap Enkripsi

Pada tahap ini setiap nilai desimal RGB dari *pixel* yang akan dienkripsi dirubah menjadi bilangan biner, lalu akan dilakukan operasi XOR dengan nilai *output z* dari perhitungan setiap tahap *Pseudo-Random Generation Algorithm (PRGA)*. Nilai *output z* dari perhitungan setiap tahap *Pseudo-Random Generation Algorithm (PRGA)* akan dirubah menjadi bilangan biner. Proses enkripsi manual setiap nilai desimal RGB *pixel*, yaitu :

Plainimage	Pixel 1			Pixel 2		
	R	G	B	R	G	B
Desimal	73	81	66	65	76	82
Biner	01001001	01010001	01000010	01000001	01001100	01010010

Output z	34	31	214	148	7	63
Biner	00100010	00011111	11010110	10010100	00000111	00111111

Nilai warna "R" *Pixel 1*

Nilai warna "R" pada *pixel 1* bernilai desimal 73 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 34.

$$73 = 01001001$$

$$34 = \underline{00100010} \oplus$$

$$01101011 = 107$$

Nilai warna "G" *Pixel 1*

Nilai warna "G" pada *pixel 1* bernilai desimal 81 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 31. $81 = 01010001$

$$31 = \underline{00011111} \oplus$$

$$01001110 = 78$$

Nilai warna “B” *Pixel 1*

Nilai warna “B” pada *pixel 1* bernilai desimal 66 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu $214.66 = 01000010$

$$214 = \underline{11010110} \oplus$$

$$10010100 = 148$$

Nilai warna “R” *Pixel 2*

Nilai warna “R” pada *pixel 2* bernilai desimal 65 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 148.

$$65 = 01000001$$

$$148 = \underline{10010100} \oplus$$

$$11010101 = 213$$

Nilai warna “G” *Pixel 2*

Nilai warna “G” pada *pixel 2* bernilai desimal 76 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu $7.76 = 01001100$

$$7 = \underline{00000111} \oplus$$

$$01001011 = 75$$

Nilai warna “B” *Pixel 2*

Nilai warna “B” pada *pixel 2* bernilai desimal 82 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu $63.82 = 01010010$

$$63 = \underline{00111111} \oplus$$

$$01101101 = 109$$

Berdasarkan proses enkripsi manual pada *plainimage* sampel 2×1 *pixel*, didapati nilai *pixel* baru yang dinamakan dengan *cipherimage* sebagai berikut:

Tabel 4.6 Proses Nilai *Pixel Cipherimage*

Pixel	Warna	Cipher
		Desimal
1	R	107
	G	78
	B	148

2	R	213
	G	75
	B	109

4.2. Proses Dekripsi Manual Algoritma Spritz

Pada proses dekripsi algoritma spritz dilakukan dengan cara yang pada saat enkripsi. Kunci yang digunakan harus sama dengan kunci saat enkripsi. Adapun kunci yang digunakan adalah “2019”. Tahap awal proses dekripsi adalah pengacakan tabel *Key Scheduling Algorithm* (KSA) sebanyak 256 kali dengan nilai desimal kunci menggunakan rumus *Key Scheduling Algorithm* (KSA), dilanjutkan dengan tahap *Pseudo-Random Generation Algorithm* (PRGA) untuk mendapatkan nilai z akhir yang akan di XOR dengan nilai dari RGB *cipherimage* pada tahap dekripsi untuk mendapatkan *plainimage*.

Berhubungan proses KSA dan PRGA dilakukan dengan cara yang sama dengan proses sebelumnya, maka pada tahap dekripsi manual algoritma *Spritz* langsung menuju proses ketika yaitu melakukan XOR nilai z dengan nilai *cipherimage* seperti dibawah ini :

<i>Cipherimage</i>	Pixel 1			Pixel 2		
	R	G	B	R	G	B
Desimal	107	78	148	213	75	109
Biner	01101011	01001110	10010100	11010101	01001011	01101101

<i>Output z</i>	34	31	214	148	7	63
Biner	00100010	00011111	11010110	10010100	00000111	00111111

Nilai warna “R” *Pixel 1*

Nilai warna “R” pada *pixel 1 cipherimage* bernilai desimal 107 akan di XOR dengan nilai *outup z* hasil perhitungan proses PRGA yang pertama yaitu 34.

$$107 = 01101011$$

$$34 = \underline{00100010} \oplus$$

$$01001001 = 73$$

Nilai warna “G” *Pixel 1*

Nilai warna “G” pada *pixel 1 cipherimage* bernilai desimal 78 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 31.

$$78 \quad = 01001110$$

$$31 \quad = \underline{00011111} \oplus$$

$$01010001 \quad = 81$$

Nilai warna “B” *Pixel 1*

Nilai warna “B” pada *pixel 1 cipherimage* bernilai desimal 148 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 214.

$$148 \quad = 10010100$$

$$214 \quad = \underline{11010110} \oplus$$

$$01000010 \quad = 66$$

Nilai warna “R” *Pixel 2*

Nilai warna “R” pada *pixel 2 cipherimage* bernilai desimal 213 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 148.

$$213 \quad = 11010101$$

$$148 \quad = \underline{10010100} \oplus$$

$$01000001 \quad = 65$$

Nilai warna “G” *Pixel 2*

Nilai warna “G” pada *pixel 2 cipherimage* bernilai desimal 75 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 7.

$$75 \quad = 01001011$$

$$7 \quad = \underline{00000111} \oplus$$

$$01001100 \quad = 76$$

Nilai warna “B” *Pixel 2*

Nilai warna “B” pada *pixel 2 cipherimage* bernilai desimal 109 akan di XOR dengan nilai *output z* hasil perhitungan proses PRGA yang pertama yaitu 63.

$$109 \quad = 01101101$$

$$63 \quad = \underline{00111111} \oplus$$

$$01010010 \quad = 82$$

Berdasarkan proses dekripsi manual pada *cipherimage* sampel 2 x 1 *pixel*, didapati nilai *pixel* awal sebagai *plainimage*:

Tabel 4.7 Proses Nilai *Pixel Cipherimage*

Pixel	Warna	Plain
		Desimal
1	R	73
	G	81
	B	66
2	R	65
	G	76
	B	82

4.3. Perancangan Interface

Perancangan *interface input output* bertujuan untuk merancang antar muka aplikasi yang hendak dibangun kedalam sebuah perangkat lunak sehingga lebih mudah dalam pembuatan aplikasi dan mudah dimengerti. Berikut adalah bentuk rancangan *input* dan *output* sistem yang nantinya akan diimplemetasikan kedalam sebuah aplikasi.

1. Rancangan Menu Utama

Menu utama adalah *form* yang akan muncul ketika program aplikasi dibuka pertama kali. Adapun tampilan rancangan *form* menu utama adalah sebagai berikut :

Gambar 4.2 Rancangan Menu Utama

Berdasarkan gambar 4.3 di atas dapat dijelaskan keterangan sebagai berikut :

- a. Menu yang menampilkan *form* untuk melakukan proses enkripsi.
 - b. Menu yang menampilkan *form* untuk melakukan proses dekripsi.
 - c. Menu yang menampilkan *form* cara pemakaian aplikasi.
 - d. Menu yang menampilkan *form* tentang penulis dan perancang aplikasi.
 - e. Menu untuk menutup aplikasi.
 - f. Logo gambar Universitas Islam Sumatera Utara
2. Rancangan *Form* Enkripsi

Form enkripsi berfungsi sebagai *interface* bagi pengguna aplikasi pada saat melakukan kegiatan enkripsi citra digital. Melalui *form* enkripsi user dapat memilih gambar yang akan dienkripsi, memasukan kunci rahasia. Adapun rancangan *interface form* enkripsi dapat dilihat pada gambar berikut :

Gambar 4.3 Rancangan *Form* Enkripsi

The diagram shows a window titled "Form Enkripsi" with a close button "X" in the top right corner. The interface is divided into several sections:

- Sumber:** A text input field labeled "b" for entering the source of the plain image.
- Pilih Gambar:** A button labeled "a" for selecting a plain image.
- Plainimage:** A large rectangular area labeled "c" for displaying the selected plain image.
- Cipherimage:** A large rectangular area labeled "d" for displaying the encrypted cipher image.
- Informasi Plainimage:** A section containing two labels: "File Name:" with a text input field labeled "e", and "Kapasitas:" with a text input field labeled "f".
- informasi Kunci Enkripsi:** A text input field labeled "g" for entering the encryption key.
- Reset Informasi:** A button labeled "h" for resetting the key information.
- Proses:** A section containing two buttons: "ENKRIPSI PLAINIMAGE" labeled "i" and "SIMPAN CIPHERIMAGE" labeled "n".
- Progress:** A progress bar labeled "i" for showing the encryption progress.
- Status:** A section containing a label "Status :- Timer/second" and a text input field labeled "k".
- Kembali:** A button labeled "m" for returning to the previous screen.

Berdasarkan gambar 4.3 di atas dapat dijelaskan keterangan sebagai berikut :

- a. *Button* untuk memilih citra *plainimage* dari media penyimpanan.
- b. *TextBox* untuk menampilkan informasi sumber *plainimage*.

- c. *PictureBox* untuk menampilkan citra *plainimage*.
- d. *PictureBox* untuk menampilkan citra *chiperimage*.
- e. *Label* untuk menampilkan nama citra *plainimage*.
- f. *Label* untuk menampilkan ukuran citra *plainimage*.
- g. *TextBox* untuk menginput kunci rahasia.
- h. *Button* untuk menghapus informasi kunci rahasia.
- i. *Button* untuk melakukan proses enkripsi pada citra *plainimage*.
- j. *ProgressBar* untuk menampilkan progres enkripsi.
- k. *TextBox* untuk menampilkan waktu enkripsi.
- l. *Label* untuk menunjukkan status proses enkripsi..
- m. *Button* untuk kembali ke menu utama aplikasi.
- n. *Button* untuk menyimpan hasil enkripsi.

3. Rancangan *Form* Dekripsi

Form dekripsi berfungsi sebagai *interface* bagi pengguna aplikasi pada saat melakukan kegiatan dekripsi citra digital hasil enkripsi. Melalui *form* dekripsi user dapat memilih gambar yang akan didekripsi, memasukan kunci rahasia. Adapun rancangan *interface form* dekripsi dapat dilihat pada gambar berikut:

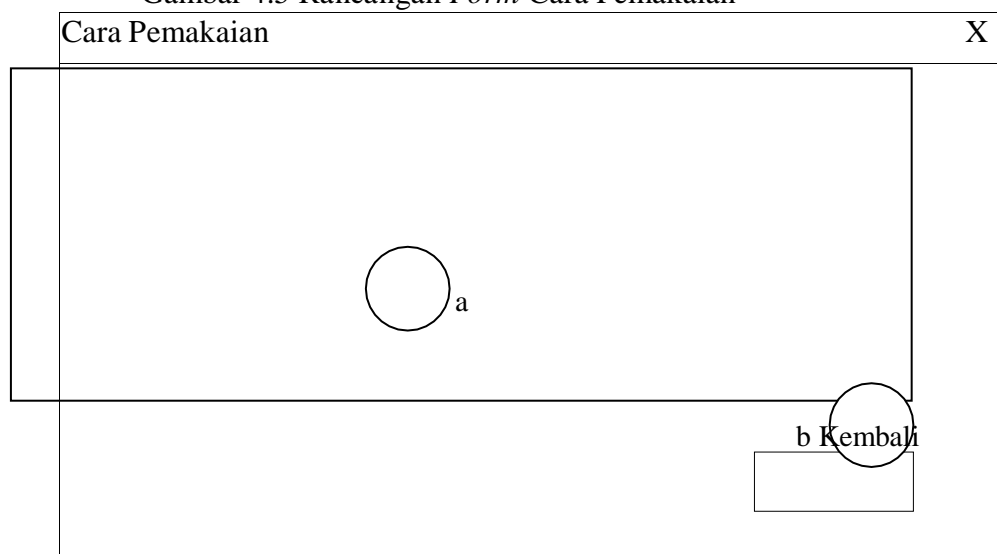
Gambar 4.4 Rancangan *Form* Dekripsi

Berdasarkan gambar 4.4 di atas dapat dijelaskan keterangan sebagai berikut :

- a. *Button* untuk memilih citra *chiperimage* dari media penyimpanan.
 - b. *TextBox* untuk menampilkan informasi sumber *chiperimage*.
 - c. *PictureBox* untuk menampilkan citra *chiperimage*.
 - d. *PictureBox* untuk menampilkan citra *plainimage*.
 - e. *Label* untuk menampilkan nama citra *chiperimage*.
 - f. *Label* untuk menampilkan ukuran citra *chiperimage*.
 - g. *TextBox* untuk menginput kunci rahasia.
 - h. *Button* untuk menghapus informasi kunci rahasia.
 - i. *Button* untuk melakukan proses dekripsi pada citra *chiperimage*.
 - j. *ProgressBar* untuk menampilkan progres dekripsi.
 - k. *TextBox* untuk menampilkan waktu dekripsi.
 - l. *Label* untuk menunjukkan status proses dekripsi..
 - m. *Button* untuk kembali ke menu utama aplikasi.
 - n. *Botton* untuk menyimpan hasil dekripsi.
4. Rancangan *Form* Cara Pemakaian

Form cara pemakaian ini berfungsi untuk menampilkan rincian cara pemakaiandari aplikasi enkripsi dan dekripsi citra digital. Adapun rancangan *interface* dari *form* cara pemakaian dapat dilihat pada gambar berikut ini :

Gambar 4.5 Rancangan *Form* Cara Pemakaian



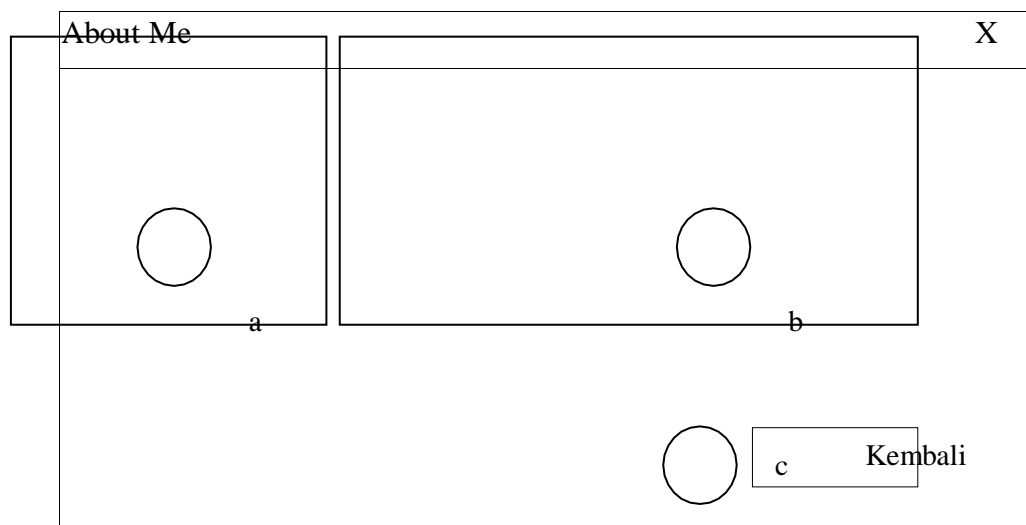
Berdasarkan gambar 4.5 di atas dapat dijelaskan keterangan sebagai berikut :

- a. *RichTextBox* untuk menampilkan informasi dari cara pemakaian aplikasi.
- b. *Button* untuk kembali pada menu utama aplikasi.

5. Rancangan *Form About Me*

Form about me berfungsi untuk menampilkan informasi tentang penulis dan perancangan aplikasi. Adapun rancangan *interface* pada *form about me* dapat dilihat pada gambar berikut :

Gambar 4.6 Rancangan *Form About Me*



Berdasarkan gambar 4.6 di atas dapat dijelaskan keterangan sebagai berikut :

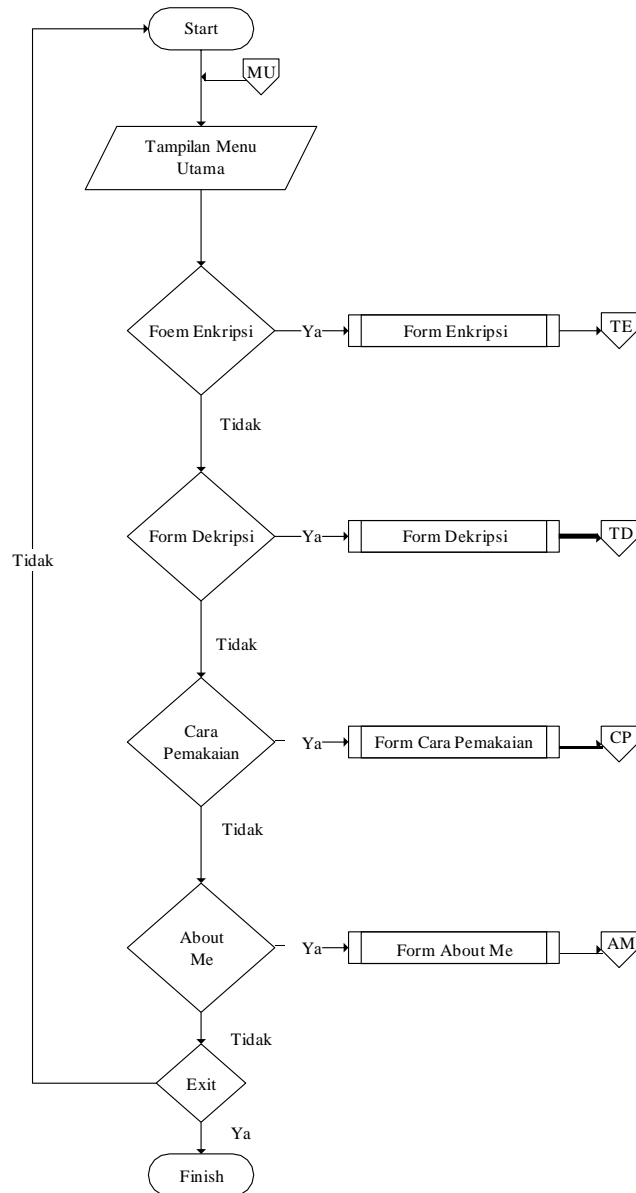
- a. *PictureBox* untuk menampilkan gambar perancang aplikasi.
- b. *RichTextBox* untuk menampilkan informasi data diri perancang aplikasi.
- c. *Button* untuk kembali pada menu utama aplikasi.

4.4. Flowchart Sistem

Flowchart sistem adalah bagian-bagian yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. Tahapan pengamanan data citra digital dilakukan dengan enkripsi dan dekripsi menggunakan algoritma *Spritz*. Adapun langkah-langkah dari enkripsi dan dekripsi menggunakan algoritma *Spritz* dapat dilihat pada gambaran *flowchart*.

1. Flowchart Sistem Keseluruhan

Flowchart Sistem aplikasi keseluruhan menjelaskan tentang alur yang akan dirancang dengan antarmuka tampilan *form* pertama yang muncul ketika aplikasi dibuka oleh *user*. Adapun alur-alur dari sistem aplikasi keseluruhan dapat dijelaskan pada gambar *flowchart* dibawah ini :



Gambar 4.7 *Flowchart* Sistem

Berdasarkan pada gambar *Flowchart* Sistem di atas dapat dijelaskan sebagai berikut :

a. Tampilan Menu Utama

Merupakan *form* yang akan tampil ketika pertama kali aplikasi dibuka oleh *user* yang menggunakan.

b. Form Enkripsi

Merupakan *form* enkripsi yang digunakan *user* untuk pemilihan gambar *plainimage* serta proses enkripsi gambar *plainimage*.

c. Form Dekripsi

Merupakan *form* dekripsi yang digunakan *user* untuk pemilihan gambar *chiperimage* serta proses dekripsi gambar *chiperimage*.

d. Cara Pemakaian

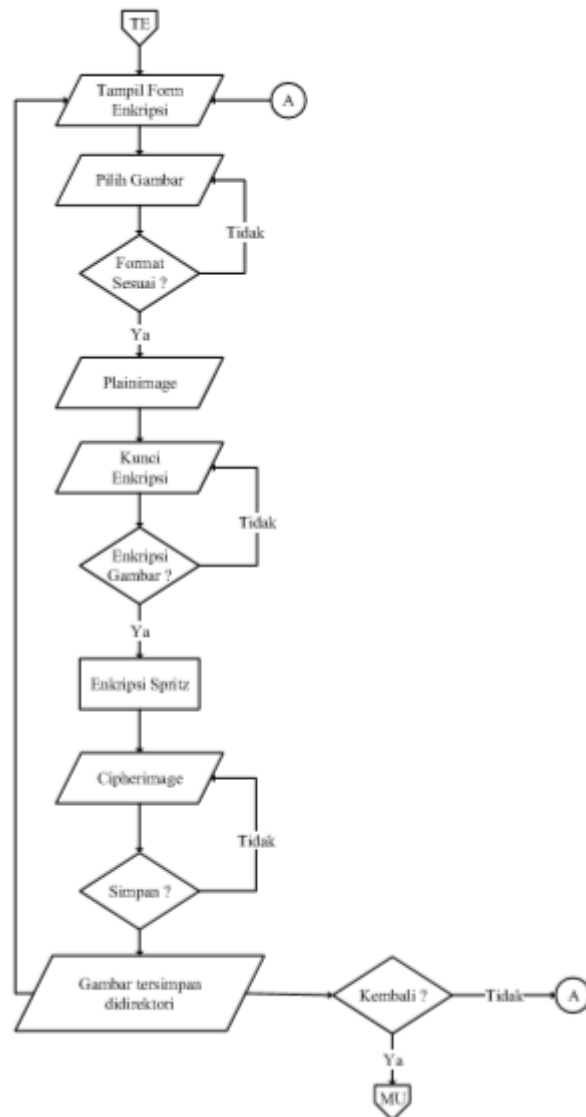
Merupakan *form* yang akan menampilkan cara pemakaian dari keseluruhan sistem aplikasi yang akan dibangun.

e. About Me

Merupakan *form* yang akan menampilkan data tentang perancang aplikasi yang akan dibangun.

2. *Flowchart* Enkripsi

Flowchart proses enkripsi pada aplikasi keamanan citra digital menggunakan algoritma *Spritz* keseluruhannya dapat dilihat pada gambar dibawah ini :



Gambar 4.8 *Flowchart* Enkripsi

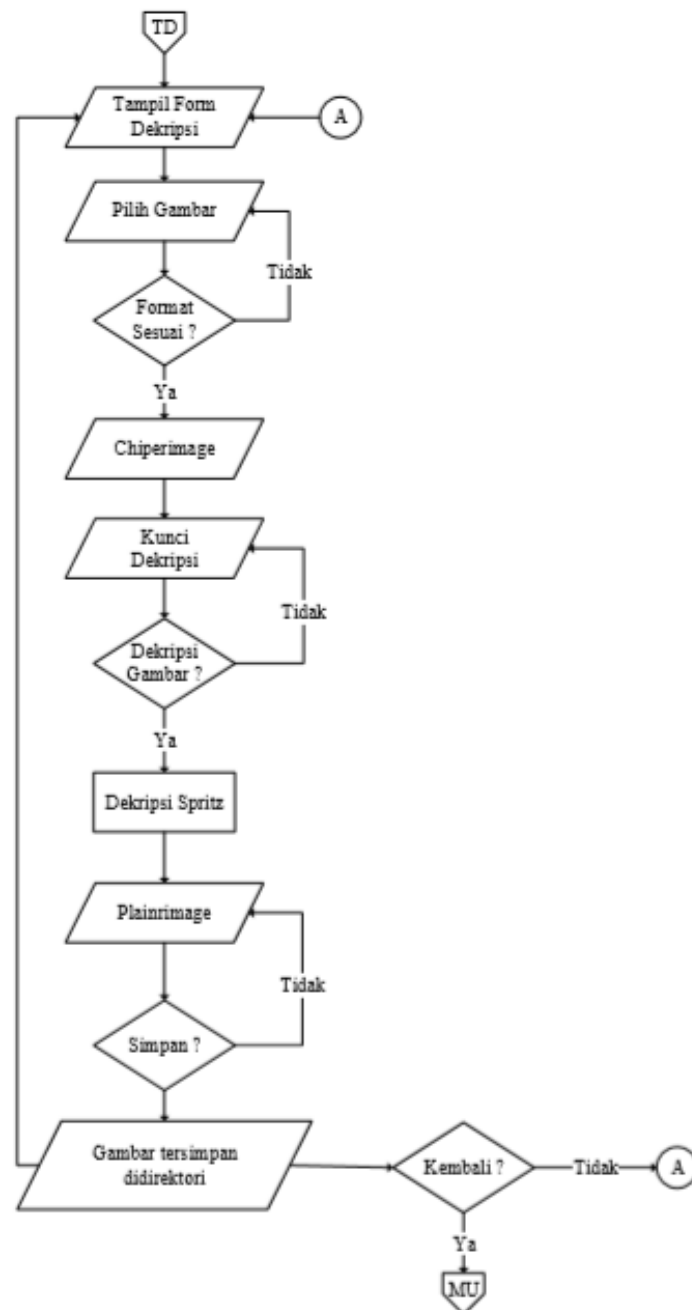
Proses enkripsi pada gambar di atas dapat dijelaskan sebagai berikut :

- Masukan gambar rahasia yang akan di enkripsi. Format gambar sudah ditentukan yaitu, jpg, png, dan bmp.
- Masukan kunci rahasia yang sudah ditentukan untuk proses enkripsi.
- Proses enkripsi menggunakan algoritma *Spritz* terhadap *plainimage* sehingga didapat *cipherimage Spritz*.

3. Flowchart Dekripsi

Flowchart proses dekripsi pada aplikasi keamanan citra digital menggunakan algoritma Spritz keseluruhan dapat dilihat pada gambar berikut :

Gambar 4.9 Flowchart Dekripsi

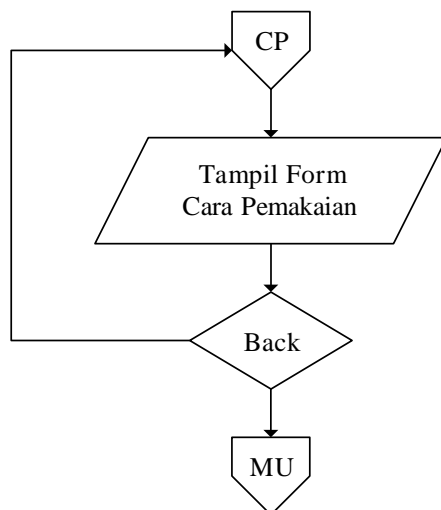


Proses dekripsi pada gambar diatas dapat dijelaskan sebagai berikut :

- a. Masukkan gambar hasil enkripsi yang akan di dekripsi. Format gambar sudah ditentukan yaitu bmp.
- b. Masukan kunci rahasia yang sudah ditentukan untuk proses dekripsi
- c. Proses dekripsi menggunakan algoritma *Spritz* terhadap *cipherimage* sehingga didapatkan *plainimage* awal dan gambar dapat disimpan.

4. *Flowchart* Cara Pemakaian

Flowchart cara pemakaian akan menjelaskan aliran kegiatan yang dapat dilakukan oleh *user* pada saat mengakses *form* aplikasi ini. Berikut *flowchart* cara pemakaian :

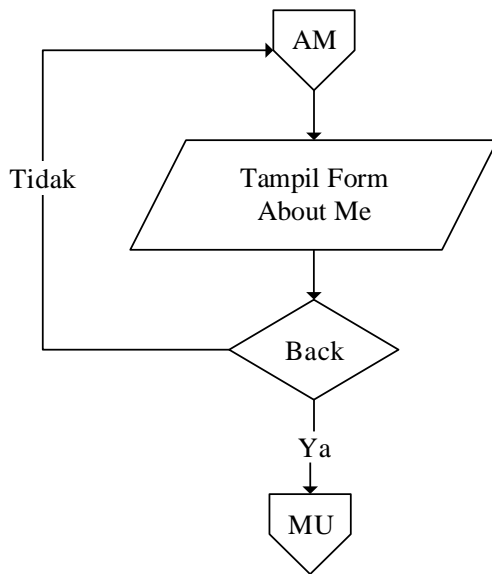


Gambar 4.10 *Flowchart* Cara Pemakaian

5. *Flowchart* About Me

Flowchart about me akan menampilkan data perancang aplikasi yang dibangun.

Berikut *flowchart* About Me :



Gambar 4.11 *Flowchart About Me*

4.5. Tampilan Program Aplikasi

Tampilan program aplikasi adalah tampilan aplikasi yang telah jadi yang sesuai dengan perancangan.

4.5.1. Tampilan Input Program Aplikasi

Tampilan *input* pada pembahasan ini terdiri dari tampilan menu utama, tampilan *form* enkripsi, tampilan *form* dekripsi, tampilan *form* cara pemakaian dan tampilan *form about me*. Kesimpulan yang diambil adalah apakah tampilan program sesuai dengan rancangan sebelumnya. Adapun tampilan keseluruhan menu program sebagai berikut :

1. Tampilan *Menu* Utama

Menu utama adalah tampilan yang akan pertama kali muncul ketika program aplikasi dibuka oleh *user*. Adapun tampilan *menu* utama aplikasi adalah sebagai berikut :

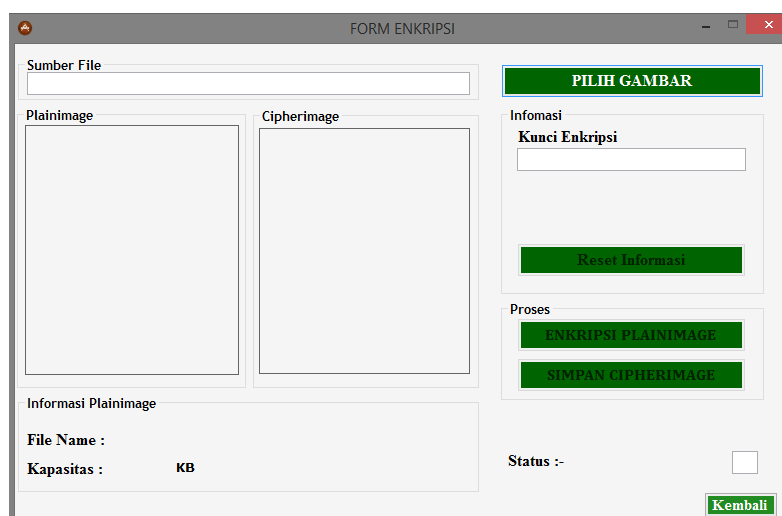


Gambar 4.12 Tampilan Menu Utama

Berdasarkan pada gambar tampilan menu utama, didapati beberapa menu diantara, menu *form* enkripsi yang digunakan untuk proses enkripsi citra, menu *form* dekripsi yang digunakan untuk proses dekripsi citra, menu cara pemakaian yang digunakan untuk mengetahui tentang pemakaian aplikasi, menu *about me* untuk mengetahui tentang penulis dan *host name* serta IP yang digunakan komputer tersebut.

2. Tampilan *Form* Enkripsi

Tampilan *form* enkripsi adalah menu yang digunakan untuk proses enkripsi citra digital. Adapun tampilan menu enkripsi dapat dilihat pada gambar di bawah ini:

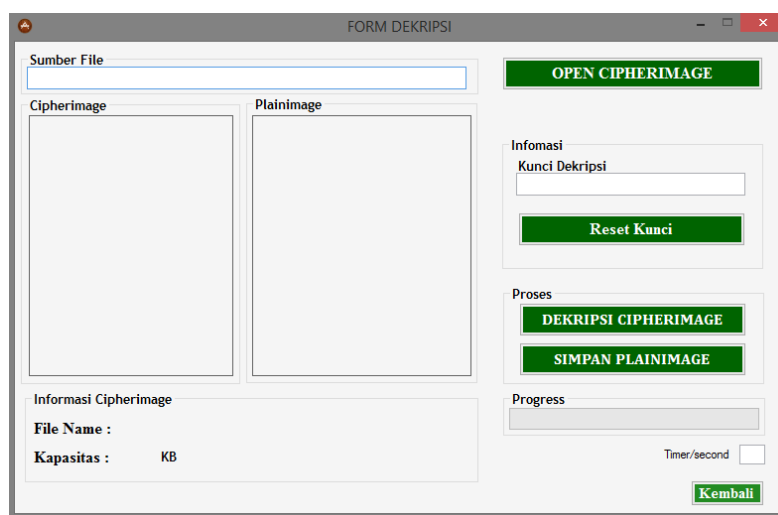


Gambar 4.13 Tampilan *Form* Enkripsi

Berdasarkan pada tampilan *form* enkripsi terdiri dari beberapa *button*. *Button* pilih gambar berfungsi untuk memilih gambar citra *image* yang akan dienkripsi. *Button* reset informasi berfungsi untuk menghapus informasi kunci. *Button* enkripsi berfungsi untuk melakukan proses enkripsi terhadap citra digital yang sudah dipilih. *Button* simpan berfungsi untuk menyimpan citra *cipherimage* didalam direktori komputer.

3. Tampilan *Form* Dekripsi

Tampilan *form* dekripsi adalah menu yang digunakan untuk proses dekripsi citra digital. Adapun tampilan menu enkripsi dapat dilihat pada gambar di bawah ini

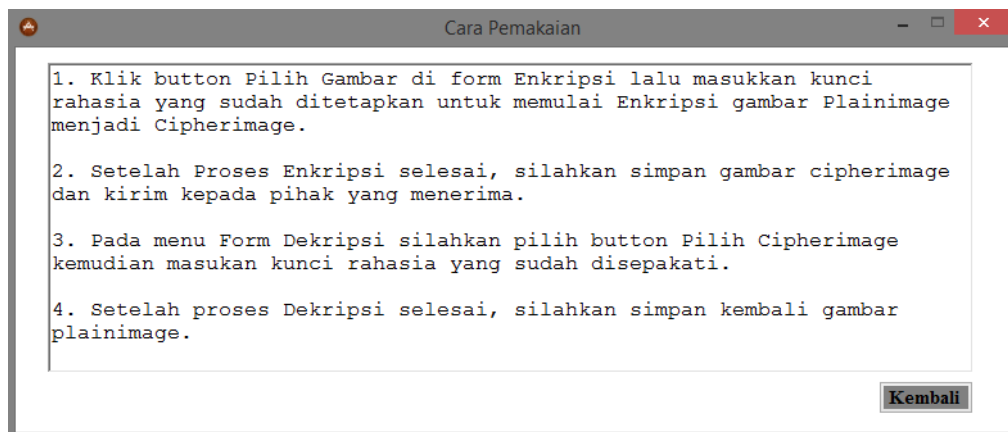


Gambar 4.14 Tampilan *Form* Dekripsi

Berdasarkan pada tampilan *form* dekripsi terdiri dari beberapa *button*. *Button* *Open Cipherimage* berfungsi untuk mencari *cipherimage* yang akan didekripsi. *Button* reset kunci berfungsi untuk menghapus kunci. *Button* dekripsi berfungsi untuk melakukan proses dekripsi terhadap citra *cipherimage* yang sudah dipilih. *Button* simpan berfungsi untuk menyimpan citra *plainimage* hasil dekripsi didalam direktori komputer.

4. Tampilan Cara Pemakaian

Tampilan cara pemakaian ini berfungsi untuk memberikan informasi terhadap pengguna aplikasi bagaimana cara menggunakan aplikasi secara singkat. Adapun tampilan cara pemakaian didalam aplikasi yang telah dibangun sebagai berikut :



Gambar 4.15 Tampilan Cara Pemakaian

Berdasarkan tampilan cara pemakaian terdapat *button* kembali yang berfungsi untuk kembali ke menu utama.

5. Tampilan *About Me*

Tampilan *about me* ini berfungsi untuk memberikan informasi terhadap pengguna aplikasi tentang penulis skripsi secara singkat. Adapun tampilan *aboutme* didalam aplikasi yang telah dibangun sebagai berikut :

Gambar 4.16 Tampilan *About Me*

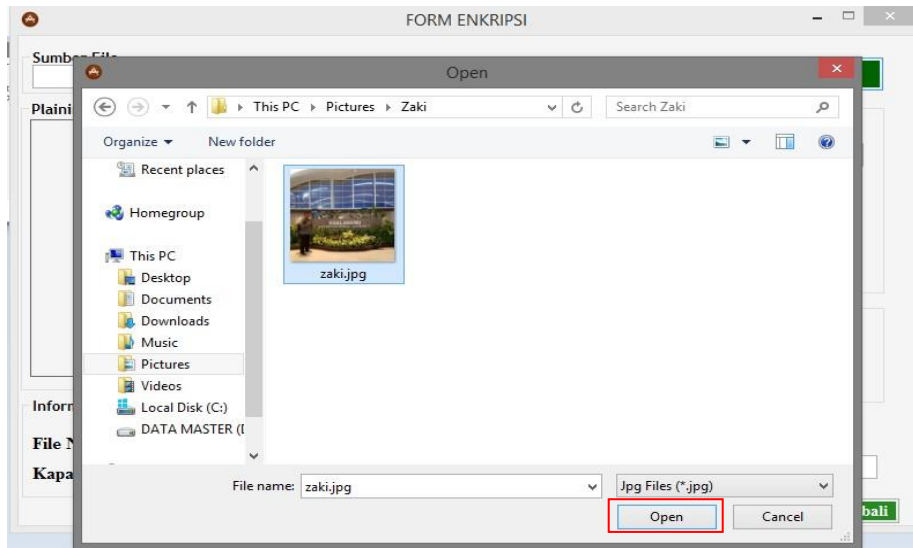
Berdasarkan tampilan *about me* terdapat *button* kembali yang berfungsi untuk kembali ke menu utama.

4.5.2. Proses Enkripsi Pada Aplikasi

Pada proses ini adalah enkripsi citra digital dengan algoritma *Spritz* menggunakan aplikasi yang telah dibangun. Proses enkripsi citra memiliki beberapa tahap yaitu, proses pemilihan citra digital awal (*plainimage*), proses pemasukan kunci, proses enkripsi dan proses simpan.

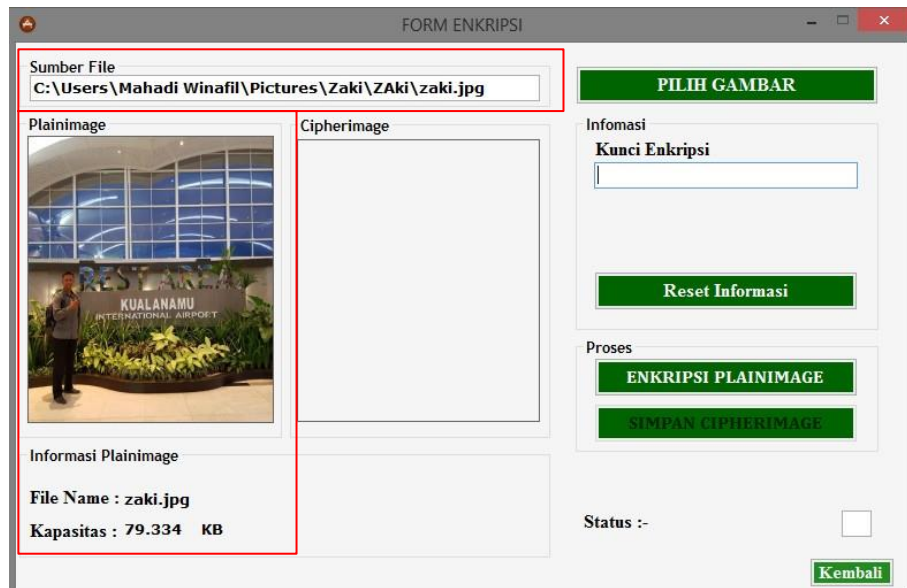
1. Pemilihan *Plainimage*

Adapun proses pemilihan citra awal (*plainimage*) dilakukan dengan memilih *Button* pilih gambar hingga muncul gambar seperti di bawah ini :



Gambar 4.17 Tampilan Pemilihan Citra *Plainimage*

Berdasarkan pada gambar di atas, untuk memilih citra *plainimage* dengan menekan *button Open*. Hingga seperti gambar di bawah :



Gambar 4.18 Tampilan Citra *Plainimage*

Berdasarkan pada gambar 4.18 proses pemilihan dengan memilih tombol pilih gambar. Hasil dari pemilihan berupa *plainimage* dengan informasi nama *file*

adalah zaki ekstensi .jpg dan kapasitasnya adalah 79,334 Kb.

2. Memasukan Kunci

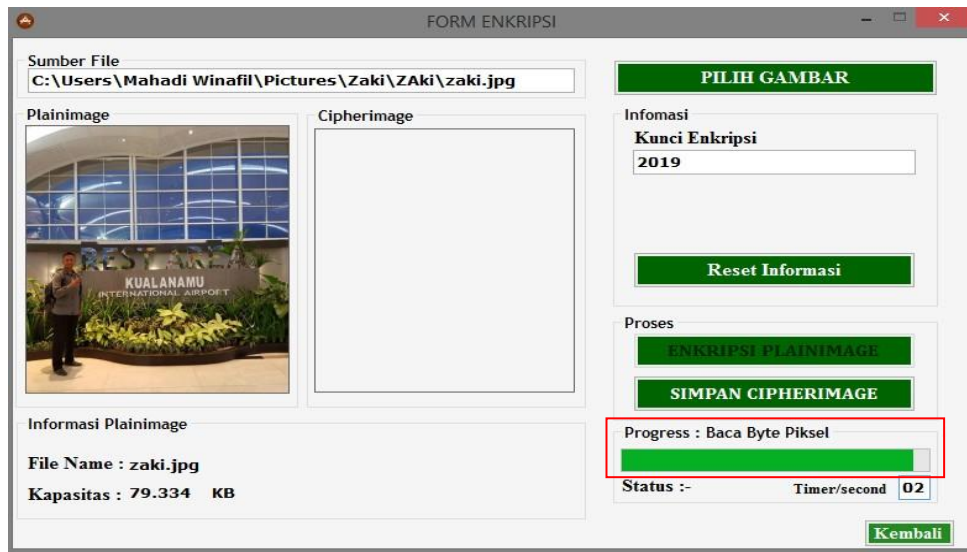
Setelah proses pemilihan citra *plainimage* proses selanjutnya adalah memasukan kunci enkripsi. Adapun kunci enkripsi dapat dilihat pada gambar di bawah ini :

Gambar 4.19 Tampilan proses memasukan kunci

Berdasarkan pada gambar di atas kunci yang digunakan untuk proses enkripsi adalah “2019”.

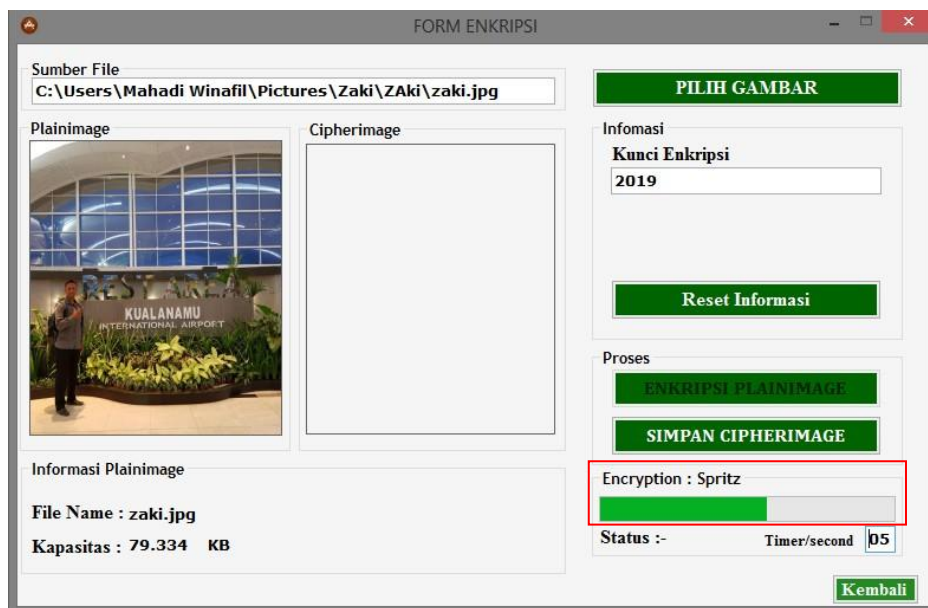
3. Enkripsi *Plainimage*

Enkripsi *plainimage* dilakukan menggunakan algoritma *Spritz* sehingga didapati *cipherimage*. Sebelum melakukan proses enkripsi, terlebih dahulu aplikasi akan membaca *byte* dari *pixel* yang diinputkan. Adapun proses enkripsi tersebut dapat dilihat pada gambar di bawah ini :



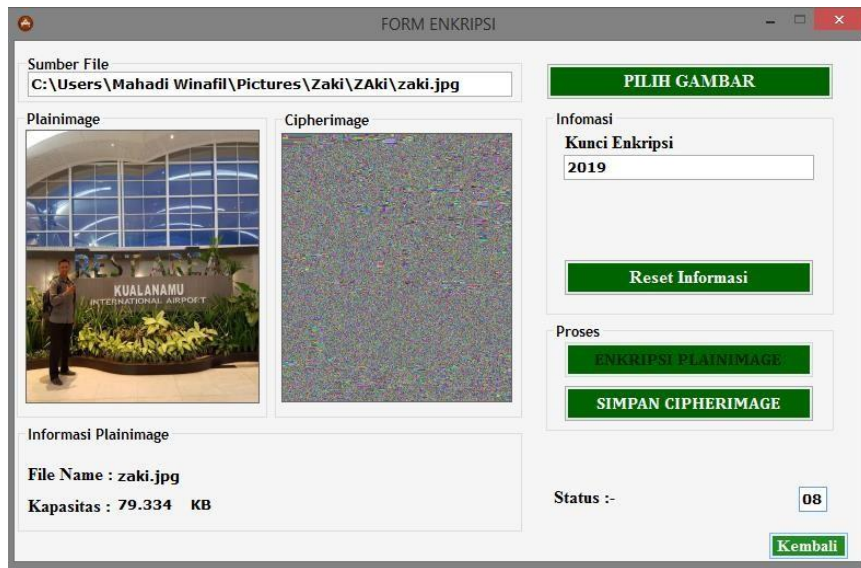
Gambar 4.20 Tampilan Proses Enkripsi *Spritz*

Berdasarkan pada gambar di atas proses pertama adalah pembacaan *byte pixel* yang kemudian dienkripsi menggunakan algoritma *Spritz* seperti gambar di bawah ini :



Gambar 4.21 Tampilan Proses Enkripsi *Spritz*

Berdasarkan pada gambar 4.21 di atas, proses kedua enkripsi dilakukan dengan algoritma *Spritz* sehingga didapati *cipherimage* sebagai berikut :

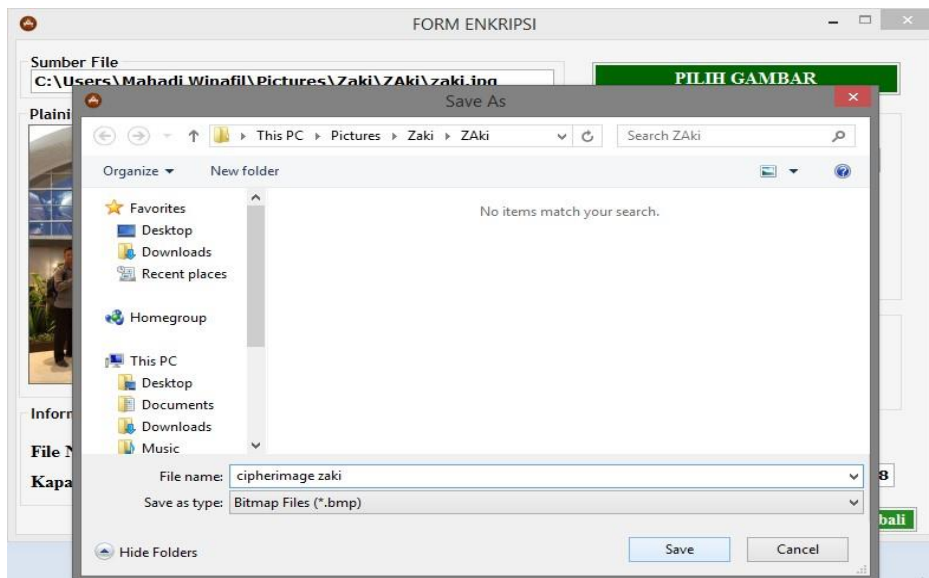


Gambar 4.22 Tampilan Proses Enkripsi *Spritz* Selesai

Berdasarkan pada gambar di atas, enkripsi citra digital menggunakan algoritma *spritz* selesai dilakukan. Sehingga menghasilkan *cipherimage* yang berada disamping *plainimage*. Adapun waktu yang dibutuhkan ada 8 detik. Sifat waktu sangat relatif, sesuai dengan ukuran *plainimage*.

4. Simpan *Cipherimage*

Proses selanjutnya adalah menyimpan *cipherimage* dengan mengklik *button* simpan sehingga tampil menu seperti gambar di bawah ini :



Gambar 4.23 Tampilan Proses Simpan *Cipherimag*

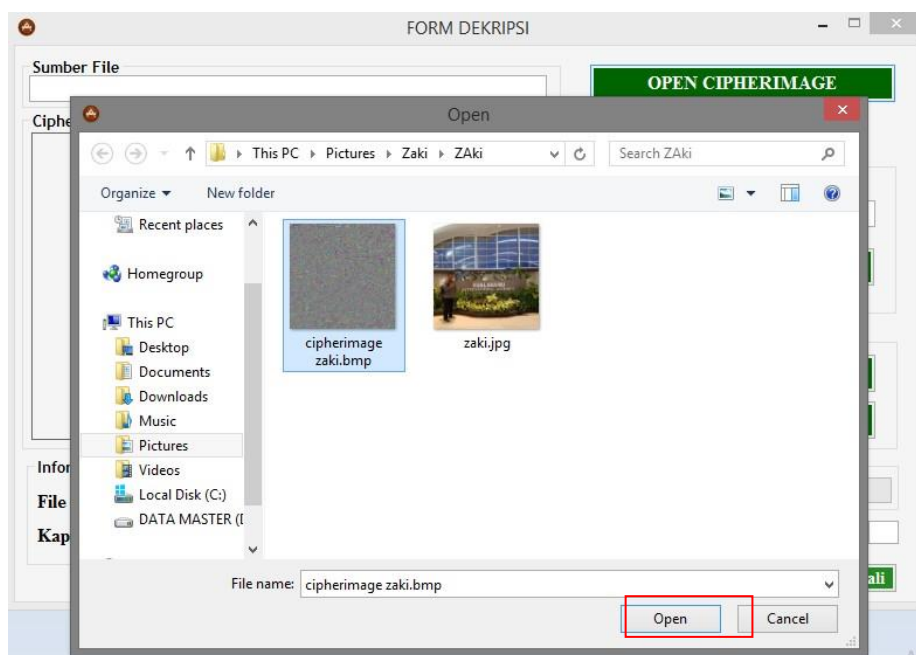
Berdasarkan pada gambar di atas, *user* hanya memilih dan menekan *button save* untuk menyimpan *cipherimage* didalam direktori.

4.5.3. Proses Dekripsi Pada Aplikasi

Pada proses ini adalah dekripsi citra digital *cipherimage* dengan algoritma *Spritz* menggunakan aplikasi yang telah dibangun. Proses dekripsi citra *cipherimage* memiliki beberapa tahap yang sama dengan proses enkripsi yaitu, proses pemilihan citra digital (*cipherimage*), proses pemasukan kunci, proses dekripsi dan proses simpan.

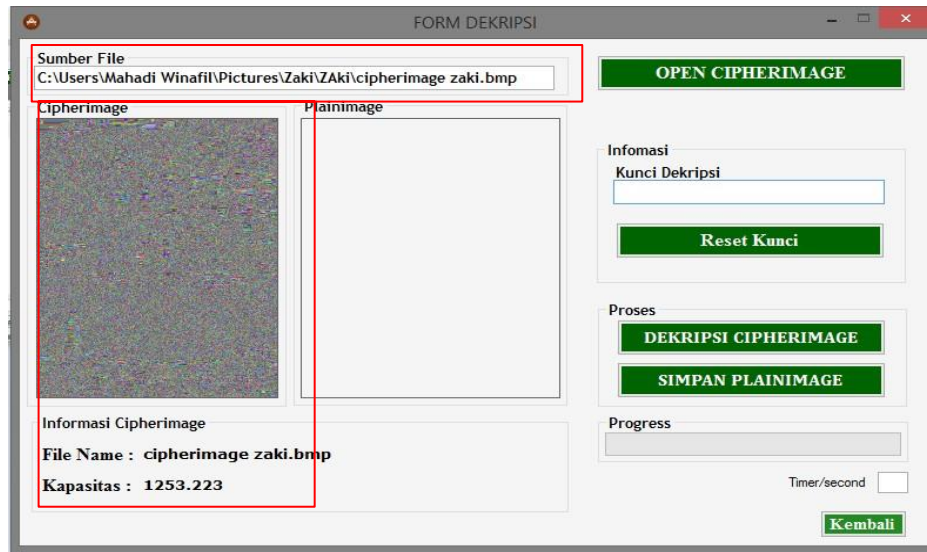
1. Pemilihan *Plainimage*

Adapun proses pemilihan citra awal (*cipherimage*) dilakukan dengan memilih *Button open cipherimage* hingga muncul gambar seperti di bawah ini :



Gambar 4.24 Tampilan Pemilihan Citra *cipherimage*

Berdasarkan pada gambar di atas, untuk memilih citra *cipherimage* dengan menekan *button Open*. Hingga seperti gambar di bawah :

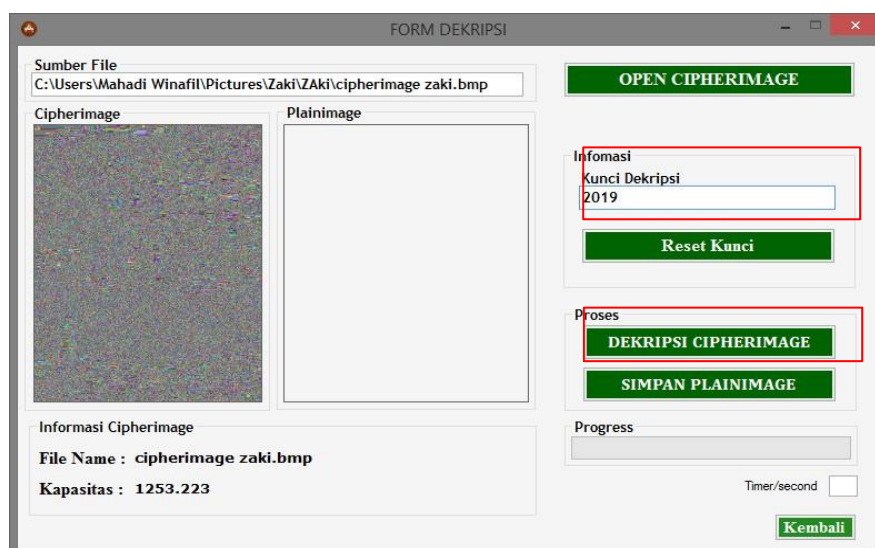


Gambar 4.25 Tampilan Citra *cipherimage*

Berdasarkan pada gambar 4.25 proses pemilihan dengan memilih tombol pilih gambar. Hasil dari pemilihan berupa *cipherimage* dengan informasi nama *file* adalah *cipherimagezaki* ekstensi *.bmp* dan kapasitasnya adalah 1253,223 Kb.

2. Memasukan Kunci

Setelah proses pemilihan citra *cipherimage* proses selanjutnya adalah memasukan kunci dekripsi yang sama ketika proses enkripsi. Adapun kunci dekripsi dapat dilihat pada gambar di bawah ini :

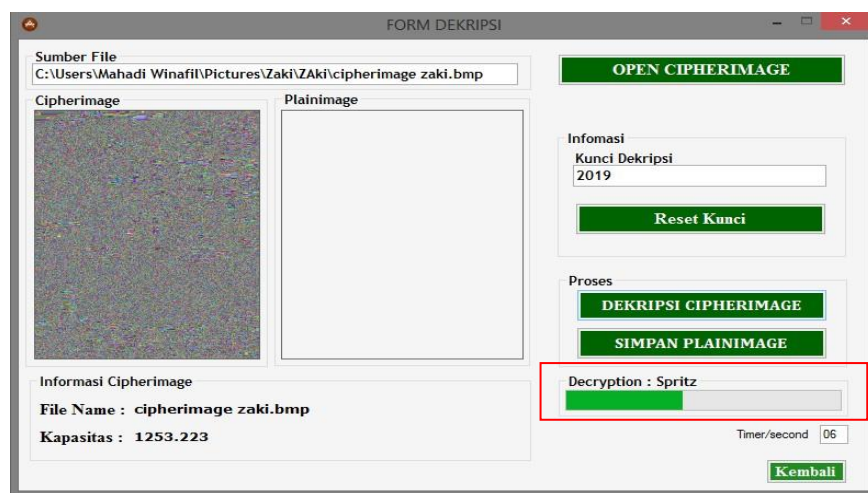


Gambar 4.26 Tampilan proses memasukan kunci

Berdasarkan pada gambar di atas kunci yang digunakan untuk proses dekripsi adalah “2019”.

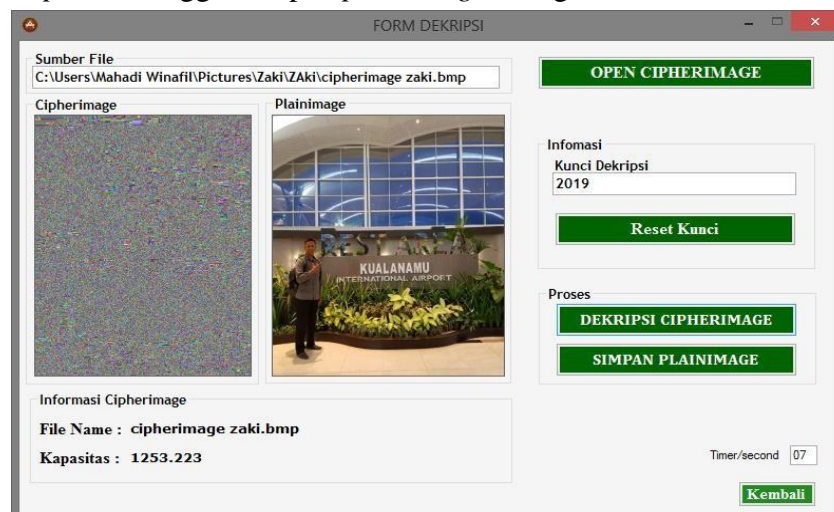
3. Dekripsi *Cipherimage*

Dekripsi *cipherimage* dilakukan menggunakan algoritma *Spritz* didapati sehingga didapati *plainimage*. Sebelum melakukan proses dekripsi, terlebih dahulu aplikasi akan membaca *byte* dari *pixel* yang diinputkan kemudian mendekripsi menggunakan algoritma *Spritz*. Adapun proses dekripsi tersebut dapat dilihat pada gambar di bawah ini :



Gambar 4.27 Tampilan Proses Dekripsi *Spritz*

Berdasarkan pada gambar 4.27 di atas, proses enkripsi dilakukan dengan algoritma *Spritz* sehingga didapati *plainimage* sebagai berikut :

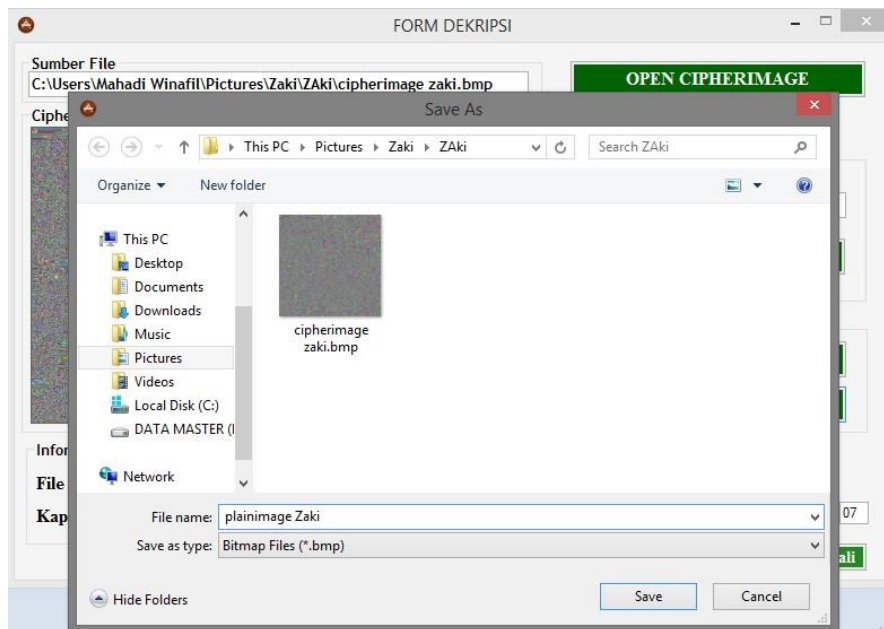


Gambar 4.28 Tampilan Proses Dekripsi *Spritz* Selesai

Berdasarkan pada gambar diatas, dekripsi citra digital *cipherimage* menggunakan algoritma *spritz* selesai dilakukan. Sehingga menghasilkan *plainimage* yang berada disamping *cipherimage*. Adapun waktu yang dibutuhkan ada 7 detik. Sifat waktu sangat relatif, sesuai dengan ukuran *cipherimage*.

4. Simpan *Plainimage*

Proses selanjutnya adalah menyimpan *plainimage* dengan mengklik *button* simpan sehingga tampil menu seperti gambar di bawah ini :



Gambar 4.29 Tampilan Proses Simpan *Plainimage*

Berdasarkan pada gambar di atas, *user* hanya memilih dan menekan *button save* untuk menyimpan *plainimage* didalam direktori. Adapun hasilnya seperti gambar bawah ini :



Gambar 4.30 Hasil Enkripsi dan Dekripsi Citra Digital


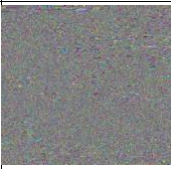

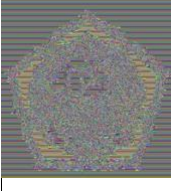


4.6. Hasil Pengujian Program Aplikasi

Hasil pengujian program meliputi dari hasil enkripsi, hasil dekripsi,. Pengujian enkripsi dilakukan dengan 3 sampel citra yang kemudian dilakukan dekripsi.

1. Hasil Pengujian Enkripsi Citra *Plainimage*

Hasil Pengujian enkripsi citra meliputi dari keterangan resolusi, *size* dan waktu enkripsi. Kunci yang digunakan adalah 2019. Contoh hasil dapat dilihat pada tabel di bawah ini :

Tabel 4.8 Pengujian Enkripsi Citra *Plainimage*

No	Plainimage	Eks	Resolusi	Size KB	cipherimage	Size KB	Waktu Enkripsi
1		Jpg	640 x 640	79		1253	8 Detik
2		jpg	483 x 512	76		405	7 Detik
3		jpg	774 x 774	52		1964	6 Detik

Berdasarkan pada tabel 4.8 di atas dapat di ambil kesimpulan :

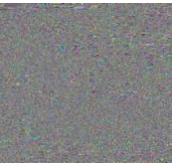

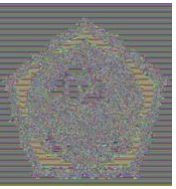



- Program aplikasi berhasil mengenkripsi sebuah *plainimage* menjadi sebuah *cipherimage*.
- Hasil enkripsi *cipherimage* sangat berbeda dengan *plainimage*, dimana *cipherimage* tidak dapat dikenali dengan pandangan mata manusia.
- Ukuran citra *plainimage* mengalami perubahan ketika proses enkripsi yang menjadi *cipherimage*. Ukuran *cipherimage* tersebut menjadi lebih besar dari *plainimage*.

d. Waktu enkripsi citra *plainimage* memiliki variasi, hal ini disebabkan karena proses enkripsi program aplikasi dipengaruhi oleh *size*, resolusi dan spesifikasi komputer.

2. Pengujian Dekripsi *Cipherimage*

Pengujian dekripsi *cipherimage* meliputi dari keterangan resolusi, *size* dan waktu dekripsi. Kunci yang digunakan sama saat proses enkripsi yaitu 2019. Adapun keterangan dekripsi dapat dilihat pada tabel di bawah ini :

Tabel 4.9 Pengujian Dekripsi Citra *Cipherimage*

No	Plainimage	Eks	Resolusi	SizeKB	cipherimage	Size KB	Waktu Dekripsi
1		bmp	640 x 640	1253		893	9 Detik
2		bmp	483 x 512	405		322	7 Detik
3		bmp	774 x 774	1964		1078	8 Detik

Berdasarkan pada tabel 4.9 di atas dapat di ambil kesimpulan :

- Program aplikasi berhasil mendekripsi sebuah *cipherimage* menjadi citra *plainimage* seperti semula.
- Ukuran citra *plainimage* hasil dekripsi mengalami perubahan dengan ukuran citra *cipherimage* sebelum proses dekripsi.
- Waktu dekripsi *cipherimage* memiliki sedikit perbedaan saat proses enkripsi.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan pengujian yang telah dilakukan pada bab sebelumnya, dapat disimpulkan bahwa :

1. Hasil enkripsi dan dekripsi citra digital menggunakan algoritma *Spritz* berupa sebuah *cipherimage* dan *plainimage* yang memiliki perubahan ukuran *size* lebih besar dari citra *plainimage* awal.
2. Proses waktu enkripsi dan dekripsi dipengaruhi oleh ukuran dan resolusi citra *plainimage* serta spesifikasi komputer yang menjalankan aplikasi. Semakin tinggi resolusi dari *plainimage* maka proses enkripsi dan dekripsinya semakin lama begitu pula sebaliknya.
3. Aplikasi kriptografi citra digital menggunakan algoritma *Spritz* berhasil menyamarkan citra digital yang dapat dilihat informasinya menjadi sebuah citra digital yang tidak dapat dikenali oleh indra mata manusia.

5.2. Saran

Adapun saran yang dapat diberikan oleh penulis skripsi untuk mengembangkan aplikasi ini adalah sebagai berikut :

1. Aplikasi enkripsi citra digital dapat ditambahkan menu *login admin* agar aplikasi hanya dapat digunakan oleh *user* yang mempunyai otoritas.
2. Proses enkripsi untuk saat ini masih memiliki waktu proses yang relatif lama, mengingat untuk saat sekarang ukuran *pixel* dari citra digital memiliki resolusi yang besar sehingga perlu adanya pengkajian ulang dari aplikasi dan algoritma yang dipakai agar pembacaan *pixel* dan enkripsi dapat dilakukan dengan lebih cepat.
3. Aplikasi yang dirancang dan dibangun pada penelitian ini berbasis *desktop windows*, mungkin dapat dikembangkan menjadi berbasis *android java* agar penggunaan aplikasi lebih efisien menggunakan *smartphone*.

DAFTAR PUSTAKA

- Agustina, E.R and Kurniati. A., 2009. *Pemanfaatan Kriptografi Dalam Mewujudkan Keamanan Informasi Pada e-Voting di Indonesia.*, Seminar Nasional Informatika, 22-28.
- Ahmad. U., 2005. *Pengolahan Citra Digital & Teknik Pemogramannya*, Yogyakarta:Graha Ilmu.
- Aleisa. N., 2015. *A Comparison of the 3DES and AES Encryption Standards.*, *International Jurnal of Security and Its Applications*, vol. 7, 241-246.
- Furqan. Mhd, Sriani dan Sofianah Harahap.L., 2020. *Klasifikasi Daun Bugenvil Menggunakan Gray Level Co-Occurrence Matrix dan K-Nearest Neighbor*, Vol.6, no.1, pp.22-29.
- Kiong. L.V., 2012 *Visual Basic 2010 Tutorial*. Dari:
<http://www.vbtutor.net/index.php/visual-basic-2010-tutorial/>, Diakses 24 November 2019.
- Mohtashim., 2015 *Cryptography Hash Functions*. Dari:
<https://www.tutorialspoint.com/cryptography/cryptography.htm>, Diakses 24 November 2019.
- Mohtashim., 2014 *UML Tutorial*. Dari:
<https://www.tutorialspoint.com/uml/index.htm>, Diakses 24 November 2019.
- Munir. R., 2006. *Kriptografi*, Bandung: InformatikaPamungkas. A., 2017 *Pengolahan Citra*. Dari:
<https://pemrogramanmatlab.com/2017/07/26/pengolahan-citra-digital/>, Diakses 24 November 2019.
- Prabowo. A., 2018. *Kriptografi-Jenis Jenis Serangan dalam Kriptografi*. Dari:
<http://www.sigitprabowo.id/2013/01/kriptografi-jenis-jenis-serangan-dalam.html>, Diakses 23 November 2020.

- Purnama. A., 2012. *Definisi dan Pengolahan Citra Digital*. Dari:
<http://elektronika-dasar.web.id/definisi-dan-pengolahan-citra-digital/>,
Diakses 24 November 2019
- Putra. D., 2010. *Pengolahan Citra Digital*, Yogyakarta: Andi Offset.
- R. A.S and M. Shalahudin., 2011. *Modul Pembelajaran: Rekayasa Perangkat Lunak(Terstruktur dan Berorientasi Objek)*, Bandung: Modula
- Setyaningsih. E., 2015. *Kriptografi & Implementasi Menggunakan Matlab*,
Yogyakarta: Andi.
- Sutanta. E., 2005. *Pengantar Teknologi Informasi*, Yogyakarta: Graha Ilmu
- Sutoyo. T., 2009. *Teori Pengolahan Citra Digital*, Yogyakarta: Andi.
- Zebua, Taronisokhi., 2018. *Encoding the Record Database of Computer Based Test Exam Based on Spritz Algorithm”* Vol.9, 52-62.

LAMPIRAN 1

LISTING PROGRAM

FORM ENKRIPSI DESIGNER

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class FormEnkripsi
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Dim resources As System.ComponentModel.ComponentResourceManager =
New System.ComponentModel.ComponentResourceManager(GetType(FormEnkripsi))
        Me.Button1 = New System.Windows.Forms.Button()
        Me.TextBox6 = New System.Windows.Forms.TextBox()
        Me.GroupBox3 = New System.Windows.Forms.GroupBox()
        Me.TextBox4 = New System.Windows.Forms.TextBox()
        Me.Button2 = New System.Windows.Forms.Button()
        Me.GroupBox4 = New System.Windows.Forms.GroupBox()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.GB1 = New System.Windows.Forms.GroupBox()
        Me.mili = New System.Windows.Forms.TextBox()
        Me.Label5 = New System.Windows.Forms.Label()
        Me.PB1 = New System.Windows.Forms.ProgressBar()
        Me.Button3 = New System.Windows.Forms.Button()
        Me.Button4 = New System.Windows.Forms.Button()
        Me.GroupBox5 = New System.Windows.Forms.GroupBox()
        Me.Label10 = New System.Windows.Forms.Label()
        Me.detik = New System.Windows.Forms.TextBox()
        Me.SaveFileDialog1 = New System.Windows.Forms.SaveFileDialog()
        Me.Timer2 = New System.Windows.Forms.Timer(Me.components)
        Me.BW2 = New System.ComponentModel.BackgroundWorker()
        Me.BW4 = New System.ComponentModel.BackgroundWorker()
        Me.Timer1 = New System.Windows.Forms.Timer(Me.components)
        Me.OpenFileDialog1 = New System.Windows.Forms.OpenFileDialog()
        Me.BW5 = New System.ComponentModel.BackgroundWorker()
```

```

Me.BW1 = New System.ComponentModel.BackgroundWorker()
Me.BW3 = New System.ComponentModel.BackgroundWorker()
Me.Button5 = New System.Windows.Forms.Button()
Me.GroupBox2 = New System.Windows.Forms.GroupBox()
Me.Label4 = New System.Windows.Forms.Label()
Me.Label3 = New System.Windows.Forms.Label()
Me.Label7 = New System.Windows.Forms.Label()
Me.Label8 = New System.Windows.Forms.Label()
Me.Label9 = New System.Windows.Forms.Label()
Me.PictureBox1 = New System.Windows.Forms.PictureBox()
Me.GroupBox1 = New System.Windows.Forms.GroupBox()
Me.GroupBox6 = New System.Windows.Forms.GroupBox()
Me.PictureBox2 = New System.Windows.Forms.PictureBox()
Me.GroupBox3.SuspendLayout()
Me.GroupBox4.SuspendLayout()
Me.GB1.SuspendLayout()
Me.GroupBox2.SuspendLayout()
CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).BeginInit()
Me.GroupBox1.SuspendLayout()
Me.GroupBox6.SuspendLayout()
CType(Me.PictureBox2,
System.ComponentModel.ISupportInitialize).BeginInit()
Me.SuspendLayout()
'
'Button1
'
Me.Button1.BackColor = System.Drawing.Color.DarkGreen
Me.Button1.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button1.ForeColor = System.Drawing.Color.White
Me.Button1.Location = New System.Drawing.Point(486, 20)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(263, 34)
Me.Button1.TabIndex = 2
Me.Button1.Text = "PILIH GAMBAR"
Me.Button1.UseVisualStyleBackColor = False
'
'TextBox6
'
Me.TextBox6.BackColor = System.Drawing.Color.White
Me.TextBox6.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.TextBox6.Location = New System.Drawing.Point(9, 16)
Me.TextBox6.Name = "TextBox6"
Me.TextBox6.Size = New System.Drawing.Size(443, 23)
Me.TextBox6.TabIndex = 42
'
'GroupBox3
'
Me.GroupBox3.Font = New System.Drawing.Font("Trebuchet MS",
9.75!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.GroupBox3.Location = New System.Drawing.Point(3, 12)
Me.GroupBox3.Name = "GroupBox3"

```

```

Me.GroupBox3.Size = New System.Drawing.Size(461, 44)
Me.GroupBox3.TabIndex = 43
Me.GroupBox3.TabStop = False
Me.GroupBox3.Text = "Sumber File"
'
'TextBox4
'
Me.TextBox4.BackColor = System.Drawing.Color.White
Me.TextBox4.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.TextBox4.Location = New System.Drawing.Point(16, 42)
Me.TextBox4.Name = "TextBox4"
Me.TextBox4.Size = New System.Drawing.Size(229, 23)
Me.TextBox4.TabIndex = 45
'
'Button2
'
Me.Button2.BackColor = System.Drawing.Color.DarkGreen
Me.Button2.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button2.ForeColor = System.Drawing.Color.White
Me.Button2.Location = New System.Drawing.Point(16, 137)
Me.Button2.Name = "Button2"
Me.Button2.Size = New System.Drawing.Size(229, 34)
Me.Button2.TabIndex = 47
Me.Button2.Text = "Reset Informasi"
Me.Button2.UseVisualStyleBackColor = False
'
'GroupBox4
'
Me.GroupBox4.Controls.Add(Me.Button2)
Me.GroupBox4.Controls.Add(Me.Label1)
Me.GroupBox4.Controls.Add(Me.TextBox4)
Me.GroupBox4.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GroupBox4.Location = New System.Drawing.Point(486, 62)
Me.GroupBox4.Name = "GroupBox4"
Me.GroupBox4.Size = New System.Drawing.Size(263, 188)
Me.GroupBox4.TabIndex = 48
Me.GroupBox4.TabStop = False
Me.GroupBox4.Text = "Infomasi"
'
'Label1
'
Me.Label1.AutoSize = True
Me.Label1.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label1.Location = New System.Drawing.Point(13, 21)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(108, 19)
Me.Label1.TabIndex = 0
Me.Label1.Text = "Kunci Enkripsi"
'

```



```

'GB1
'
Me.GB1.Controls.Add(Me.mili)
Me.GB1.Controls.Add(Me.Label5)
Me.GB1.Controls.Add(Me.PB1)
Me.GB1.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GB1.Location = New System.Drawing.Point(486, 358)
Me.GB1.Name = "GB1"
Me.GB1.Size = New System.Drawing.Size(263, 75)
Me.GB1.TabIndex = 49
Me.GB1.TabStop = False
Me.GB1.Text = "Progress"
'
'mili
'
Me.mili.Location = New System.Drawing.Point(231, 49)
Me.mili.Name = "mili"
Me.mili.Size = New System.Drawing.Size(26, 23)
Me.mili.TabIndex = 46
'
'Label5
'
Me.Label5.AutoSize = True
Me.Label5.Font = New System.Drawing.Font("Times New Roman", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label5.Location = New System.Drawing.Point(143, 54)
Me.Label5.Name = "Label5"
Me.Label5.Size = New System.Drawing.Size(82, 15)
Me.Label5.TabIndex = 55
Me.Label5.Text = "Timer/second"
'
'PB1
'
Me.PB1.BackColor = System.Drawing.Color.Olive
Me.PB1.Location = New System.Drawing.Point(6, 24)
Me.PB1.Name = "PB1"
Me.PB1.Size = New System.Drawing.Size(251, 22)
Me.PB1.TabIndex = 0
'
'Button3
'
Me.Button3.BackColor = System.Drawing.Color.DarkGreen
Me.Button3.Font = New System.Drawing.Font("Cambria", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button3.ForeColor = System.Drawing.Color.White
Me.Button3.Location = New System.Drawing.Point(502, 274)
Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(229, 34)
Me.Button3.TabIndex = 50
Me.Button3.Text = "ENKRIPSI PLAINIMAGE"
Me.Button3.UseVisualStyleBackColor = False
'
'Button4

```

```

    '
    Me.Button4.BackColor = System.Drawing.Color.DarkGreen
    Me.Button4.Font = New System.Drawing.Font("Cambria", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.Button4.ForeColor = System.Drawing.Color.White
    Me.Button4.Location = New System.Drawing.Point(502, 314)
    Me.Button4.Name = "Button4"
    Me.Button4.Size = New System.Drawing.Size(229, 34)
    Me.Button4.TabIndex = 51
    Me.Button4.Text = "SIMPAN CIPHERIMAGE"
    Me.Button4.UseVisualStyleBackColor = False
    '
    'GroupBox5
    '
    Me.GroupBox5.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.GroupBox5.Location = New System.Drawing.Point(486, 256)
    Me.GroupBox5.Name = "GroupBox5"
    Me.GroupBox5.Size = New System.Drawing.Size(263, 100)
    Me.GroupBox5.TabIndex = 52
    Me.GroupBox5.TabStop = False
    Me.GroupBox5.Text = "Proses"
    '
    'Label10
    '
    Me.Label10.AutoSize = True
    Me.Label10.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.Label10.Location = New System.Drawing.Point(489, 407)
    Me.Label10.Name = "Label10"
    Me.Label10.Size = New System.Drawing.Size(65, 19)
    Me.Label10.TabIndex = 53
    Me.Label10.Text = "Status :-"
    '
    'detik
    '
    Me.detik.BackColor = System.Drawing.Color.White
    Me.detik.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.detik.Location = New System.Drawing.Point(717, 407)
    Me.detik.Name = "detik"
    Me.detik.Size = New System.Drawing.Size(26, 23)
    Me.detik.TabIndex = 54
    '
    'SaveFileDialog1
    '
    Me.SaveFileDialog1.Filter = "Bitmap Files|*.bmp"
    '
    'Timer2
    '
    '
    'BW2
    '

```

```

Me.BW2.WorkerReportsProgress = True
Me.BW2.WorkerSupportsCancellation = True
'
'BW4
'
Me.BW4.WorkerReportsProgress = True
Me.BW4.WorkerSupportsCancellation = True
'
'Timer1
'
'
'OpenFileDialog1
'
Me.OpenFileDialog1.FileName = "OpenFileDialog1"
Me.OpenFileDialog1.Filter = "Jpg Files|*.jpg|PNG Files|*.png|Bitmap
Files|*.bmp"
'
'BW5
'
Me.BW5.WorkerReportsProgress = True
Me.BW5.WorkerSupportsCancellation = True
'
'BW1
'
Me.BW1.WorkerReportsProgress = True
Me.BW1.WorkerSupportsCancellation = True
'
'BW3
'
Me.BW3.WorkerReportsProgress = True
Me.BW3.WorkerSupportsCancellation = True
'
'Button5
'
Me.Button5.BackColor = System.Drawing.Color.ForestGreen
Me.Button5.Font = New System.Drawing.Font("Times New Roman", 11.25!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button5.ForeColor = System.Drawing.SystemColors.ControlLightLight
Me.Button5.Location = New System.Drawing.Point(689, 448)
Me.Button5.Name = "Button5"
Me.Button5.Size = New System.Drawing.Size(74, 26)
Me.Button5.TabIndex = 56
Me.Button5.Text = "Kembali"
Me.Button5.UseVisualStyleBackColor = False
'
'GroupBox2
'
Me.GroupBox2.Controls.Add(Me.Label4)
Me.GroupBox2.Controls.Add(Me.Label3)
Me.GroupBox2.Controls.Add(Me.Label7)
Me.GroupBox2.Controls.Add(Me.Label8)
Me.GroupBox2.Controls.Add(Me.Label9)
Me.GroupBox2.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GroupBox2.Location = New System.Drawing.Point(3, 350)

```

```

Me.GroupBox2.Name = "GroupBox2"
Me.GroupBox2.Size = New System.Drawing.Size(461, 98)
Me.GroupBox2.TabIndex = 41
Me.GroupBox2.TabStop = False
Me.GroupBox2.Text = "Informasi Plainimage"
Me.GroupBox2.UseWaitCursor = True
'
'Label4
'
Me.Label4.AutoSize = True
Me.Label4.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label4.Location = New System.Drawing.Point(88, 65)
Me.Label4.Name = "Label4"
Me.Label4.Size = New System.Drawing.Size(93, 16)
Me.Label4.TabIndex = 7
Me.Label4.Text = "File Name : "
Me.Label4.UseWaitCursor = True
Me.Label4.Visible = False
'
'Label3
'
Me.Label3.AutoSize = True
Me.Label3.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label3.Location = New System.Drawing.Point(88, 39)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(93, 16)
Me.Label3.TabIndex = 6
Me.Label3.Text = "File Name : "
Me.Label3.UseWaitCursor = True
Me.Label3.Visible = False
'
'Label7
'
Me.Label7.AutoSize = True
Me.Label7.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label7.Location = New System.Drawing.Point(155, 65)
Me.Label7.Name = "Label7"
Me.Label7.Size = New System.Drawing.Size(27, 16)
Me.Label7.TabIndex = 5
Me.Label7.Text = "KB"
Me.Label7.UseWaitCursor = True
'
'Label8
'
Me.Label8.AutoSize = True
Me.Label8.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label8.Location = New System.Drawing.Point(5, 65)
Me.Label8.Name = "Label8"
Me.Label8.Size = New System.Drawing.Size(89, 19)

```

```

Me.Label8.TabIndex = 2
Me.Label8.Text = "Kapasitas : "
Me.Label8.UseWaitCursor = True
'
'Label9
'
Me.Label9.AutoSize = True
Me.Label9.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label9.Location = New System.Drawing.Point(5, 36)
Me.Label9.Name = "Label9"
Me.Label9.Size = New System.Drawing.Size(91, 19)
Me.Label9.TabIndex = 1
Me.Label9.Text = "File Name : "
Me.Label9.UseWaitCursor = True
'
'PictureBox1
'
Me.PictureBox1.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle
Me.PictureBox1.Location = New System.Drawing.Point(8, 19)
Me.PictureBox1.Name = "PictureBox1"
Me.PictureBox1.Size = New System.Drawing.Size(214, 250)
Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
Me.PictureBox1.TabIndex = 0
Me.PictureBox1.TabStop = False
'
'GroupBox1
'
Me.GroupBox1.Controls.Add(Me.PictureBox1)
Me.GroupBox1.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GroupBox1.Location = New System.Drawing.Point(2, 62)
Me.GroupBox1.Name = "GroupBox1"
Me.GroupBox1.Size = New System.Drawing.Size(229, 282)
Me.GroupBox1.TabIndex = 3
Me.GroupBox1.TabStop = False
Me.GroupBox1.Text = "Plainimage"
'
'GroupBox6
'
Me.GroupBox6.Controls.Add(Me.PictureBox2)
Me.GroupBox6.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GroupBox6.Location = New System.Drawing.Point(238, 63)
Me.GroupBox6.Name = "GroupBox6"
Me.GroupBox6.Size = New System.Drawing.Size(226, 281)
Me.GroupBox6.TabIndex = 57
Me.GroupBox6.TabStop = False
Me.GroupBox6.Text = "Cipherimage"
'
'PictureBox2
'

```

```

        Me.PictureBox2.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle
        Me.PictureBox2.Location = New System.Drawing.Point(6, 21)
        Me.PictureBox2.Name = "PictureBox2"
        Me.PictureBox2.Size = New System.Drawing.Size(211, 246)
        Me.PictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
        Me.PictureBox2.TabIndex = 0
        Me.PictureBox2.TabStop = False
    '
    'FormEnkripsi
    '
        Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
        Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
        Me.BackColor = System.Drawing.Color.WhiteSmoke
        Me.ClientSize = New System.Drawing.Size(775, 475)
        Me.Controls.Add(Me.GroupBox6)
        Me.Controls.Add(Me.GroupBox1)
        Me.Controls.Add(Me.GroupBox2)
        Me.Controls.Add(Me.Button5)
        Me.Controls.Add(Me.detik)
        Me.Controls.Add(Me.Label10)
        Me.Controls.Add(Me.Button4)
        Me.Controls.Add(Me.Button3)
        Me.Controls.Add(Me.GB1)
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.GroupBox3)
        Me.Controls.Add(Me.GroupBox4)
        Me.Controls.Add(Me.GroupBox5)
        Me.Icon = CType(resources.GetObject("$this.Icon"),
System.Drawing.Icon)
        Me.MaximizeBox = False
        Me.Name = "FormEnkripsi"
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
        Me.Text = "FORM ENKRIPSI"
        Me.GroupBox3.ResumeLayout(False)
        Me.GroupBox3.PerformLayout()
        Me.GroupBox4.ResumeLayout(False)
        Me.GroupBox4.PerformLayout()
        Me.GB1.ResumeLayout(False)
        Me.GB1.PerformLayout()
        Me.GroupBox2.ResumeLayout(False)
        Me.GroupBox2.PerformLayout()
        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox6.ResumeLayout(False)
        CType(Me.PictureBox2,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()

End Sub
Friend WithEvents Button1 As System.Windows.Forms.Button
Friend WithEvents TextBox6 As System.Windows.Forms.TextBox
Friend WithEvents GroupBox3 As System.Windows.Forms.GroupBox

```

```

Friend WithEvents TextBox4 As System.Windows.Forms.TextBox
Friend WithEvents Button2 As System.Windows.Forms.Button
Friend WithEvents GroupBox4 As System.Windows.Forms.GroupBox
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents GB1 As System.Windows.Forms.GroupBox
Friend WithEvents PB1 As System.Windows.Forms.ProgressBar
Friend WithEvents Button3 As System.Windows.Forms.Button
Friend WithEvents Button4 As System.Windows.Forms.Button
Friend WithEvents GroupBox5 As System.Windows.Forms.GroupBox
Friend WithEvents Label10 As System.Windows.Forms.Label
Friend WithEvents Label5 As System.Windows.Forms.Label
Friend WithEvents detik As System.Windows.Forms.TextBox
Friend WithEvents SaveFileDialog1 As System.Windows.Forms.SaveFileDialog
Friend WithEvents Timer2 As System.Windows.Forms.Timer
Friend WithEvents BW2 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW4 As System.ComponentModel.BackgroundWorker
Friend WithEvents Timer1 As System.Windows.Forms.Timer
Friend WithEvents OpenFileDialog1 As System.Windows.Forms.OpenFileDialog
Friend WithEvents BW5 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW1 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW3 As System.ComponentModel.BackgroundWorker
Friend WithEvents mili As System.Windows.Forms.TextBox
Friend WithEvents Button5 As System.Windows.Forms.Button
Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
Friend WithEvents Label4 As System.Windows.Forms.Label
Friend WithEvents Label3 As System.Windows.Forms.Label
Friend WithEvents Label7 As System.Windows.Forms.Label
Friend WithEvents Label8 As System.Windows.Forms.Label
Friend WithEvents Label9 As System.Windows.Forms.Label
Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
Friend WithEvents GroupBox6 As System.Windows.Forms.GroupBox
Friend WithEvents PictureBox2 As System.Windows.Forms.PictureBox

```

End Class

FORM DEKRIPSI DESIGNER

```

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class FormDekripsi
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

```

'NOTE: The following procedure is required by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.

```
<System.Diagnostics.DebuggerStepThrough()> _  
Private Sub InitializeComponent()  
    Me.components = New System.ComponentModel.Container()  
    Dim resources As System.ComponentModel.ComponentResourceManager =  
New System.ComponentModel.ComponentResourceManager(GetType(FormDekrispi))  
    Me.GroupBox3 = New System.Windows.Forms.GroupBox()  
    Me.TextBox1 = New System.Windows.Forms.TextBox()  
    Me.GroupBox1 = New System.Windows.Forms.GroupBox()  
    Me.PictureBox1 = New System.Windows.Forms.PictureBox()  
    Me.GroupBox2 = New System.Windows.Forms.GroupBox()  
    Me.Label4 = New System.Windows.Forms.Label()  
    Me.Label3 = New System.Windows.Forms.Label()  
    Me.Label7 = New System.Windows.Forms.Label()  
    Me.Label8 = New System.Windows.Forms.Label()  
    Me.Label9 = New System.Windows.Forms.Label()  
    Me.Button1 = New System.Windows.Forms.Button()  
    Me.GroupBox4 = New System.Windows.Forms.GroupBox()  
    Me.TextBox4 = New System.Windows.Forms.TextBox()  
    Me.Button3 = New System.Windows.Forms.Button()  
    Me.Label1 = New System.Windows.Forms.Label()  
    Me.GB1 = New System.Windows.Forms.GroupBox()  
    Me.PB1 = New System.Windows.Forms.ProgressBar()  
    Me.Button4 = New System.Windows.Forms.Button()  
    Me.Button5 = New System.Windows.Forms.Button()  
    Me.GroupBox5 = New System.Windows.Forms.GroupBox()  
    Me.Label12 = New System.Windows.Forms.Label()  
    Me.detik = New System.Windows.Forms.TextBox()  
    Me.mili = New System.Windows.Forms.TextBox()  
    Me.Timer1 = New System.Windows.Forms.Timer(Me.components)  
    Me.SaveFileDialog1 = New System.Windows.Forms.SaveFileDialog()  
    Me.BW2 = New System.ComponentModel.BackgroundWorker()  
    Me.BW1 = New System.ComponentModel.BackgroundWorker()  
    Me.BW3 = New System.ComponentModel.BackgroundWorker()  
    Me.BW5 = New System.ComponentModel.BackgroundWorker()  
    Me.BW4 = New System.ComponentModel.BackgroundWorker()  
    Me.Timer2 = New System.Windows.Forms.Timer(Me.components)  
    Me.Button6 = New System.Windows.Forms.Button()  
    Me.GroupBox6 = New System.Windows.Forms.GroupBox()  
    Me.PictureBox2 = New System.Windows.Forms.PictureBox()  
    Me.ToolTip1 = New System.Windows.Forms.ToolTip(Me.components)  
    Me.OpenFileDialog1 = New System.Windows.Forms.OpenFileDialog()  
    Me.GroupBox3.SuspendLayout()  
    Me.GroupBox1.SuspendLayout()  
    CType(Me.PictureBox1,  
System.ComponentModel.ISupportInitialize).BeginInit()  
    Me.GroupBox2.SuspendLayout()  
    Me.GroupBox4.SuspendLayout()  
    Me.GB1.SuspendLayout()  
    Me.GroupBox6.SuspendLayout()  
    CType(Me.PictureBox2,  
System.ComponentModel.ISupportInitialize).BeginInit()  
    Me.SuspendLayout()  
    '  
    'GroupBox3
```



```

    '
    Me.GroupBox3.Controls.Add(Me.TextBox1)
    Me.GroupBox3.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.GroupBox3.Location = New System.Drawing.Point(6, 9)
    Me.GroupBox3.Name = "GroupBox3"
    Me.GroupBox3.Size = New System.Drawing.Size(461, 44)
    Me.GroupBox3.TabIndex = 45
    Me.GroupBox3.TabStop = False
    Me.GroupBox3.Text = "Sumber File"
    '
    'TextBox1
    '
    Me.TextBox1.BackColor = System.Drawing.Color.White
    Me.TextBox1.Location = New System.Drawing.Point(6, 16)
    Me.TextBox1.Name = "TextBox1"
    Me.TextBox1.Size = New System.Drawing.Size(443, 23)
    Me.TextBox1.TabIndex = 43
    '
    'GroupBox1
    '
    Me.GroupBox1.Controls.Add(Me.PictureBox1)
    Me.GroupBox1.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.GroupBox1.Location = New System.Drawing.Point(6, 55)
    Me.GroupBox1.Name = "GroupBox1"
    Me.GroupBox1.Size = New System.Drawing.Size(221, 290)
    Me.GroupBox1.TabIndex = 46
    Me.GroupBox1.TabStop = False
    Me.GroupBox1.Text = "Cipherimage"
    '
    'PictureBox1
    '
    Me.PictureBox1.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle
    Me.PictureBox1.Location = New System.Drawing.Point(8, 19)
    Me.PictureBox1.Name = "PictureBox1"
    Me.PictureBox1.Size = New System.Drawing.Size(206, 263)
    Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
    Me.PictureBox1.TabIndex = 0
    Me.PictureBox1.TabStop = False
    '
    'GroupBox2
    '
    Me.GroupBox2.Controls.Add(Me.Label4)
    Me.GroupBox2.Controls.Add(Me.Label3)
    Me.GroupBox2.Controls.Add(Me.Label7)
    Me.GroupBox2.Controls.Add(Me.Label8)
    Me.GroupBox2.Controls.Add(Me.Label9)
    Me.GroupBox2.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.GroupBox2.Location = New System.Drawing.Point(10, 351)
    Me.GroupBox2.Name = "GroupBox2"

```

```

Me.GroupBox2.Size = New System.Drawing.Size(457, 93)
Me.GroupBox2.TabIndex = 41
Me.GroupBox2.TabStop = False
Me.GroupBox2.Text = "Informasi Cipherimage"
Me.GroupBox2.UseWaitCursor = True
'
'Label14
'
Me.Label14.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label14.Location = New System.Drawing.Point(92, 59)
Me.Label14.Name = "Label14"
Me.Label14.Size = New System.Drawing.Size(93, 16)
Me.Label14.TabIndex = 7
Me.Label14.Text = "File Name : "
Me.Label14.UseWaitCursor = True
Me.Label14.Visible = False
'
'Label13
'
Me.Label13.AutoSize = True
Me.Label13.Font = New System.Drawing.Font("Verdana", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label13.Location = New System.Drawing.Point(92, 30)
Me.Label13.Name = "Label13"
Me.Label13.Size = New System.Drawing.Size(93, 16)
Me.Label13.TabIndex = 6
Me.Label13.Text = "File Name : "
Me.Label13.UseWaitCursor = True
Me.Label13.Visible = False
'
'Label17
'
Me.Label17.AutoSize = True
Me.Label17.Location = New System.Drawing.Point(134, 58)
Me.Label17.Name = "Label17"
Me.Label17.Size = New System.Drawing.Size(24, 18)
Me.Label17.TabIndex = 5
Me.Label17.Text = "KB"
Me.Label17.UseWaitCursor = True
'
'Label18
'
Me.Label18.AutoSize = True
Me.Label18.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Label18.Location = New System.Drawing.Point(5, 58)
Me.Label18.Name = "Label18"
Me.Label18.Size = New System.Drawing.Size(89, 19)
Me.Label18.TabIndex = 2
Me.Label18.Text = "Kapasitas : "
Me.Label18.UseWaitCursor = True
'
'Label19

```

```

    '
    Me.Label9.AutoSize = True
    Me.Label9.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.Label9.Location = New System.Drawing.Point(5, 29)
    Me.Label9.Name = "Label9"
    Me.Label9.Size = New System.Drawing.Size(91, 19)
    Me.Label9.TabIndex = 1
    Me.Label9.Text = "File Name : "
    Me.Label9.UseWaitCursor = True
    '
    'Button1
    '
    Me.Button1.BackColor = System.Drawing.Color.DarkGreen
    Me.Button1.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.Button1.ForeColor = System.Drawing.SystemColors.ButtonHighlight
    Me.Button1.Location = New System.Drawing.Point(491, 15)
    Me.Button1.Name = "Button1"
    Me.Button1.Size = New System.Drawing.Size(263, 34)
    Me.Button1.TabIndex = 47
    Me.Button1.Text = "OPEN CIPHERIMAGE"
    Me.Button1.UseVisualStyleBackColor = False
    '
    'GroupBox4
    '
    Me.GroupBox4.Controls.Add(Me.TextBox4)
    Me.GroupBox4.Controls.Add(Me.Button3)
    Me.GroupBox4.Controls.Add(Me.Label1)
    Me.GroupBox4.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.GroupBox4.Location = New System.Drawing.Point(491, 95)
    Me.GroupBox4.Name = "GroupBox4"
    Me.GroupBox4.Size = New System.Drawing.Size(263, 134)
    Me.GroupBox4.TabIndex = 49
    Me.GroupBox4.TabStop = False
    Me.GroupBox4.Text = "Infomasi"
    '
    'TextBox4
    '
    Me.TextBox4.BackColor = System.Drawing.Color.White
    Me.TextBox4.Location = New System.Drawing.Point(14, 37)
    Me.TextBox4.Name = "TextBox4"
    Me.TextBox4.Size = New System.Drawing.Size(231, 23)
    Me.TextBox4.TabIndex = 49
    '
    'Button3
    '
    Me.Button3.BackColor = System.Drawing.Color.DarkGreen
    Me.Button3.Font = New System.Drawing.Font("Times New Roman", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
    Me.Button3.ForeColor = System.Drawing.SystemColors.ButtonHighlight
    Me.Button3.Location = New System.Drawing.Point(16, 77)

```

```

Me.Button3.Name = "Button3"
Me.Button3.Size = New System.Drawing.Size(229, 34)
Me.Button3.TabIndex = 47
Me.Button3.Text = "Reset Kunci"
Me.Button3.UseVisualStyleBackColor = False
'
'Label1
'
Me.Label1.AutoSize = True
Me.Label1.Location = New System.Drawing.Point(13, 21)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(97, 18)
Me.Label1.TabIndex = 0
Me.Label1.Text = "Kunci Dekripsi"
'
'GB1
'
Me.GB1.Controls.Add(Me.PB1)
Me.GB1.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GB1.Location = New System.Drawing.Point(492, 351)
Me.GB1.Name = "GB1"
Me.GB1.Size = New System.Drawing.Size(264, 46)
Me.GB1.TabIndex = 50
Me.GB1.TabStop = False
Me.GB1.Text = "Progress"
'
'PB1
'
Me.PB1.Location = New System.Drawing.Point(6, 18)
Me.PB1.Name = "PB1"
Me.PB1.Size = New System.Drawing.Size(252, 22)
Me.PB1.TabIndex = 0
'
'Button4
'
Me.Button4.BackColor = System.Drawing.Color.DarkGreen
Me.Button4.Font = New System.Drawing.Font("Cambria", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button4.ForeColor = System.Drawing.SystemColors.ButtonHighlight
Me.Button4.Location = New System.Drawing.Point(508, 303)
Me.Button4.Name = "Button4"
Me.Button4.Size = New System.Drawing.Size(229, 34)
Me.Button4.TabIndex = 54
Me.Button4.Text = "SIMPAN PLAINIMAGE"
Me.Button4.UseVisualStyleBackColor = False
'
'Button5
'
Me.Button5.BackColor = System.Drawing.Color.DarkGreen
Me.Button5.Font = New System.Drawing.Font("Cambria", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button5.ForeColor = System.Drawing.SystemColors.ButtonHighlight
Me.Button5.Location = New System.Drawing.Point(508, 263)

```

```

Me.Button5.Name = "Button5"
Me.Button5.Size = New System.Drawing.Size(229, 34)
Me.Button5.TabIndex = 53
Me.Button5.Text = "DEKRIPSI CIPHERIMAGE"
Me.Button5.UseVisualStyleBackColor = False
'
'GroupBox5
'
Me.GroupBox5.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GroupBox5.Location = New System.Drawing.Point(492, 245)
Me.GroupBox5.Name = "GroupBox5"
Me.GroupBox5.Size = New System.Drawing.Size(263, 100)
Me.GroupBox5.TabIndex = 55
Me.GroupBox5.TabStop = False
Me.GroupBox5.Text = "Proses"
'
'Label12
'
Me.Label12.AutoSize = True
Me.Label12.Location = New System.Drawing.Point(651, 409)
Me.Label12.Name = "Label12"
Me.Label12.Size = New System.Drawing.Size(73, 13)
Me.Label12.TabIndex = 57
Me.Label12.Text = "Timer/second"
'
'detik
'
Me.detik.Location = New System.Drawing.Point(730, 406)
Me.detik.Name = "detik"
Me.detik.Size = New System.Drawing.Size(26, 20)
Me.detik.TabIndex = 56
'
'mili
'
Me.mili.BackColor = System.Drawing.Color.White
Me.mili.Location = New System.Drawing.Point(730, 406)
Me.mili.Name = "mili"
Me.mili.Size = New System.Drawing.Size(26, 20)
Me.mili.TabIndex = 58
'
'Timer1
'
'SaveFileDialog1
'
Me.SaveFileDialog1.Filter = "Bitmap Files|.bmp"
'
'BW2
'
Me.BW2.WorkerReportsProgress = True
Me.BW2.WorkerSupportsCancellation = True
'
'BW1
'
Me.BW1.WorkerReportsProgress = True

```

```

Me.BW1.WorkerSupportsCancellation = True
'
'BW3
'
Me.BW3.WorkerReportsProgress = True
Me.BW3.WorkerSupportsCancellation = True
'
'BW5
'
Me.BW5.WorkerReportsProgress = True
Me.BW5.WorkerSupportsCancellation = True
'
'BW4
'
Me.BW4.WorkerReportsProgress = True
Me.BW4.WorkerSupportsCancellation = True
'
'Timer2
'
'Button6
'
Me.Button6.BackColor = System.Drawing.Color.ForestGreen
Me.Button6.Font = New System.Drawing.Font("Times New Roman", 11.25!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.Button6.ForeColor = System.Drawing.Color.White
Me.Button6.Location = New System.Drawing.Point(681, 442)
Me.Button6.Name = "Button6"
Me.Button6.Size = New System.Drawing.Size(74, 26)
Me.Button6.TabIndex = 59
Me.Button6.Text = "Kembali"
Me.Button6.UseVisualStyleBackColor = False
'
'GroupBox6
'
Me.GroupBox6.Controls.Add(Me.PictureBox2)
Me.GroupBox6.Font = New System.Drawing.Font("Trebuchet MS", 9.75!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
Me.GroupBox6.Location = New System.Drawing.Point(233, 54)
Me.GroupBox6.Name = "GroupBox6"
Me.GroupBox6.Size = New System.Drawing.Size(233, 290)
Me.GroupBox6.TabIndex = 60
Me.GroupBox6.TabStop = False
Me.GroupBox6.Text = "Plainimage"
'
'PictureBox2
'
Me.PictureBox2.BorderStyle =
System.Windows.Forms.BorderStyle.FixedSingle
Me.PictureBox2.Location = New System.Drawing.Point(6, 20)
Me.PictureBox2.Name = "PictureBox2"
Me.PictureBox2.Size = New System.Drawing.Size(221, 263)
Me.PictureBox2.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
Me.PictureBox2.TabIndex = 0

```

```

Me.PictureBox2.TabStop = False
'
'OpenFileDialog1
Me.OpenFileDialog1.FileName = "OpenFileDialog1"
'FormDekrispi
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.BackColor = System.Drawing.Color.WhiteSmoke
Me.ClientSize = New System.Drawing.Size(775, 475)
Me.Controls.Add(Me.GroupBox6)
Me.Controls.Add(Me.Button6)
Me.Controls.Add(Me.GroupBox2)
Me.Controls.Add(Me.mili)
Me.Controls.Add(Me.Label12)
Me.Controls.Add(Me.detik)
Me.Controls.Add(Me.Button4)
Me.Controls.Add(Me.Button5)
Me.Controls.Add(Me.GroupBox5)
Me.Controls.Add(Me.GB1)
Me.Controls.Add(Me.GroupBox4)
Me.Controls.Add(Me.Button1)
Me.Controls.Add(Me.GroupBox3)
Me.Controls.Add(Me.GroupBox1)
Me.Icon = CType(resources.GetObject("$this.Icon"),
System.Drawing.Icon)
Me.MaximizeBox = False
Me.Name = "FormDekrispi"
Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
Me.Text = "FORM DEKRIPSI"
Me.GroupBox3.ResumeLayout(False)
Me.GroupBox3.PerformLayout()
Me.GroupBox1.ResumeLayout(False)
CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).EndInit()
Me.GroupBox2.ResumeLayout(False)
Me.GroupBox2.PerformLayout()
Me.GroupBox4.ResumeLayout(False)
Me.GroupBox4.PerformLayout()
Me.GB1.ResumeLayout(False)
Me.GroupBox6.ResumeLayout(False)
CType(Me.PictureBox2,
System.ComponentModel.ISupportInitialize).EndInit()
Me.ResumeLayout(False)
Me.PerformLayout()

End Sub
Friend WithEvents GroupBox3 As System.Windows.Forms.GroupBox
Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox
Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
Friend WithEvents Label14 As System.Windows.Forms.Label
Friend WithEvents Label13 As System.Windows.Forms.Label

```

```
Friend WithEvents Label17 As System.Windows.Forms.Label
Friend WithEvents Label18 As System.Windows.Forms.Label
Friend WithEvents Label19 As System.Windows.Forms.Label
Friend WithEvents Button1 As System.Windows.Forms.Button
Friend WithEvents GroupBox4 As System.Windows.Forms.GroupBox
Friend WithEvents TextBox4 As System.Windows.Forms.TextBox
Friend WithEvents Button3 As System.Windows.Forms.Button
Friend WithEvents Label1 As System.Windows.Forms.Label
Friend WithEvents GB1 As System.Windows.Forms.GroupBox
Friend WithEvents PB1 As System.Windows.Forms.ProgressBar
Friend WithEvents Button4 As System.Windows.Forms.Button
Friend WithEvents Button5 As System.Windows.Forms.Button
Friend WithEvents GroupBox5 As System.Windows.Forms.GroupBox
Friend WithEvents Label12 As System.Windows.Forms.Label
Friend WithEvents detik As System.Windows.Forms.TextBox
Friend WithEvents mili As System.Windows.Forms.TextBox
Friend WithEvents Timer1 As System.Windows.Forms.Timer
Friend WithEvents SaveFileDialog1 As System.Windows.Forms.SaveFileDialog
Friend WithEvents BW2 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW1 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW3 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW5 As System.ComponentModel.BackgroundWorker
Friend WithEvents BW4 As System.ComponentModel.BackgroundWorker
Friend WithEvents Timer2 As System.Windows.Forms.Timer
Friend WithEvents Button6 As System.Windows.Forms.Button
Friend WithEvents GroupBox6 As System.Windows.Forms.GroupBox
Friend WithEvents PictureBox2 As System.Windows.Forms.PictureBox
Friend WithEvents ToolTip1 As System.Windows.Forms.ToolTip
Friend WithEvents OpenFileDialog1 As System.Windows.Forms.OpenFileDialog
End Class
```


LAMPIRAN 2
DAFTAR RIWAYAT HIDUP



I. Data Pribadi

Nama : Imam Zaki Husein Nasution
Tempat/ Tanggal Lahir : Pancur Batu, 27 Juni 1995
Alamat : Jl. Pengabdian Gg. Nazib No.450 Bandar Setia
Agama : Islam
Jenis Kelamin : Laki-Laki
Status : Sudah Menikah
No. Hp : 085206241739
Email : zakynst1995@gmail.com

II. Pendidikan Formal

1. SD MIN Medan Tembung tamatan tahun 2007
2. SMP Pondok Pesantren Ar-Raudhatul Hasanah tamatan tahun 2010
3. SMA Pondok Pesantren Mawaridussalam tamatan tahun 2013
4. S1 Universitas Islam Negeri Sumatera Utara Medan Jurusan Ilmu Komputer 2015 – 2020

LAMPIRAN 3

KARTU BIMBINGAN SRKIPSI

Buku Laporan Kegiatan Akademik Mahasiswa Fakultas SAINTEK UIN SU Medan

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik /


Nama : Imam Zaki Husain	Pembimbing I : Dr. Mhd. Furqan, S.Si, M.Com, B.Sc
NIM : 71153010	Pembimbing II : Rakhmat, Kurnawati R, ST, M. Kom
Prog. Studi : Ilmu komputer	SK Pembimbing :
Judul Skripsi : sistem keamanan citra Digital Menggunakan Algoritma S.Pitz	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	28/10/2019	Bimbingan dan Revisi Judul		28/10/2019	Bimbingan dan Revisi Judul	
II	08/11/2019	Revisi Bab I Latar Belakang Batasan masalah		08/11/2020	Revisi Bab I Latar Belakang Batasan Masalah	
III	20/11/2019	Revisi BAB I Rumusan Masalah		20/11/2019	REVISI BAB I Rumusan Masalah	
IV	13/12/2019	Acc BAB I Lanjut Bab II		13/12/2019	Acc' Bab I Lanjut Bab II	
V	18/12/2019	Acc Bab II Lanjut Bab III		20/12/2019	Acc Bab II Lanjut Bab III	

Buku Laporan Kegiatan Akademik Mahasiswa Fakultas SAINTEK UIN SU Medan

VI	14/01/2020	Acc BAB III dan Seminar Proposal		14/01/2020	ACC BAB III dan seminar Proposal	
VII	21/01/2020	Revisi Proposal		21/01/2020	Revisi Proposal	
VIII	18/06/2020	Revisi Bab IV		18/06/2020	Revisi Bab IV	
IX	25/06/2020	Revisi BAB IV dan V		25/06/2020	Revisi BAB IV dan V	
X	26/06/2020	Acc Skripsi		26/06/2020	ACC Skripsi	

Medan, 20
An. Dekan
Ketua Jurusan/Program Studi


Dr. Mhd. Furqan, S.Si., M.com.P. Sc.
NIP. 198008062006041003

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing