IMPLEMENTASI PENYEMBUNYIAN PESAN PADA CITRA DIGITAL DENGAN MENGGABUNGKAN ALGORITMA *HILL CIPHER* DAN METODE *LEAST SIGNIFICANT BIT* (*LSB*)

SKRIPSI

oleh

Ramma Eka Putera

E41182130

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2022

IMPLEMENTASI PENYEMBUNYIAN PESAN PADA CITRA DIGITAL DENGAN MENGGABUNGKAN ALGORITMA *HILL CIPHER* DAN METODE *LEAST SIGNIFICANT BIT* (*LSB*)

SKRIPSI

sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Tr.Kom.)
di Program Studi Teknik Informatika Jurusan Teknologi Informasi

oleh

Ramma Eka Putera

E41182130

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2022

KEMENTERIAN PENDIDIKAN KEBUDAYAAN RISET DAN TEKNOLOGI
POLITEKNIK NEGERI JEMBER
JURUSAN TEKNOLOGI INFORMASI

HALAMAN PENGESAHAN SKRIPSI

Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* Dan Metode *Least Significant Bit* (*LSB*)

Ramma Eka Putera (NIM E41182130**)**

Telah diuji pada tanggal 7 Juni 2022 dan dinyatakan memenuhi syarat.

Ketua Penguji

Mukhamad Angga Gumilang, S. Pd., M. Eng
NIP.199408122019031013

Sekretariat Penguji,

I Gede Wiryawan, S.Kom., M.Kom.
NIP. 19880117201901008

Anggota Penguji,

Arvita Agus Kurniasari, S.ST.,M.Tr.Kom
NIP. 199308312021032001

Dosen Pembimbing,

I Gede Wiryawan, S.Kom., M.Kom.
NIP. 19880117 20190 1 008

Ketua Jurusan,

Hendra Yufit Riskiawan, S.Kom, M.Cs
NIP. 19830203 200604 1 003

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Ramma Eka Putera

NIM : E41182130

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Laporan Skripsi saya berjudul "Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* dan Metode *Least Significant Bit* (*LSB*)" merupakan gagasan dan hasil karya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam daftar pustaka di bagian akhir Laporan Skripsi ini.

Jember, 07 Juni 2022

Ramma Eka Putera

NIM E41182130

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Yang bertanda tangan di bawah ini, saya:

Nama : Ramma Eka Putera

NIM : E41182130

Program Studi : Teknik Informatika

Jurusan : Teknologi Informasi

Demi pengembangan Ilmu Pengetahuan, saya menyetujui untuk memberikan kepada UPT.Perpustakaan Politeknik Negeri Jember, Hak Bebas Royalti Non-Eksklusif *(Non-Exclusive Royality Free Right)* atas Karya Ilmiah **berupa Laporan Skripsi saya yang berjudul :**

Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* Dan Metode *Least Significant Bit* (*LSB*)

Dengan Hak Bebas Royalti Non-Eksklusif ini UPT. Perpustakaan Politeknik Negeri Jember berhak menyimpan, mengalih media atau format, mengelola dalam bentuk Pangkalan Data (*Database*), mendistribusikan karya dan menampilkan atau mempublikasikannya di Internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis atau pencipta.

Saya bersedia untuk menanggung secara pribadi tanpa melibatkan pihak Politeknik Negeri Jember, Segala bentuk tuntutan hukum yang timbul atas Pelanggaran Hak Cipta dalam Karya ilmiah ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat : Jember

Pada Tanggal : 17 Mei 2022

Yang Menyatakan,

Nama : Ramma Eka Putera

NIM : E41182130

HALAMAN MOTTO

“Barang siapa bertakwa kepada Allah maka Dia akan menjadikan jalan keluar baginya, dan memberinya rezeki dari jalan yang tidak ia sangka, dan barang siapa yang bertawakal kepada Allah maka cukuplah Allah baginya, Sesungguhnya Allah melaksanakan kehendak-Nya, Dia telah menjadikan untuk setiap sesuatu kadarnya.”

(*Q.S. Ath-Thalaq* ayat 2-3)

“Allah akan meninggikan orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat.”

(*Q.S. Al-Mujadalah* ayat 11)

"Bersemangatlah atas hal-hal yang bermanfaat bagimu. Minta tolonglah pada Allah, jangan engkau lemah." - HR. Muslim

"Waktu bagaikan pedang. Jika engkau tidak memanfaatkannya dengan baik (untuk memotong), maka ia akan memanfaatkanmu (dipotong)." - HR. Muslim

“Sebaik-baik manusia adalah yang bermanfaat bagi manusia yang lain” – HR. al-Tabrani dalam Mu’jam al-Awsathnya

PERSEMBAHAN

Karya Tulis Ilmiah ini saya persembahkan kepada :

1. Orang tua tercinta. Terima kasih atas segala bimbingannya dan dukungannya, serta doa yang tak pernah berhenti untuk saya dalam menempuh pendidikan selama ini. Terima kasih yang tak terhingga atas semuanya hingga menjadikanku sampai saat ini.
2. Bapak I Gede Wiryawan, S.Kom., M.Kom, selaku pembimbing saya yang banyak memberi bimbingan, arahan dan motivasi kepada penulis selama proses pengerjaan skripsi.
3. Bapak Ibu Dosen Jurusan Teknologi Informasi dan seluruh civitas akademik Politeknik Negeri Jember yang telah memberikan ilmu dan pengetahuan serta pengalaman
4. Teman-teman seperjuangan TIF Angkatan 2018 yang tidak bisa disebutkan satu persatu, terima kasih telah memberi saran dan masukkan serta pengalaman berharga selama empat tahun ini.
5. Semua orang yang telah membantu saya dikala sedih, senang maupun susah. Semuanya saya ucapkan terima kasih sebesar-besarnya.

IMPLEMENTASI PENYEMBUNYIAN PESAN PADA CITRA DIGITAL DENGAN MENGGABUNGKAN ALGORITMA *HILL CIPHER* DAN METODE *LEAST SIGNIFICANT BIT* (*LSB*), *(IMPLEMENTATION OF MESSAGE HIDING IN DIGITAL IMAGES BY COMBINED HILL CIPHER ALGORITHM AND LEAST SIGNIFICANT BIT (LSB) METHODS*)

Pembimbing (1 orang)

Informatics Engineering Study Program

Department of Information Technology

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

ABSTRAK

Pesatnya perkembangan teknologi di era saat ini, memudahkan masyarakat bertukar informasi dalam media digital seperti teks, audio, video, dan citra. Perkembangan Informasi dan Komunikasi menjadikan kegiatan penyampaian informasi maupun data menjadi lebih efisien. Perkembangan teknologi saat ini yang sangat signifikan memberikan dampak bagi masyarakat dalam bertukar informasi maupun melakukan komunikasi. Secara umum informasi dikategorikan menjadi dua, yaitu informasi yang bersifat rahasia dan informasi yang tidak bersifat rahasia. Informasi bersifat rahasia yaitu setiap informasi yang ada didalamnya sangat berharga bagi pihak yang membutuhkan karena informasi tersebut dapat dengan mudah digandakan. Banyak cara yang dapat dilakukan untuk penyembunyian. Salah satu cara penyembunyian pesan dalam pengiriman yaitu dengan penyandian dan penyisipan menggunakan teknik kriptografi dan steganografi. Maka dari itu penulis menggabungkan metode kriptografi yaitu algoritma *Hill Cipher* dan metode steganografi yaitu Least Significant Bit (LSB).

Kata kunci: Steganografi, Kriptografi, LSB, Hill Cipher

IMPLEMENTASI PENYEMBUNYIAN PESAN PADA CITRA DIGITAL DENGAN MENGGABUNGKAN ALGORITMA *HILL CIPHER* DAN METODE *LEAST SIGNIFICANT BIT* (*LSB*), *(IMPLEMENTATION OF MESSAGE HIDING IN DIGITAL IMAGES BY COMBINED HILL CIPHER ALGORITHM AND LEAST SIGNIFICANT BIT (LSB) METHODS*)

Pembimbing (1 orang)

Informatics Engineering Study Program

Department of Information Technology

Program Studi Teknik Informatika

Jurusan Teknologi Informasi

ABSTRACT

The rapid development of technology in the current era makes it easier for people to exchange information in digital media such as text, audio, video, and images. The development of Information and Communication has made the delivery of information and data more efficient. The current technological developments which are very significant have an impact on the community in exchanging information and communicating. In general, information is categorized into two, namely confidential information and non-confidential information. Information is confidential, that is, any information contained in it is very valuable for those who need it because the information can be easily duplicated. There are many ways to hide. One way to hide messages in delivery is by encoding and embedding using cryptography and steganography techniques. Therefore, the author combines the cryptographic method, namely the Hill Cipher algorithm and the steganographic method, namely the Least Significant Bit (LSB).

Keywords: Steganography, Cryptography, LSB, Hill Cipher

RINGKASAN

Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* dan Metode *Least Significant Bit* (*LSB*). Ramma Eka Putera, NIM E41182130, Tahun 2022, Teknologi Informasi, Politeknik Negeri Jember, I Gede Wiryawan, S.Kom, M.Kom (Dosen Pembimbing).

Pesatnya perkembangan teknologi di era saat ini, memudahkan masyarakat bertukar informasi dalam media digital seperti teks, audio, video, dan citra. Perkembangan Informasi dan Komunikasi menjadikan kegiatan penyampaian informasi maupun data menjadi lebih efisien. Perkembangan teknologi saat ini yang sangat signifikan memberikan dampak bagi masyarakat dalam bertukar informasi maupun melakukan komunikasi. Secara umum informasi dikategorikan menjadi dua, yaitu informasi yang bersifat rahasia dan informasi yang tidak bersifat rahasia. Informasi yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Informasi bersifat rahasia yaitu setiap informasi yang ada didalamnya sangat berharga bagi pihak yang membutuhkan karena informasi tersebut dapat dengan mudah digandakan. Saat ini telah banyak cara yang dapat dilakukan untuk penyembunyian pesan dalam pengiriman data dengan merubah data menjadi yang tidak dimengerti oleh pihak yang tidak memiliki akses untuk menerima pesan tersebut. Salah satu cara penyembunyian pesan dalam pengiriman yaitu dengan penyandian dan penyisipan menggunakan teknik kriptografi dan steganografi. Maka dari itu penulis menggabungkan metode kriptografi yaitu algoritma *Hill Cipher* dan metode steganografi yaitu *Least Significant Bit* (*LSB*).

PRAKATA

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas berkat, rahmat dan karunia-Nya sehingga skripsi dengan judul "Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* Dan Metode *Least Significant Bit* (*LSB*)" dapat terselesaikan pada tepat pada waktunya dengan baik, serta kami ucapkan terimakasih kepada semua pihak yang telah mendukung semua proses penelitian ini sehingga dapat terselesaikannya penulisan skripsi ini.

Penulis menyampaikan penghargaan dan ucapan terimakasih yang sebesar besarnya kepada:

1. Saiful Anwar, S.TP, MP, selaku Direktur Politeknik Negeri Jember,
2. Bapak Hendra Yufit Riskiawan, S.Kom, M.Cs, selaku Ketua Jurusan Teknologi Informasi,
3. Trismayanti Dwi P, S.Kom, M.Cs selaku Ketua Prodi Teknik Informatika,
4. I Gede Wiryawan, S.Kom, M.Kom, selaku dosen pembimbing yang mengarahkan dengan sangat baik selama proses penelitian,
5. Keluarga, rekan-rekan TIF angkatan 2018, dan semua pihak yang telah membantu memberikan dukungan, bimbingan, kritik dan saran dalam proses penelitian dan penyelesaian skripsi.

Penulisan skripsi ini masih kurang dari kata sempurna. Penulis juga mengharapkan saran dan kritik yang membangun agar menjadi lebih baik lagi guna perbaikan dan pengembangan di masa mendatang. Semoga laporan ini bermanfaat bagi para pembaca. Jurusan Teknologi Informasi

Jember, 7 Juni 2022

Penulis

DAFTAR ISI

Halaman

HALAMAN PENGESAHAN SKRIPSI iii

SURAT PERNYATAAN iv

HALAMAN MOTTO vi

PERSEMBAHAN vii

ABSTRAK viii

ABSTRACT ix

RINGKASAN x

PRAKATA xi

DAFTAR ISI xii

DAFTAR GAMBAR xvi

DAFTAR TABEL xix

BAB 1. PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 4

1.3 Tujuan 4

1.4 Manfaat 5

1.5 Batasan Masalah 5

BAB 2. TINJAUAN PUSTAKA 7

2.1 Kriptografi 7

2.2 Algoritma Hill *Cipher* 8

2.3 Steganografi 9

2.4 Citra Digital (Bitmap) 9

2.5 Metode Least Significant Bit (LSB) 11

2.6 Aplikasi 14
2.7 Website 14
2.8 XAMPP 14
2.9 Basis Data 15
2.10 MySQL 15
2.11 PHP 16
2.12 Framework CodeIgniter 17
2.13 Pengembangan Sistem 17
2.13.1 Unified Modeling Language 17
2.13.2 Diagram *Use Case* 18
2.13.3 Data Flow Diagram (DFD) 20
2.13.4 Flowchart 21
2.13.5 Entity-Relationship Diagram (ERD) 22
2.14 Pengujian Perangkat Lunak 23
2.14.1 Black Box Testing 23
2.14.2 *Peak Signal-to-Noise Ratio* (PSNR) 24
2.14.3 *Mean Square Error* (MSE) 24
2.15 State of The Art 24
BAB 3. METODE PENELITIAN 27
3.1 Waktu dan Tempat 27
3.1.1 Tempat Pelaksanaan 27
3.1.2 Waktu Pelaksanaan 27
3.2 Alat dan Bahan 27
3.2.1 Alat 27
3.2.2 Bahan 28

3.3 Tahapan Penelitian 28
3.3.1 Identifikasi Masalah 29
3.3.2 Studi Literatur 29
3.3.3 Pengumpulan Data 29
3.3.4 Analisa 29
3.3.5 Perancangan 29
3.3.6 Pengujian 33
3.3.7 Pembuatan Laporan 34
3.4 Jadwal Kegiatan 34
BAB 4. HASIL DAN PEMBAHASAN 36
4.1 Analisa Perangkat Lunak 36
4.1.1 Deskripsi Aplikasi 36
4.1.2 Alur Penggunaan Aplikasi 37
4.1.3 Kebutuhan Data 37
4.1.4 Kelebihan Aplikasi 37
4.1.5 Kekurangan Aplikasi 37
4.1.6 Tujuan Pengembangan Aplikasi 38
4.2 Perancangan Sistem 38
4.2.1 *Algoritma Hill Cipher* 38
4.2.2 Metode Least Significant Bit 39
4.2.3 Perancangan Flowchart 39
4.3 Perhitungan Manual 43
4.3.1 Perhitungan Manual Untuk Proses Encode 43
4.3.2 Perhitungan Manual Untuk Proses Decode 45
4.4 Implementasi 47

4.4.1 Interface 47
4.5 Pengujian 57
4.5.1 Pengujian Menu Login 57
4.5.2 Pengujian Menu Registrasi 60
4.5.3 Pengujian Menu Encode 64
4.5.4 Pengujian Menu Decode 70
4.5.5 Pengujian Menu Informasi dan Test 74
4.5.6 Pengujian Pengiriman Gambar 75
BAB 5 KESIMPULAN DAN SARAN 80
5.1 Kesimpulan 80
5.2 Saran 81
DAFTAR PUSTAKA 82
LAMPIRAN 84

DAFTAR GAMBAR

Halaman

Gambar 2. 1 Proses enkripsi dan dekripsi 7
Gambar 2. 2 Proses Penyimpanan data rahasia ke dalam media digital dengan teknik steganografi 9
Gambar 2. 3 Citra grayscale 8 bit 10x10 pikselCitra grayscale 8 bit 10x10 piksel 12
Gambar 2. 4 Citra grayscale 8 piksel yang diambil Langkah kedua adalah mengganti bit terakhir (LSB) dari piksel citra dengan bit-bit dari huruf A. 12
Gambar 2. 5 Grayscale piksel yang berubah 13
Gambar 3. 1 Tahapan Penelitian 28
Gambar 3. 2 DFD level0 30
Gambar 3. 3 *Flowchart* Sistem 30
Gambar 3. 4 Halaman *Login* 31
Gambar 3. 5 Halaman registrasi 32
Gambar 3. 6 Halaman Embed 32
Gambar 3. 7 Halaman Ekstrak 33
Gambar 4. 1 *Flowchart* Pengacakan Pesan 39
Gambar 4. 2 *Flowchart* Penyisipan Pesan 40
Gambar 4. 3 *Flowchart* Ekstraksi Pesan Pada Gambar 41
Gambar 4. 4 *Flowchart* Ekstraksi Pesan *Hill Cipher* 42
Gambar 4. 5 Tabel Angka *Cipher* yang Digunakan Untuk Pengganti Huruf 43
Gambar 4. 6 *Plaintext* dan Matriks Kunci 43
Gambar 4. 7 Pengelompokkan *Plaintext* Untuk Perkalian matriks 2x2 44
Gambar 4. 8 Mengganti Huruf Menjadi Angka Berdasarkan Tabel *Cipher* 44
Gambar 4. 9 Perhitungan Dilakukan Agar Menemukan Hasil Dari *Ciphertext* ... 44
Gambar 4. 10 Hasil Dari *Ciphertext* yang Sudah Didapat 45
Gambar 4. 11 Ciphertext yang akan didekrip 45
Gambar 4. 12 Determinan Dari Nilai Matriks Kunci 46
Gambar 4. 13 Invers Matriks Kunci 46
Gambar 4. 14 Proses Perhitungan Untuk *Decript* Pesan 47

Gambar 4. 15 Fitur *Login Website* 47
Gambar 4. 16 Fitur Registrasi pada *Website* 48
Gambar 4. 17 Tampilan *Dashboard* Saat Pengguna Berhasil *Login* 48
Gambar 4. 18 Tampilan Menu Encode Saat Belum Melalui Proses Cipher 49
Gambar 4. 19 Tampilan Menu Encode Saat Sudah Melalui Proses Cipher 49
Gambar 4. 20 Tampilan Menu Encode Untuk Menyisipkan Pesan ke Dalam Gambar 50
Gambar 4. 21 Tampilan Menu Decode Untuk Ekstraksi Pesan Yang Ada Dalam Sebuah Gambar 51
Gambar 4. 22 Tampilan Menu Decode Untuk Ekstraksi Huruf Acak Menjadi Sebuah Kalimat 52
Gambar 4. 23 Tampilan Menu Informasi Pada Website 52
Gambar 4. 24 Hasil Dari Tes MSE dan PSNR 54
Gambar 4. 25 Proses Cipher Berhasil 55
Gambar 4. 26 Proses Decipher Gagal 55
Gambar 4. 27 Proses Cipher Berhasil 56
Gambar 4. 28 Proses Decipher Berhasil 56
Gambar 4. 29 Proses Login Gagal Dengan Keterangan Password Salah 59
Gambar 4. 30 Poses login gagal dengan keterangan email yang dimasukkan salah 59
Gambar 4. 31 Poses login gagal dengan keterangan email dan password harus diisi 60
Gambar 4. 32 Proses Registrasi Gagal Dengan Keterangan Harus Mengisi Semua Data Pada Form Yang Disediakan 62
Gambar 4. 33 Proses Registrasi Gagal Dengan Keterangan Nama Lengkap Belum Diisikan 62
Gambar 4. 34 Proses Registrasi Gagal Dengan Keterangan Email Belum Diisikan 63
Gambar 4. 35 Proses Registrasi Gagal Dengan Keterangan Password Belum Diisikan 63

Gambar 4. 36 Proses Registrasi Berhasil Dan Pengguna kan Diarahkan Ke Halaman Login 64
Gambar 4. 37 Proses Enkripsi Gagal Dengan Keterangan Pesan Belum Diisikan 66
Gambar 4. 38 Proses Enkripsi Gagal Dengan Keterangan Kunci Belum Dimasukkan 67
Gambar 4. 39 Proses Enkripsi Berhasil dan Akan Muncul Pesan Hasil Enkripsi 67
Gambar 4. 40 Proses Penyisipan Pesan Gagal Dengan Keterangan Gambar Belum Dimasukkan 68
Gambar 4. 41 Proses Penyisipan Pesan Gagal Dengan Keterangan Gambar dan Pesan Belum Dimasukkan 68
Gambar 4. 42 Proses Penyisipan Pesan Gagal Dengan Keterangan Pesan Belum Dimasukkan 69
Gambar 4. 43 Proses Penyisipan Berhasil dan Akan Menampilkan Gambar yang Telah Disisipi Pesan 69
Gambar 4. 44 Proses Embedded Gagal Dengan Keterangan Belum Ada Gambar Yang Diunggah 71
Gambar 4. 45 Proses Embedded Berhasil Dan Akan Muncul Pesan Yang Tersembunyi Di Dalam Gambar 72
Gambar 4. 46 Proses Decipher Gagal Dengan Keterangan Belum Memasukkan Pesan Dan Kunci 73
Gambar 4. 47 Proses Decipher Berhasil Dan Akan Terlihat Pesan Yang Telah Didekrip 73
Gambar 4. 48 Pengiriman Gambar Stegano Melalui Dokumen WhatsApp 75
Gambar 4. 49 Gambar Stegano Dikirim Melalui Dokumen WhatsApp 76
Gambar 4. 50 Gambar Stegano Sesudah Dikirim Melalui Dokumen WhatsApp. 76
Gambar 4. 51 Gambar Stegano Dikirim Menggunakan Media WhatsApp 78
Gambar 4. 52 Gambar Stegano Dikirim Menggunakan Media WhatsApp 78

DAFTAR TABEL

Halaman

Tabel 2. 1 Jenis Diagram Resmi UML 18
Tabel 2. 2 Simbol-simbol Use Case 19
Tabel 2. 3 Simbol-simbol DFD 20
Tabel 2. 4 Simbol-simbol Flowchart 21
Tabel 2. 5 Simbol-simbol ERD 22
Tabel 2. 6 State of The Art 25
Tabel 3. 1 Jadwal Kegiatan 35
Tabel 4. 1 Jumlah Piksel RGB Pada Gambar "ig2.png" 53
Tabel 4. 2 Selisih Piksel Pada Gambar "ig2.png" Setelah Melalui proses MSE .. 53
Tabel 4. 3 Nilai PSNR yang Didapatkan Pada Gambar "ig2.png" 54
Tabel 4. 4 Tabel Pengujian Menu *Login* Menggunakan Metode *BlackBox* Testing 57
Tabel 4. 5 Pengujian Menu Registrasi 60
Tabel 4. 6 Pengujian Menu *Encode* 64
Tabel 4. 7 Pengujian Menu *Decode* Menggunakan Metode *BlackBox Testing* 70
Tabel 4. 8 Pengujian Menu Informasi dan *Test* Menggunakan Metode *BlackBox Testing* 74
Tabel 4. 9 Pengujian Menggunakan Dokumen Whatapp 77
Tabel 4. 10 Pengujian Menggunakan Media Whatapp 79

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pesatnya perkembangan teknologi di era saat ini, memudahkan masyarakat bertukar informasi dalam media digital seperti teks, audio, video, dan citra. Perkembangan Informasi dan Komunikasi menjadikan kegiatan penyampaian informasi maupun data menjadi lebih efisien. Perkembangan teknologi saat ini yang sangat signifikan memberikan dampak bagi masyarakat dalam bertukar informasi maupun melakukan komunikasi. Secara umum informasi dikategorikan menjadi dua, yaitu informasi yang bersifat rahasia dan informasi yang tidak bersifat rahasia. Informasi yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Informasi bersifat rahasia yaitu setiap informasi yang ada didalamnya sangat berharga bagi pihak yang membutuhkan karena informasi tersebut dapat dengan mudah digandakan. (Desimeri Laoli, 2020).

Saat ini telah banyak cara yang dapat dilakukan untuk penyembunyian pesan dalam pengiriman data dengan merubah data menjadi yang tidak dimengerti oleh pihak yang tidak memiliki akses untuk menerima pesan tersebut. Salah satu cara penyembunyian pesan dalam pengiriman yaitu dengan penyandian dan penyisipan menggunakan teknik kriptografi dan steganografi. Maka dari itu penulis menggabungkan metode kriptografi yaitu algoritma *Hill Cipher* dan metode steganografi yaitu *Least Significant Bit* (*LSB*). (Jane Irma Sari, 2017). Tujuan untuk dapat menjaga dan memberikan keamanan yang berlapis tanpa mengurangi atau merusak pesan teks pada citra digital yang akan dikirim.

Seiring dengan majunya perkembangan teknologi, menyebabkan adanya cara-cara terbaru, yang digunakan dengan tidak bertanggung jawab oleh beberapa oknum yang menyalahgunakan fungsi keamanan akan sebuah sistem informasi. Ini sangat berisiko karena kemudahan dalam mendapatkan informasi saat ini memudahkan oknum dalam mendapatkan informasi pribadi milik seseorang

sehingga sangat merugikan dan meresahkan. .karena Informasi tersebar ke tangan okum lain dapat menimbulkan efek negatif untuk pemilik informasi dan disalah gunakan seperti merubah di bagian-bagian tertentu sehingga banyak terjadi perubahan. Pihak yang tidak bertanggung jawab menggunakan data tersebut untuk kepentingan tertentu yang tidak ada kaitannya dengan pemilik informasi tersebut. Kemudian Data-data yang penting jika tidak dilindungi secara optimal maka rentan akan pencurian atau kehilangan data. Pencurian atau kehilangan data tentunya akan merugikan seseorang.

Mengacu pada permasalahan yang dibahas maka diperlukan untuk merancang sebuah sistem keamanan yang dapat melindungi data yang dianggap penting dengan penyandian data, serta membuat kunci rahasia untuk dapat membuka data tersebut yang sulit untuk dideteksi oleh pihak yang tidak berhak. Gabungan dari metode kriptografi yakni algoritma *Hill Cipher* untuk enkripsi pesan dengan metode steganografi yaitu *Least Significant Bit* (*LSB*) dapat menambah keamanan dalam sebuah pesan (Desimeri Laoli, 2020).

Kriptografi merupakan salah satu ilmu maupun seni untuk menjaga kerahasiaan sebuah pesan dengan cara menyandikan ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa alur kerja dari proses enkripsi. Algoritma yang dipakai pada penyusunan skripsi berikut ini yaitu *Hill Cipher* (Fresly Nandar Pabokory, 2015). Algoritma *Hill Cipher* menggunakan matriks berukuran m x m sebagai kunci untuk melakukan enkripsi dan dekripsi. Satu cara untuk mendapatkan kembali naskah asli tentunya dengan menerka kunci dekripsi, jadi proses menerka kunci dekripsi harus menjadi sesuatu yang sulit. memecahkan chiperteks menjadi plainteks tanpa mengetahui kunci yang digunakan.

Sedangkan steganografi adalah seni untuk menyisipkan pesan rahasia kedalam suatu media, dimana pesan rahasia yang akan disembunyikan tidak diubah bentuknya, melainkan disisipkan pada sebuah citra digital, sehingga orang lain tidak mengetahui bahwa di dalam citra digital tersebut ada pesan rahasia.

Setelah dibubuhi pesan rahasia, setiap pixel dibangun kembali menjadi gambar yang utuh menyerupai dengan media gambar semula (Desimeri Laoli, 2020). Metode steganografi yang digunakan adalah metode *Least Significant Bit* (*LSB*). Metode ini merupakan penyembunyian pesan yang dilakukan mengganti bit-bit data yang kurang berarti dalam segmen citra dengan bit-bit rahasia pada bit terakhir (Jane Irma Sari, 2017).

Beberapa penelitian terdahulu yang terkait dengan pengembangan implementasi penggabungan algoritma *hill cipher* dan metode *least significant bit* (*LSB*) yaitu penelitian dari (Ardiansyah dan Kurniasih, 2018) dengan judul *Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit* berdasarkan pengembangan sistem yang telah dibuat menghasilkan beberapa fitur diantaranya *add file, add text, cover image, set password, embed.* Kemudian penelitian dari (Agustinus Noertjahyana dkk, 2012) dengan judul Aplikasi Metode *Steganography* Pada Citra Digital Dengan Menggunakan Metode *LSB (Least Significant Bit)* dari pengembangan sistem yang telah dibuat menghasilkan beberapa fitur diantaranya *home, embed, help,* dan *exit*. Lalu penelitian dari (Hasugian, 2013) *Implementasi Algoritma Hill Cipher Dalam Penyandian Data* dari pengembangan sistem yang telah dibuat menghasilkan beberapa fitur diantaranya enkripsi dan dekripsi.

Berdasarkan beberapa penelitian terdahulu saya melakukan sebuah inovasi untuk membangun sebuah sistem Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* Dan Metode *Least Significant Bit (LSB)* berbasis website dengan beberapa fitur unggulan yang ada di dalamnya. Ini merupakan sebuah sistem yang berfokus pada pengaman pesan *text* yang terlebih dahulu dilakukan proses enkripsi lalu pesan yang telah dilakukan proses enkripsi tersebut dimasukkan ke dalam sebuah citra *digital* berupa gambar JPG dengan menggunakan metode *least significant bit (LSB).* Adapun fitur yang ada pada aplikasi ini adalah *button encode* untuk melakukan proses penyembunyian pesan pada digital, yang kedua *button decode* untuk melakukan proses ekstraksi pada pesan dari steganografi *image*.

Berdasarkan permasalahan yang sudah dijabarkan di atas, maka saya Ramma Eka Putera selaku peneliti terkait untuk melakukan penelitian yang berjudul "IMPLEMENTASI PENYEMBUNYIAN PESAN PADA CITRA *DIGITAL* DENGAN MENGGABUNGKAN ALGORITM*A HILL CIPHER* DAN METODE *LEAST SIGNIFICANT BIT (LSB)*". Harapannya hasil dari penelitian ini dapat membantu pengguna aplikasi untuk menyembunyikan informasi yang bersifat rahasia agar tidak mudah ditemukan atau diakses oleh orang yang tidak bertanggung jawab sehingga keamanan data atau informasi pengirim lebih terjaga.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka didapatkan rumusan masalah sebagai berikut :

a. Bagaimana cara implementasi penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dan metode *least significant bit* yang berbasis web?
b. Bagaimana cara mengukur perbedaaan citra *digital* sebelum dan sesudah steganografi menggunakan *peak signal-to-noise ratio (PSNR,)* dan *Mean Square Error (MSE)*?
c. Bagaimana cara mengevaluasi fungsionalitas sistem menggunakan metode *black box testing*?
d. Bagaimana proses steganografi pada gambar berformat PNG sebagai format yang digunakan untuk stiker WhatsApp?
e. Bagaimana perbedaan gambar sebelum dan sesudah dikirim melalui whatsapp?

1.3 Tujuan

Tujuan dari dilakukannya penelitian ini adalah sebagai berikut:

a. Untuk mengembangkan aplikasi dengan mengimplementasikan penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dan metode *least significant bit* yang berbasis *web*

b. Untuk mengetahui apakah aplikasi penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dan metode *least significant bit* fungsionalitas telah berjalan dengan baik sesuai yang diharapkan pengguna dengan melakukan *Black box testing,*
c. Untuk mengetahui apakah aplikasi penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dan metode *least significant bit* memiliki perbedaan yang signifikan pada gambar yang digunakan pada *cover image* .

1.4 Manfaat

Manfaat yang diharapkan setelah melakukan penelitian ini adalah sebagai berikut:

a. Manfaat bagi pengguna, dapat membantu pengguna dalam menyisipkan pesan ke dalam citra *digital* sehingga isi pesan tersebut yang bersifat rahasia tidak mudah untuk disalahgunakan oleh oknum yang tidak bertanggung jawab.
b. Manfaat bagi penulis, dapat menambah pengetahuan tentang bagaimana cara mengembangkan aplikasi *website* serta mengevaluasinya menggunakan *Black box testing, Peak signal-to-noise ratio (PSNR,)* dan *measure of the quality of an estimator (MSE).*
c. Bisa memberikan informasi yang tepat dan akurat kepada penerima pesan yang dikirim oleh pengirim pesan dengan mengurangi risiko penyalahgunaan pesan dari pihak yang tidak bertanggung jawab.

1.5 Batasan Masalah

Agar pembahasan masalah tidak terlalu melebar dan lebih terfokus, maka permasalahan dibatasi oleh beberapa hal:

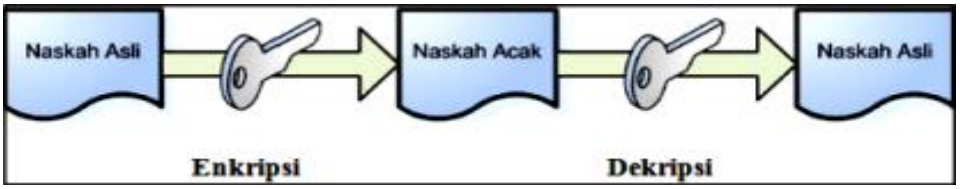
a. Penelitian ini ditujukan untuk pengguna aplikasi yang sudah melakukan proses registrasi.
b. Aplikasi penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dan metode *least significant* dibuat berbasis web.

c. Model pengembangan aplikasi menggunakan algoritma *hill cipher* dan metode *least significant bit.*
d. Alat evaluasi aplikasi yang digunakan adalah *Black box testing* untuk mengevaluasi fungsionalitas sistem, *Peak signal-to-noise ratio (PSNR,)* dan *measure of the quality of an estimator (MSE)* untuk mengukur gambar citra sesudah dan sebelum melalui proses steganografi.
e. Dalam penelitian ini hanya menggunakan media penyembunyian berupa citra digital gambar dengan format PNG.
f. Aplikasi ini dibuat hanya untuk menyembunyikan data rahasia yang berupa data teks.

BAB 2. TINJAUAN PUSTAKA

2.1 Kriptografi

Apakah sebenarnya kriptografi itu? Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Dekripsi menggunakan kunci dekripsi mendapatkan kembali data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci).



Gambar 2. 1 Proses Enkripsi dan Dekripsi

Gambar diatas menunjukkan efek dari proses enkripsi dan proses dekripsi. Secara garis besar, proses enkripsi adalah proses pengacakan “naskah asli" (*plaintext*) menjadi “naskah acak" (*ciphertext*) yang “sulit untuk dibaca" oleh seseorang yang tidak mempunyai kunci dekripsi. Yang dimaksud dengan “sulit untuk dibaca" disini adalah probabilitas mendapat kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu yang tidak terlalu lama adalah sangat kecil. Jadi suatu proses enkripsi yang baik menghasilkan naskah acak yang memerlukan waktu yang lama (contohnya satu juta tahun) untuk didekripsi oleh seseorang yang tidak mempunyai kunci dekripsi. (Sentot Kromodimoeljo, 2010)

2.2 Algoritma Hill *Cipher*

Hill Cipher diciptakan oleh Lester S. Hill pada tahun 1929. Teknik kriptografi ini diciptakan dengan maksud untuk dapat menciptakan *cipher* (kode) yang tidak dapat dipecahkan menggunakan teknik analisis frekuensi. *Hill Cipher* merupakan salah satu algoritma kriptografi kunci simetris yang memiliki beberapa kelebihan dalam enkripsi data. Untuk menghindari matrik kunci yang tidak *invertible*, matrik kunci dibangkitkan menggunakan koefisien binomial newton. Proses enkripsi dan dekripsi menggunakan kunci yang sama, plaintext dapat menggunakan media gambar atau *text*.

Algoritma *Hill Cipher* menggunakan matriks berukuran m x m sebagai kunci untuk melakukan enkripsi dan dekripsi. Dasar teori matriks yang digunakan dalam *Hill Cipher* antara lain adalah perkalian antara matriks dan melakukan *invers* pada matriks. *Hill Cipher* merupakan penerapan aritmatika modulo pada kriptografi. Teknik kriptografi ini menggunakan sebuah matriks persegi sebagai kunci yang digunakan untuk melakukan enkripsi dan dekripsi. *Hill Cipher* tidak mengganti setiap abjad yang sama pada plaintext dengan abjad lainnya yang sama pada *ciphertext* karena menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya.

Hill Cipher yang merupakan *polyalphabetic cipher* dapat dikategorikan sebagai *block cipher* karena teks yang akan diproses akan dibagi menjadi blok-blok dengan ukuran tertentu. Setiap karakter dalam satu blok akan saling mempengaruhi karakter lainnya dalam proses enkripsi dan dekripsinya, sehingga karakter yang sama tidak dipetakan menjadi karakter yang sama pula. *Hill Cipher* termasuk kepada algoritma kriptografi klasik yang sangat sulit dipecahkan oleh kriptanalis apabila dilakukan hanya dengan mengetahui berkas *ciphertex*t saja. Namun, teknik ini dapat dipecahkan dengan cukup mudah apabila kriptanalis memiliki berkas *ciphertext* dan potongan berkas *plaintex*t. Teknik kriptanalisis ini disebut *known-plaintext attack*. (Muamal Khoerudin, 2015)

2.3 Steganografi

Steganografi adalah seni menyembunyikan informasi untuk mencegah pendeteksian pesan yang disembunyikan (Guillermito. 2004). Steganografi berasal dari bahasa Yunani yang memiliki arti penulisan terlapis (*covered writing*), termasuk di dalamnya suatu metode komunikasi rahasia dalam jumlah besar yang menyembunyikan pesan dengan sangat baik. *Steganography* dan *Cryptography* memiliki garis besar tujuan yang sama yaitu mengamankan suatu informasi namun terdapat perbedaan mendasar yang terletak pada cara pengamanannya. *Cryptography* mengacak pesan sehingga tidak dapat terbaca, sedangkan *Steganography* bertujuan untuk menyembunyikan informasi sehingga tidak dapat terlihat. Pada *cryptography*, informasi yang tersimpan dalam bentuk *ciphertext* dapat menimbulkan kecurigaan pada penerima sehingga dapat menyebabkan timbulnya usaha untuk melakukan pembobolan (*hacking*), namun hal ini tidak terjadi pada informasi tersembunyi (*hidden message*) yang diolah dengan metode Steganografi.

Secara garis besar metode Steganography terdiri dari 2 bagian utama (Kharrazi, 2004), yaitu proses penyembunyian data (*hidden message*) dan proses pengembalian data ke bentuk semula (*reveal message*). Kedua proses ini dilakukan dengan menggunakan sebuah kata kunci rahasia (*secret key*) yang akan digunakan di dalam prosesnya untuk meningkatkan keamanan data. Untuk lebih jelas mengenai konsep steganografi.

Gambar 2. 2 Proses Penyimpanan Data Rahasia ke Dalam Media Digital dengan Teknik Steganografi

2.4 Citra Digital (Bitmap)

Citra adalah representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada

monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra digital adalah citra yang dapat diolah oleh komputer. Istilah citra digital sangat popular pada masa sekarang. Banyak peralatan elektronik, misalnya *scanner*, kamera digital, mikroskop digital, dan fingerprint reader (pembaca sidik jari), yang menghasilkan citra digital juga sangat populer digunakan oleh pengguna untuk mengolah foto atau untuk berbagai keperluan lain. Sebagai contoh, Adobe Photoshop dan GIMP (GNU Image Manipulation Program) menyajikan berbagai fitur untuk memanipulasi citra digital. (Abdul Kadir dan Adhi Susanto, 2013)

Citra digital yang digunakan untuk menyembunyikan pesan yang adalah *Bitmap*. Yang paling penting dari kriteria ini adalah kedalaman warna (berapa banyak bit per *pixel* yang didefinisikan dari sebuah warna) sebagai berikut :

4 bit = 16 warna (16 gray scales).

8 bit = 256 warna (256 gray scales).

24 bit = 16.777.216 warna.

Secara umum semakin banyaknya warna, maka akan diperlukan keamanan yang ketat atau tinggi dikarenakan bitmap memiliki area yang sangat luas dalam sebuah warna yang seharusnya dihindarkan. Dilihat dari kedalaman atau kejelasan dari sebuah warna, bitmap dapat mengambil sejumlah data tersembunyi dengan perbandingan sebagai berikut (ukuran perbandingan dari bitmap dalam byte = ukuran dari data yang disembunyikan) :

4 bit = 16 warna : 4 : 1

8 bit = 256 warna : 8 : 1

24 bit = 16.777.216 warna : 8 : 1

Manipulasi pada bitmap tidak dapat di convert atau diubah ke dalam bentuk format grafik yang lain karena data tersembunyi dalam *file* tersebut akan hilang. Format menggunakan metode kompresi yang lain (seperti JPEG) tidak

dapat digunakan. Mengurangi ukuran dari file pembawa sangatlah penting untuk melakukan transmisi online, yaitu dengan menggunakan utilitas kompresi (seperti : ARZ, LZH, PKZIP, WinZip), dikarenakan kerja mereka tidak terlalu berat. Untuk dapat menyisipkan pesan rahasia pada file bitmap maka terlebih dahulu harus diketahui struktur *file Bitmap*. (Jane Irma Sari, 2017)

2.5 Metode Least Significant Bit (LSB)

Least Significant Bit adalah salah satu metode untuk menyembunyikan pesan dalam media digital dengan cara menyisipkan pesan tersebut pada satu bit paling kanan ke pixel file objek. Dalam menyisipkan data pesan ke dalam berkas citra digital dengan menggunakan metode *Least Significant Bit (LSB) Modification*. Misalkan untuk menyisipkan suatu segmen pesan hasil dan modulasi sebesar 4 byte dengan modifikasi 1 bit *LSB*, maka dibutuhkan 32 data citra digital untuk menampungnya. dari segmen pesan ' 1 0 1 0 ' dengan 4 byte data citra digital sebagai berikut:

'0 1101110 00100011 01000010 01101101'

Maka dengan operasi penggantian bit terakhir dengan 4 bit segmen pesan secara berurutan menjadi sebagai berikut:

Data citra digital: ’0 1 1 0 1 1 1 0 0 0 1 0 0 0 1 1 0 10 0 0 0 1 0 0 1 1 0 1 1 0 1’

Pesan: 1 0 1 0

Hasil: ’0 1 1 0 1 1 1 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 1 1 0 1 1 0 1 1 0 0’

Dengan sedikit modifikasi ini, maka efek dari perubahan nilai warna yang terjadi akibat perubahan bit tersebut tidak terlalu berpengaruh terhadap kualitas gambar. Perhatikan contoh untuk menyisipkan sebuah karakter A ke dalam citra *grayscale*.

Sebuah pesan huruf A akan disisipkan ke dalam citra *grayscale* 8 bit ukuran 10x10 piksel.

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3

Gambar 2. 3 Citra grayscale 8 bit 10x10 pikselCitra grayscale 8 bit 10x10 piksel

Langkah pertama adalah mengubah kedua data tersebut (huruf A dan citra) menjadi biner. Nilai biner untuk A adalah 10000011. Karena jumlah digit biner huruf A hanya 8 bit maka jumlah piksel citra *grayscale* yang dibutuhkan cukup 8 piksel saja. Perhatikan 8 piksel pertama dari citra yang diubah menjadi biner.

8 piksel pertama diambil

1	6	5	3	7	4	7	4	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Gambar 2. 4 Citra grayscale 8 piksel yang diambil Langkah kedua adalah mengganti bit terakhir (LSB) dari piksel citra dengan bit-bit dari huruf A.

Perhatikan bit-bit yang ditandai dengan kotak. Bit-bit piksel citra mengalami perubahan (dalam hal ini yang berubah hanya 4 piksel saja) sehingga citra berubah menjadi:

8 piksel pertama diambil

1	6	4	2	6	4	7	5	1	0
3	5	3	5	5	5	5	7	7	0
0	0	0	2	2	6	6	6	6	6
5	5	4	4	4	4	4	4	7	3
2	2	0	0	0	0	1	1	1	1
7	5	5	5	7	7	7	6	3	3
3	3	3	3	3	3	3	3	7	5
5	5	5	5	5	5	5	5	2	3
0	0	0	0	0	0	4	4	4	4
3	3	3	3	3	1	1	1	6	2

Gambar 2. 5 Grayscale piksel yang berubah

Tampak bahwa piksel-piksel yang mengalami perubahan hanya kurang lebih intensitas saja. Maka, secara kasat mata hal ini tidak begitu berpengaruh. Selain itu, tidak semua piksel mengalami perubahan intensitas. Ukuran data maksimum yang bisa disembunyikan dengan metode *LSB* yaitu seperti contoh yang berikut ini:

Media penampung : Citra *grayscale* 8 bit berukuran 64x32 pixel.

Ukuran media penampung = 64 x 32 x 8 bit

= 16384 bit

1 piksel media penampung = 8 bit

Untuk menampung 1 bit data pesan diperlukan 1 piksel citra media penampung berukuran 8 bit karena setiap 8 bit hanya bisa menyembunyikan satu bit di *LSB*-nya. Oleh karena itu, citra ini hanya mampu menampung data pesan sebesar maksimum 16384/8 = 2048 bit dikurangi panjang nama pada *file* karena penyembunyian data rahasia tidak hanya menyembunyikan isi data tersebut, tetapi juga nama pada *file*. Semakin besar data yang disembunyikan di dalam citra,

semakin besar pula kemungkinan data tersebut rusak akibat manipulasi pada citra penampung.(Sutoyo, 2009)

2.6 Aplikasi

Menurut Kamus Besar Bahasa Indonesia (KBBI) aplikasi merupakan penerapan rancang sistem untuk mengolah data dengan menggunakan aturan tertentu atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang digunakan untuk melakukan tugas tertentu dari *user* (pengguna).

Aplikasi adalah suatu program yang siap pakai yang digunakan untuk menjalankan fungsi tertentu oleh pengguna (Juansyah, 2015). Secara istilah aplikasi merupakan sebuah rancang sistem yang dibuat dengan bahasa pemrograman tertentu yang berguna untuk melakukan tugas tertentu dari *user* pengguna.

2.7 Website

Website merupakan kumpulan dari beberapa halaman web yang dipublikasikan melalui jaringan *internet*. Setiap website memiliki alamat domain/*URL* (*Uniform Resource Locator*) sehingga dapat diakses oleh seluruh pengguna internet dengan cara mengetikkan alamat dari *website* tersebut (Rudianto, 2011).

Halaman *website* biasanya ditulis dengan dokumen yang berformat *Hyper Text Markup Language* (HTML), yang bisa diakses melalui HTTP, HTTPS yang berfungsi sebagai protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada *user* (pengguna) melalui web browser (Nofyat, dkk, 2018).

2.8 XAMPP

XAMPP merupakan sebuah perangkat lunak berbasis *web server* bersifat *open source* (bebas). perangkat ini mendukung banyak sistem operasi, baik Linux, Windows, dan Mac OS. XAMPP digunakan sebagai *standalone server* (berdiri sendiri) atau biasa disebut dengan *localhost* (*web server* lokal yang ada di dalam

komputer kita sendiri) sehingga kita dapat dengan mudah mengedit *script website*, mendesain, pengembangan aplikasi.

Kata XAMPP sendiri berasal dari:

a. X berarti *cross platform* bisa di jalankan di beberapa sistem operasi, yaitu Linux, Windows, Mac OS.
b. A berarti Apache sebagai web servernya.
c. M berarti MySQL sebagai *Database Management System* (DBMS)-nya.
d. PP berarti PHP dan Pearl merupakan bahasa pemrograman yang didukungnya.

(Hidayatullah dan Kawistara, 2017).

2.9 Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data. Basis berarti gudang, markas, tempat berkumpul, kemudian data yaitu fakta yang mewakili suatu objek seperti hewan, manusia, barang, kendaraan, dan sebagainya yang direkam dalam bentuk simbol, angka, teks, atau kombinasinya. Definisi dari basis data adalah kumpulan data yang disimpan dalam media penyimpanan elektronik yang saling berhubungan dan diorganisasikan sedemikian rupa tanpa pengulangan (redudansi) agar nantinya data tersebut dapat dimanfaatkan kembali dengan cepat dan mudah (Yanto, 2016).

2.10 MySQL

Database Management System (DBMS) adalah aplikasi yang digunakan untuk mengelola basis data. DBMS memiliki beberapa fitur yang terintegrasi meliputi:

1. Membuat menghapus, menambah, dan memodifikasi basis data.
2. Pengelolaan berbasis windows sehingga lebih mudah digunakan.

3. Kemampuan berkomunikasi dengan aplikasi lain, misalnya mengakses basis data MySQL menggunakan aplikasi yang menggunakan bahasa pemrograman PHP.
4. Memberikan keamanan data.
5. Kemampuan komunikasi antarkomputer (*client server*).

MySQL sendiri adalah salah satu aplikasi DBMS yang sering digunakan dalam pengembangan aplikasi web, contoh lain dari DBMS yang lain adalah, PostgreSQL, SQL Server, Microsoft Access, Oracle, Dbase, dan lain sebagainya (Hidayatullah dan Kawistara, 2017).

Kelebihan dari MySQL adalah bersifat *open source* atau gratis. Keunggulan dari menggunakan basis data MySQL adalah:

1. Kecepatan.
2. Kemudahan bagi user.
3. Gratis atau *open source*.
4. Support dengan bahasa *query*.
5. User dapat mengakses lebih dari satu dalam satu waktu.
6. MySQL mudah didapatkan karena *source code* yang mudah untuk disebarluaskan.
7. Akses data dapat dilakukan di setiap tempat dengan fasilitas internet.

(Yanto, 2016).

2.11 PHP

PHP merupakan bahasa yang khusus dirancang untuk pengembangan aplikasi *website*. PHP merupakan *tool* untuk membuat halaman *website* yang dinamis. PHP pada awalnya adalah kepanjangan dari *Personal Home Pages* (Situs Pribadi). PHP dibuat pertama kali oleh Rasmus Lerdorf pada tahun 1995 dan diberi nama FI (*Form Interpreted*) digunakan untuk mengolah data form yang ada di *web*site (Sianipar, 2015).

Hypertext Preprocessor atau biasa disebut dengan PHP adalah salah satu bahasa *scripting* yang digunakan untuk pengembangan aplikasi web. Karena sifatnya yang *scripting* untuk menjalankannya harus menggunakan *web server*. PHP dapat diintegrasikan dengan HTML, *Javascript, JQuery, Ajax* (Hidayatullah dan Kawistara, 2017).

2.12 Framework CodeIgniter

CodeIgniter merupakan salah satu *framework* dari bahasa pemrograman PHP yang bersifat *open source* atau gratis. *Framework* ini menggunakan konsep MVC (*Model, View, Controller*). *CodeIgniter* memiliki *library* yang lengkap sehingga ketika menjalankan operasi-operasi umum yang ada pada suatu *website* seperti mengakses basis data, memvalidasi *form* menjadi semakin mudah. *CodeIgniter* memiliki dokumentasi yang lengkap dan mudah dipahami. Di dalam *source code CodeIgniter* terdapat *coment* sehingga mempermudah programmer dalam memahami suatu fungsi tersebut yang menghasilkan *code* yang *clean* (bersih) dan *Search Engine Friendly* (SEF). *CodeIgniter* juga mempermudah para programmer dalam mengembangkan aplikasi *website*, karena sudah memiliki kerangka kerja sehingga tidak perlu menuliskan kode dari awal secara manual, selain itu *CodeIgniter* memiliki struktur folder yang teratur sehingga lebih cepat dalam pengembangan aplikasi dan dapat fokus dalam menentukan fitur-fitur apa saja yang ada di dalam aplikasi (Basuki, 2010).

2.13 Pengembangan Sistem

2.13.1 Unified Modeling Language

Unified Modelling Language (*UML*) adalah bahasa yang dijadikan standar dalam industri dalam bentuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. *UML* mendefinisikan notasi dan *syntax*/semantik. *UML* diibaratkan cetak biru pada sebuah bangunan. Desain dengan *UML* digunakan oleh seorang programmer untuk melakukan pengkodean. Bisa jadi seorang desainer adalah juga seorang *programmer* (Mujilan, 2017).

UML memiliki 13 jenis diagram resmi dan masing-masing memiliki kegunaan yang berbeda seperti yang ada pada tabel berikut:

Tabel 2. 1 Jenis Diagram Resmi UML (Fowler, 2005).

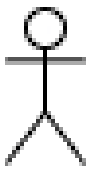
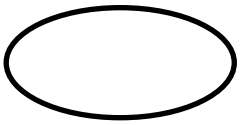
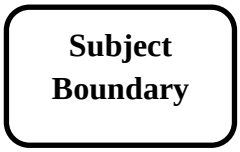
No.	Diagram	Kegunaan
1.	*Activity*	*Behavior* procedural dan parallel.
2.	*Class*	*Class, Fitur, dan hubungan-hubungan.*
3.	*Communication*	Interaksi antar objek; penekanan pada jalur.
4.	*Component*	Struktur dan koneksi component.
5.	*Composite structure*	Dekomposisi *runtime* sebuah *class.*
6.	*Deployment*	Pemindahan artifak ke *node.*
7.	*Interaction overview*	Campuran *sequence* dan *activity* diagram.
8.	*Object*	Contoh konfigurasi.
9.	*Package*	Struktur hirarki *compile-time.*
10.	*Sequence*	Interaksi antar objek; penekanan pada *sequence*.
11.	*State machine*	Menggambarkan perubahan yang berasal dari objek.
12.	*Timing*	Interaksi antar objek; penekanan pada *timing.*
13.	*Use case*	Bagaimana pengguna berinteraksi dengan sebuah sistem.

2.13.2 Diagram *Use Case*

Diagram *use case* adalah diagram yang menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam *use case* di gambarkan interaksi antara aktor dengan sistem. Aktor sendiri merupakan gambaran dari sebuah entitas manusia atau mesin yang melaksanakan tugas-tugas tertentu. Diagram *use case* sangat membantu dalam Menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan sistem dengan

client, dan merancang *test case* untuk semua fitur yang ada di dalam sistem (Mujilan, 2017). Berikut ini elemen-elemen dari diagram *use case*:

Tabel 2. 2 Simbol-simbol Use Case (Indriyani, 2019).

No.	Bentuk Elemen	Keterangan	Penjelasan
1.		*Actor*	*Actor* mewakili entitas dari manusia atau sistem lain yang berinteraksi dengan sistem saat ini.
2.		*Use case*	Bagian utama dari fungsionalitas sistem. Bisa *extends* (memperluas) *use case* lainnya. Ditempatkan dalam *system boundary* (batasan sistem). Dilabeli dengan kata kerja, frase kata benda.
3.		*Subject boundary*	Berupa kotak batasan sistem, di bagian dalam sebelah atas kotak tempat meletakkan judul sistem. *Actor* berada di luar kotak ini.
4.		*Association Relationship*	Menghubungkan *actor* dengan *use case*. Menggambarkan komunikasi 2 arah (menggambarkan komunikasi 1 arah jika menggunakan tanda panah).

Lanjutan Tabel 2.2 Simbol-simbol Use Case (Indriyani, 2019).

5.	<< include >>	*Include Relationship*	Memasukkan *use case* ke dalam *use case* lainnya.
6.	<< extend >>	*Extend Relationship*	Memperluas *use case* memasukkan perilaku opsional.
7.		*Generalization Relationship*	Mewakili *use case* untuk *case* yang lebih umum.

2.13.3 Data Flow Diagram (DFD)

Data flow diagram atau bila disingkat DFD berasal dari sistem informasi manajemen (SIM), digunakan untuk melakukan penggambaran proses sistem, arus antar proses, dan sumber, tujuan, serta penyimpanan data. Terdapat dua tipe DFD yaitu, *logical* DFD yang berfokus pada aktivitas sistem, dan *physical* DFD (mujilan, 2017). Di bawah ini adalah simbol-simbol dari DFD:

Tabel 2. 3 Simbol-simbol DFD (Bagir dan putro, 2018).

NO.	Simbol	Keterangan	Penjelasan
1		Eksternal entity	Sumber tujuan aliran dari suatu sistem.
2		Data store	Menggambarkan bentuk penyimpanan data.

Lanjutan Tabel 2.3 Simbol-simbol DFD (Bagir dan putro, 2018).

3		Proses	Menggambarkan proses bagaimana suatu input menjadi output.
4		Aliran Data	Menggambarkan aliran data dari proses satu ke proses lainnya.

2.13.4 Flowchart

Flowchart atau diagram alur digunakan untuk menunjukan proses informasi seperti arus logika (*logic flows*), *inputs*, *outputs*, penyimpanan data (*data storage*), dan proses operasional (*operating processes*) seperti arus fisik, aktifitas, dan entitas (mujilan, 2017). Berikut ini merupakan simbol-simbol yang ada pada *Flowchart* :

Tabel 2. 4 Simbol-simbol Flowchart (Moutofani dan Rottie, 2018).

Simbol	Keterangan	Penjelasan
	Arus/*flow*	Jalannya arus suatu proses.
	Proses	Proses yang dilakukan oleh.
	Terminal	Menandakan awal atau akhir dari suatu program.

Lanjutan Tabel 2.4 Simbol-simbol Flowchart (Moutofani dan Rottie, 2018).

	Connector	Menyambung dari proses satu ke proses lainnya di halaman yang sama
	Offline connector	Menyambung dari proses satu ke proses lainnya di halaman yang berbeda.
	Input/output	Proses *input/output* tanpa tergantung jenis peralatannya.
	Decision	Kondisi tertentu yang menghasilkan dua kondisi: yaitu YA dan TIDAK

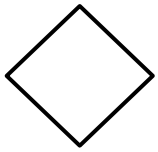
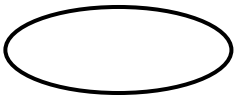
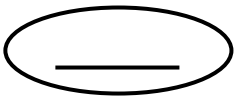
2.13.5 Entity-Relationship Diagram (ERD)

Entity Relationship Diagram disingkat ERD merupakan metode yang dikembangkan oleh Chen pada tahun 1976, berguna untuk menggambarkan struktur database. ERD mengilustrasikan struktur logika dari database dengan memperhatikan entitas-entitas di dalam sistem (Mujilan, 2017). Berikut merupakan simbol-simbol dari ERD:

Tabel 2. 5 Simbol-simbol ERD (Mujilan, 2017).

No.	Simbol	Keterangan	Penjelasan
1.		*Entity*	Merupakan objek yang dapat dibedakan di dunia nyata.

Lanjutan Tabel 2.5 Simbol-simbol ERD (Mujilan, 2017).

2.		*Relationship*	Hubungan yang terjadi antara satu *entity* atau lebih.
3.		Atribut	Atribut merupakan karakteristik yang menjelaskan detail *entity* atau *relationship.*
4.		Atribut *primary key*	Atribut yang menjelaskan *primary key* dari suatu *entity*.

2.14 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses mengevaluasi sebuah perangkat lunak baik secara manual maupun otomatis untuk menguji apakah perangkat lunak telah memenuhi persyaratan atau belum (Clone dan Rood, 2011).

Pengujian perangkat lunak bermaksud untuk mencari kesalahan yang ada pada program serta mengevaluasi kualitasnya (Jin dan Xue, 2011). Singkat kata pengujian perangkat lunak merupakan aktivitas untuk menemukan perbedaaan antara hasil yang diharapkan dengan hasil yang sebenarnya (Sulistiyanto dan Azhari, 2014).

2.14.1 Black Box Testing

Black Box Testing adalah teknik pengujian yang berfokus pada pengujian fungsional dari perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengujian pada spesifikasi fungsional program (Hidayat dan Muttaqin, 2018). Tujuan dari *Black Box Testing* adalah untuk mengetahui cara beroperasi perangkat lunak, apakah proses *input* data dan *output* yang dihasilkan

telah sesuai yang diharapkan dan apakah informasi yang disimpan selalu dijaga kemutakhirannya (Maharani dan Merlina, 2014).

2.14.2 *Peak Signal-to-Noise Ratio* (PSNR)

Peak Signal-to-Noise Ratio (PSNR) adalah salah satu metode yang cukup populer digunakan dalam pengukuran kualitas video secara objektif. Metode ini menggunakan sinyal video sebagai parameter objektif. Metode ini membandingkan sinyal dari setiap frame video 47 pada video hasil transmisi (S-video) dengan setiap frame video pada video sebelum ditransmisikan (O-video) dan mengukur perbedaan keduanya.(Vranjes, 2008).

2.14.3 *Mean Square Error* (MSE)

Mean Squared Error (MSE) adalah Rata-rata Kesalahan kuadrat di antara nilai aktual dan nilai peramalan. Metode *Mean Squared Error* secara umum digunakan untuk mengecek estimasi berapa nilai kesalahan pada peramalan. Nilai *Mean Squared Error* yang rendah atau nilai *mean squared error* mendekati nol menunjukkan bahwa hasil peramalan sesuai dengan data aktual dan bisa dijadikan untuk perhitungan peramalan di periode mendatang.(Khoiri, 2020).

2.15 State of The Art

Penelitian terdahulu oleh Achmad Ardiansyah dan Mepa Kurniasih yang berjudul "Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit" pada tahun 2018 dari aplikasi penyembunyian pesan yang telah dibuat menghasilkan beberapa fitur diantaranya *add file, add text, cover image, set password, embed.*

Selanjutnya penelitian yang serupa oleh Agustinus Noertjahyana dkk yang berjudul "Aplikasi Metode *Steganography* Pada Citra Digital Dengan Menggunakan Metode *Lsb (Least Significant Bit*)" pada tahun 2015 berdasarkan pengembangan sistem yang telah dibuat menghasilkan beberapa fitur diantaranya *home, embed, help,* dan *exit.*

Abdul Halim Hasugian dalam papernya yang berjudul "*Implementasi Algoritma Hill Cipher Dalam Penyandian Data*" pada tahun 2013, dalam paper

tersebut menjelaskan bahwa dari pengembangan sistem yang telah dibuat menghasilkan fitur-fitur, enkripsi dan dekripsi.

Berdasarkan beberapa penelitian terdahulu saya berinisiatif untuk membuat sebuah Aplikasi Penyembunyian Pesan Pada Citra *Digital* Dengan Menggabungkan *Algoritma Hill Cipher* Dan Metode *Least Significant Bit (Lsb)* berbasis *website* dengan beberapa fitur unggulan di dalamnya.

Untuk melihat perbedaan dan klasifikasi tiap penelitian dapat dilihat dalam tabel berikut :

Tabel 2. 6 State of The Art

No	Nama Pengarang	Tahun	Judul	Hasil
1	Achmad Ardiansyah dan Mepa Kurniasih	2018	Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit	Fitur Aplikasi: *add file, add text, cover image, set password, embed.*
2	Agustinus Noertjahyana dkk	2015	Aplikasi Metode *Steganography* Pada Citra Digital Dengan Menggunakan *Metode Lsb (Least Significant Bit)*	Fitur Aplikasi: *home, embed, help,* dan *exit.*
3	Abdul Halim Hasugian	2013	Implementasi Algoritma Hill *Cipher* Dalam Penyandian Data	Fitur Aplikasi: enkripsi *dan* dekripsi

Lanjutan Tabel 2.6 State of The Art

4	Ramma Eka Putera	2021	Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* Dan Metode *Least Significant Bit* (*LSB*)	Fitur Aplikasi: *Cipher*, *Embed*, *Extract*, *Decipher*

BAB 3. METODE PENELITIAN

3.1 Waktu dan Tempat

3.1.1 Tempat Pelaksanaan

Penelitian ini dilakukan di Politeknik Negeri Jember, Jl. Mastrip No.164, Krajan Timur, Sumbersari, Kec. Sumbersari, Kabupaten Jember, Jawa Timur 68121.

3.1.2 Waktu Pelaksanaan

Waktu pelaksanaan dilaksanakan pada rentang waktu 6 (enam) bulan, dimulai pada bulan September 2021 sampai bulan Februari 2022.

3.2 Alat dan Bahan

3.2.1 Alat

Adapun software dan hardware yang digunakan pada proses implementasi yang digunakan yaitu :

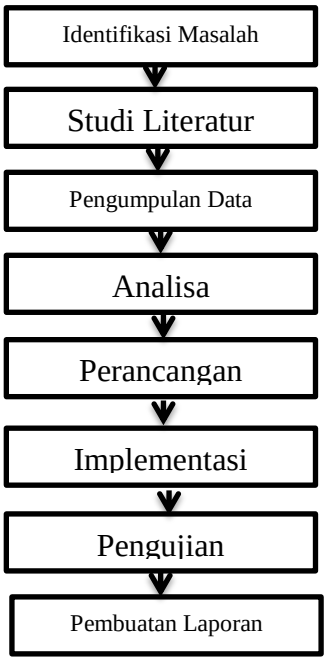
1. Hardware
   a. Processor : Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40 GHz
   b. Memory : 12,0 GB
   c. Harddisk : 500 GB
   d. SSD : 240 GB
2. Software
   a. Bahasa Pemrograman : PHP
   b. DBMS : MySQL database
   c. Web Server : Apache
   d. Browser : Google Chrome
   e. Text Editor : Visual Studio Code

3.2.2 Bahan

Bahan-bahan yang dilakukan dalam pembangunan sistem ini adalah data berupa citra digital yang dapat dilihat pada lampiran 1.

3.3 Tahapan Penelitian

Tahapan penelitian merupakan pedoman untuk melakukan penelitian. Di dalam tahapan penelitian akan menjelaskan tentang tahapan-tahapan yang ada dalam penelitian, agar berjalan sesuai dengan tujuan yang telah ditentukan dan sesuai dengan keinginan. Berikut merupakan tahapan penelitian yang akan dilakukan dalam penyelesaian tugas akhir yang berjudul "Implementasi penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dan metode *least significant bit*". Bisa dilihat pada Gambar 3.1 di bawah:



Gambar 3. 1 Tahapan Penelitian

3.3.1 Identifikasi Masalah

Identifikasi masalah merupakan langkah awal dari metodologi penelitian. Tahap ini untuk mengidentifikasi suatu masalah pada penyalahgunaan pesan rahasia oleh orang yang tidak bertanggung jawab berdasarkan penelitian terkait sebelumnya dan fakta yang berhubungan dengan penelitian.

3.3.2 Studi Literatur

Studi literatur merupakan proses pencarian informasi yang dilakukan dengan pengumpulan teori-teori yang berkaitan dengan penelitian ini sebagai acuan. Sumber teori berasal dari referensi yang berkaitan dengan permasalahan pada penelitian ini melalui hasil penelitian (jurnal atau skripsi), buku terkait baik itu text book maupun e-book, media online dan artikel-artikel terkait.

3.3.3 Pengumpulan Data

Tahapan pengumpulan data diperlukan untuk memperoleh data dan informasi yang berhubungan dengan penelitian ini. Data yang dikumpulkan berasal dari pengambilan sebanyak tiga puluh objek citra digital dengan format PNG untuk dijadikan sebagai data uji.

3.3.4 Analisa

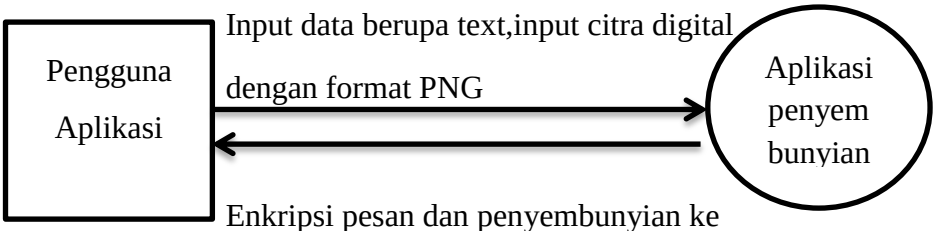
Setelah melakukan tahapan identifikasi masalah, studi pustaka, dan pengumpulan data maka selanjutnya melakukan tahapan analisa. Analisa merupakan langkah-langkah memproses data citra digital dengan format PNG yang dimasukkan ke dalam aplikasi, data ini dimasukkan dengan cara mengunggah menggunakan fitur unggah yang ada pada aplikasi. Kemudian data tersebut diproses dengan menggunakan penggabungan algoritma *hill cipher* dan metode *least significant bit*.

3.3.5 Perancangan

Tahapan setelah melakukan analisa sistem adalah proses perancangan sistem. Tahapan perancangan sistem terdiri dari:

1. DFD *level* 0

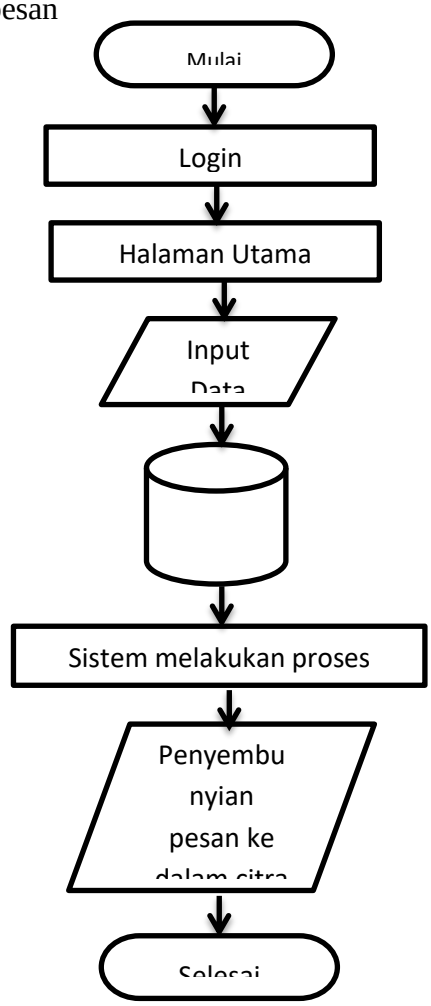
Berikut Gambar 3.2 adalah gambaran umum pada aplikasi yang tergambar dalam bentuk DFD *level* 0



Gambar 3. 2 DFD level0

2. Flowchart Sistem

   Berikut Gambar 3.3 adalah gambaran *flowchart* sistem dari sistem sistem penyembunyian pesan



Gambar 3. 3 *Flowchart* Sistem

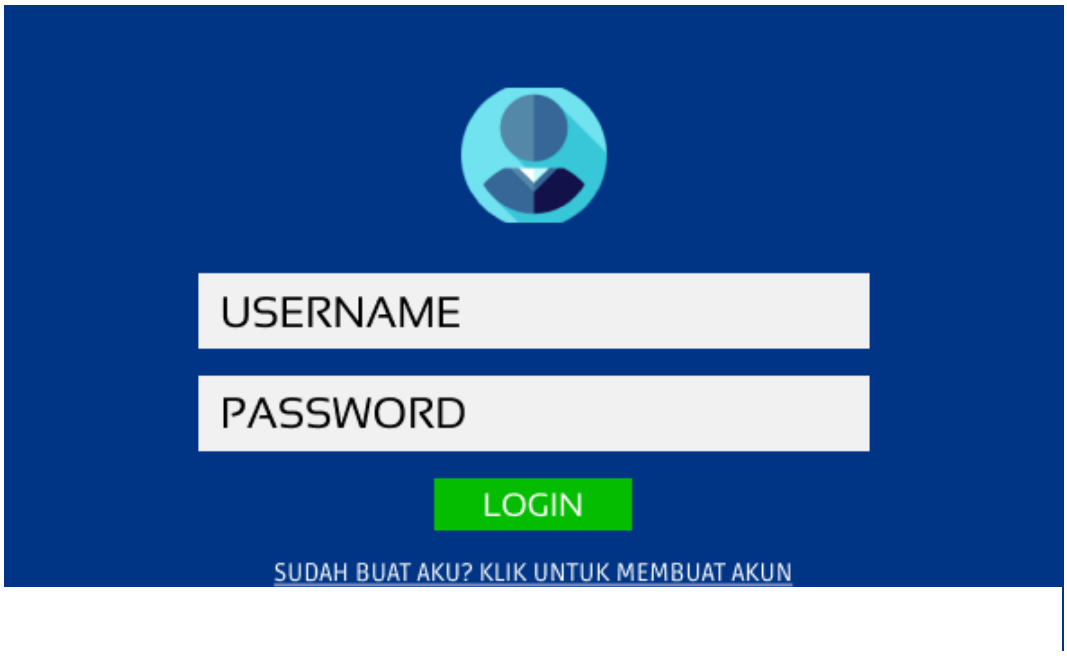
3.3.5.1 Desain Antarmuka

Pada pembuatan desain antarmuka ini menggunakan *tools* figma. Antarmuka sistem dibuat untuk memberikan kemudahan kepada pengguna dalam memahami dan mengoperasikan fungsi – fungsi yang ada pada sistem. Dalam sistem penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* yang akan dibuat, terdapat 1 aktor yang dapat mengakses, yaitu User.

User merupakan seorang aktor yang dapat melakukan penyembunyian pesan pada citra *digital* dengan menggabungkan algoritma *hill cipher* dengan menggunakan menu *encode*. Adapun halaman yang disediakan bagi *User* adalah sebagai berikut

a. Halaman Login

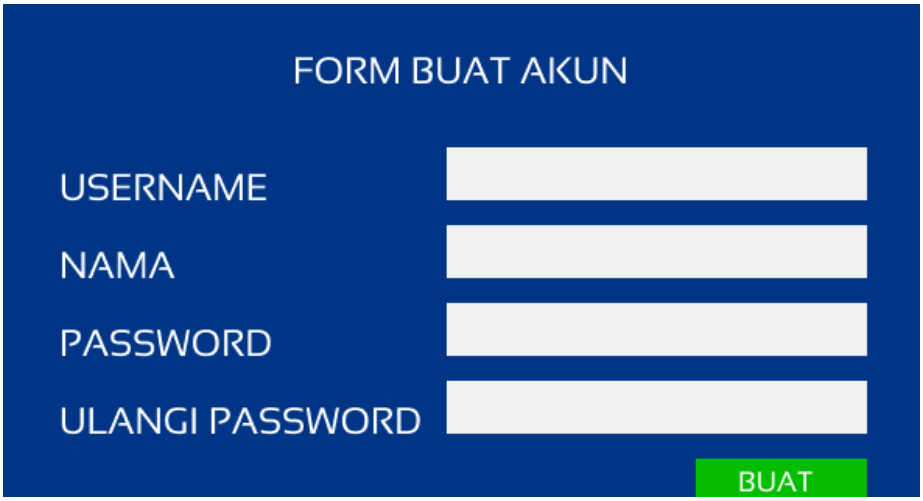
Pada Gambar 3.4 halaman ini merupakan halaman awal sistem ketika diakses.



Gambar 3. 4 Halaman *Login*

b. Halaman Registrasi

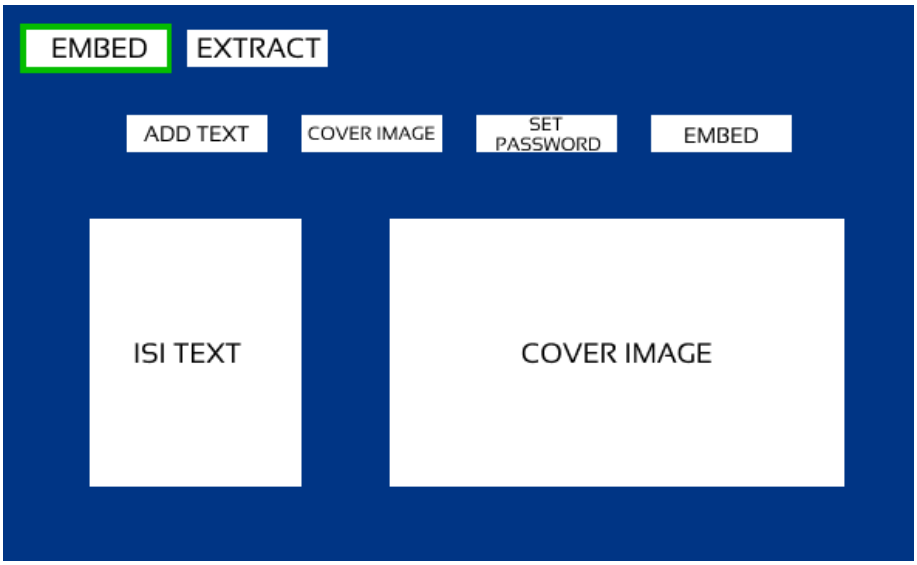
Pada gambar di bawah yaitu Gambar 3.5 halaman ini merupakan langkah yang harus dijalankan oleh pengguna untuk dapat masuk ke dalam aplikasi.



Gambar 3. 5 Halaman registrasi

c. Halaman *Embed*

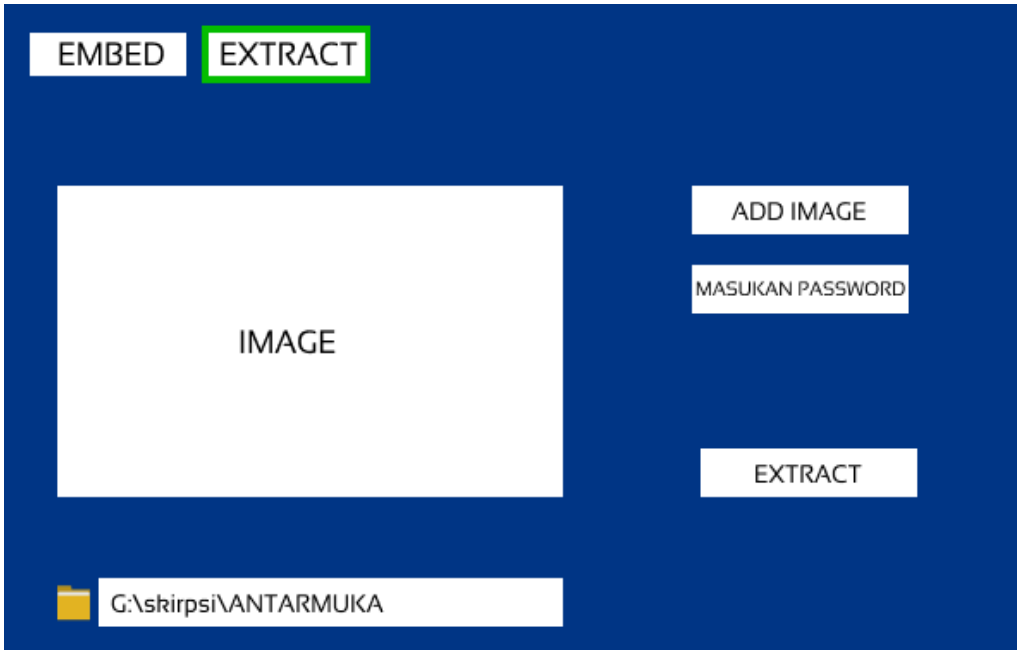
Gambar di bawah ini yaitu Gambar 3.6 merupakan tampilan setelah user melakukan login atau registrasi, halaman ini untuk melakukan proses *add text*, *add cover image*, *set password*, dan *embed*



Gambar 3. 6 Halaman Embed

d. Halaman Ekstrak

Halaman ini merupakan Gambar 3.7 halaman untuk ekstrak dokumen berupa citra yang sudah dilakukan proses *embed*



Gambar 3. 7 Halaman Ekstrak

3.3.6 Pengujian

Pengujian merupakan tahapan setelah proses implementasi selesai. Dalam tahap ini dilakukan testing terhadap aplikasi atau sistem yang telah dibuat. Tahapan pengujian bertujuan sebagai ukuran bahwa sistem dapat berjalan sesuai dengan yang diharapkan. Proses pengujian pada penelitian ini menggunakan cara sebagai berikut :

a. Blackbox

Pengujian *blackbox* adalah untuk mengetahui cara beroperasi perangkat lunak, apakah proses *input* data dan *output* yang dihasilkan telah sesuai yang diharapkan dan apakah informasi yang disimpan selalu dijaga kemutakhirannya.

b. PSNR dan MSE

PSNR digunakan untuk mengetahui perbandingan kualitas citra *cover* sebelum dan sesudah disisipkan pesan. Dalam suatu pengembangan dan pelaksanaan rekonstruksi gambar diperlukan perbandingan antara gambar hasil rekonstruksi dengan gambar asli. Ukuran umum yang digunakan untuk tujuan ini adalah *Peak Signal to Noise Ratio* (PSNR). Nilai PSNR yang lebih tinggi menyiratkan kemiripan yang lebih erat antara hasil rekonstruksi dan gambar asli. PSNR didefinisikan sebagai :

$$PSNR = 10\log_{10}\left(\frac{C^2max}{MSE}\right) \tag{3.1}$$

Dimana MSE dinyatakan sebagai mean square error yang didefinisikan sebagai :

$$MSE = \frac{1}{MN}\sum_{X=}^{M}\sum_{y=}^{N}(S_{xy} - C_{xy})^2 \tag{3.2}$$

Dimana x dan y adalah koordinat dari gambar, M dan N adalah dimensi dari gambar, S_xy menyatakan stego-image dan C_xy menyatakan *cover-image*. C_max^2 memiliki nilai maksimum dalam gambar yaitu nilai maksimum dari nilai piksel adalah 255 dan minimum adalah 1.

3.3.7 Pembuatan Laporan

Pada tahapan kegiatan terakhir ini yaitu hal yang dilakukan adalah melaporkan hasil kegiatan dari penelitian yang telah dilakukan.

3.4 Jadwal Kegiatan

Jadwal penelitian dilakukan selama 6 bulan dengan rincian kegiatan seperti pada tabel berikut.

Tabel 3. 1 Jadwal Kegiatan

No	Keterangan	1	2	3	4	5	6
1	Identifikasi Masalah						
2	Studi Literatur						
3	Pengumpulan data						
4	Analisa						
5	Perancangan						
6	Pengujian						

BAB 4. HASIL DAN PEMBAHASAN

Aplikasi steganografi yang akan dibuat dalam penelitian ini akan diimplementasikan menjadi aplikasi berbasi *website* dengan nama ram-stega. Fungsi utama dari aplikasi ini adalah mengenkripsi sebuah pesan berupa teks lalu menyisipkannya ke dalam citra gambar dan mengekstraknya kembali, pada saat menjalankan aplikasi ini maka akan ditampilkan fitur *encode* dan *decode*, pada fitur *encode* berfungsi untuk mengenkripsi sebuah pesan berupa teks lalu menyisipkan pesan berupa teks ke dalam citra gambar dengan ekstensi PNG dan fitur *decode* berfungsi untuk mengekstrak gambar yang sudah tersisipi pesan teks di dalamnya. *Output* dari aplikasi ini berupa gambar dengan ekstensi png yang didalamnya terdapat sebuah pesan rahasia berupa teks

4.1 Analisa Perangkat Lunak

Aplikasi yang akan dibuat pada penelitian ini adalah aplikasi steganografi dengan nama ram-stega, berikut ini akan dijelaskan tentang deskripsi aplikasi, alur penggunaan aplikasi, kebutuhan data, kelebihan aplikasi, kelemahan aplikasi dan tujuan pengembangan aplikasi.

4.1.1 Deskripsi Aplikasi

Aplikasi steganografi dengan nama ram-stega ini memiliki dua fitur utama, yaitu fitur enkripsi pesan teks menggunakan algoritma *Hill Cipher* lalu menyisipkan pesan teks ke dalam file citra gambar digital dengan menggunakan metode *Least Significant Bit* dan fitur ekstraksi *file* citra gambar digital yang telah disisipi pesan teks. Untuk fitur penyisipan ini adalah sebuah *file* citra gambar digital dengan ekstensi png yang akan disisipkan sebuah data berupa data teks .

Hasil dari ekstraksi file ini adalah melakukan proses ekstraksi ke sebuah *file* citra gambar digital dengan ekstensi png untuk menampilkan data pesan teks yang sudah disisipkan di dalam citra gambar tersebut.

4.1.2 Alur Penggunaan Aplikasi

Aplikasi ram-stega adalah sebuah aplikasi steganografi yang berfungsi sebagai media penyimpanan data berupa teks kedalam sebuah citra gambar dengan penggabungan algoritma *Hill Cipher* dan metode *LSB*. Aplikasi ram-stega memiliki dua fitur utama yang mengimplementasikan dua fungsi, yaitu penyisipan pesan dan ekstraksi pesan.

Penyisipan pesan memiliki fungsi sebagai media untuk menyisipkan pesan berupa pesan teks. Pesan yang disisipkan kemudian akan dilakukan proses penyisipan untuk menghasilkan *file Stego Image*. Sedangkan ekstraksi pesan memiliki fungsi untuk menampilkan pesan yang sudah disisipkan di dalam citra gambar sistem kemudian melakukan proses ekstraksi dan menampilkan pesan dari *file Stego Image*.

4.1.3 Kebutuhan Data

Aplikasi ram-stega yang dibuat pada penelitian ini memerlukan beberapa data sebagai berikut :

a. Citra Digital dengan ekstensi PNG
b. Ukuran atau resolusi gambar
c. Pesan berupa teks

4.1.4 Kelebihan Aplikasi

Aplikasi ram-stega yang dibuat pada penelitian ini diharapkan dapat melakukan hal-hal sebagai berikut :

a. Menyisipkan pesan teks pada citra gambar ekstensi png
b. Dapat melakukan ekstraksi pesan pada citra gambar ekstensi png yang sudah menjadi *file Stego Image*

4.1.5 Kekurangan Aplikasi

Aplikasi ram-stega yang dibuat pada penelitian ini memiliki beberapa kekurangan diantaranya sebagai berikut :

a. Pesan yang disisipkan hanya berupa teks.

b. Citra gambar yang digunakan untuk media penyimpanan pesan dan hasil *Stego Image* adalah ekstensi PNG.
c. Jika terdapat spasi pada proses enkripsi maka spasi akan dihilangkan pada *ciphertext*.
d. Apabila huruf pada pesan yang dimasukkan berjumlah ganjil maka diakhir pesan pada *ciphertext* akan ditambahkan huruf x.

4.1.6 Tujuan Pengembangan Aplikasi

Tujuan dari aplikasi ram-stega ini diharapkan bisa menyisipkan data teks ke dalam citra gambar digital dengan ekstensi PNG. Aplikasi ram-stega juga diharapkan bisa melakukan ekstraksi data teks yang sudah disisipkan ke dalam citra gambar digital ekstensi PNG

4.2 Perancangan Sistem

Pada perancangan aplikasi ini dibutuhkan sistem komputerisasi yang mampu memenuhi kebutuhan tersebut yaitu dengan menggunakan *Visual Studio Code* sebagai media pembuatan aplikasi steganografi tersebut dengan menggabungkan algoritma *Hill Cipher* dan metode *Least Significant Bit*. Diharapkan dapat melakukan hal-hal sebagai berikut :

a. Menyisipkan pesan teks pada citra gambar ekstensi PNG
b. Dapat melakukan ekstraksi pesan pada citra gambar ekstensi PNG yang sudah menjadi *file Stego Image*

4.2.1 *Algoritma Hill Cipher*

Algoritma *Hill Cipher* ini digunakan untuk pengacakan data dengan mengganti setiap huruf yang ada menggunakan perkalian matriks ordo 2 x 2 modular 26. Dengan pengacakan terlebih dahulu akan menghasilkan keamanan ganda yang lebih aman untuk data yang akan disembunyikan. Keuntungan teknik ini adalah tidak dapat mengetahui pesan yang sebenarnya jika tidak memiliki *key* dari pesan tersebut. Teknik ini sering digunakan untuk menghindari pencurian data yang dilakukan oleh orang yang kurang bertanggung jawab.

4.2.2 Metode Least Significant Bit

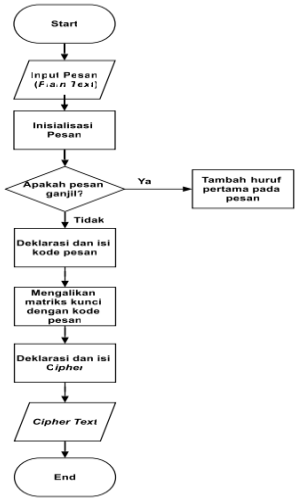
Metode *Least Significant Bit* ini digunakan untuk penyembunyian data dengan mengganti bit-bit di dalam segmen citra (*image*) dengan bit-bit pesan. Bit yang akan diganti adalah bit *LSB* yaitu 8 bit terakhir citra , karena penggantian hanya mengubah nilai byte tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan byte tersebut di dalam gambar menyatakan warna tertentu, maka perubahan satu bit *LSB* tidak mengubah warna tersebut secara signifikan. Keuntungan teknik ini perubahan yang ada tidak terlihat oleh mata manusia. Teknik ini sering digunakan untuk menghindari kecurigaan orang dan menghindari keinginan orang untuk mengetahui isi pesan rahasia tersebut.

4.2.3 Perancangan Flowchart

Perancangan *flowchart* pada aplikasi steganografi ini dengan nama ram-stega terbagi dua yaitu *flowchart* penyisipan pesan dan *flowchart* ekstraksi pesan adalah sebagai berikut :

1. Flowchart Penyisipan Pesan

   Flowchart penyisipan pesan merupakan suatu cara menggambarkan algoritma dalam pengacakan data lalu menyisipkan data kedalam media gambar citra digital



Gambar 4. 1 *Flowchart* Pengacakan Pesan

Gambar 4. 2 *Flowchart* Penyisipan Pesan

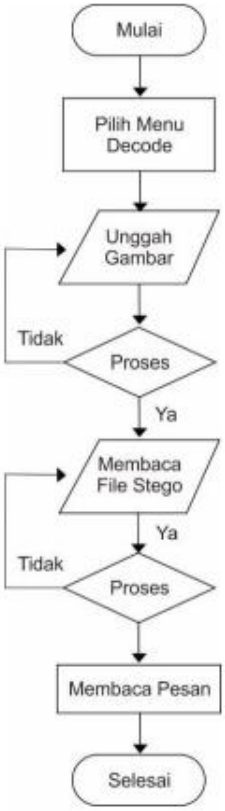
Berikut ini adalah penjelasan Diagram alir (*flowchart*) dari proses penyisipan pesan :

1. Dimulai dari tampilan awal ada dua menu utama yaitu *encode* dan *decode*.
2. Menu *encode* berfungsi untuk mengacak pesan agar tidak terlihat isi asli dari pesan dan penyembunyian pesan ke dalam citra gambar.
3. Setelah masuk ke menu *encode* akan ditampilkan beberapa fitur diantaranya *form* pesan yang akan diacak, hasil acak pesan, *key* dari pesan tersebut, dan tombol *cipher*.
4. Kemudian masukkan pesan yang akan dicak pada form input.
5. Lalu masukkan key yang akan digunakan sebagai perkalian matriks ordo 2 x 2 modular 26.
6. Jika berhasil maka akan muncul pesan hasil enkripsi dalam *form output*
7. Salin pesan dalam form output lalu tekan tombol sembunyikan pesan anda.
8. Setelah itu pengguna akan dialihkan ke halaman *embed*

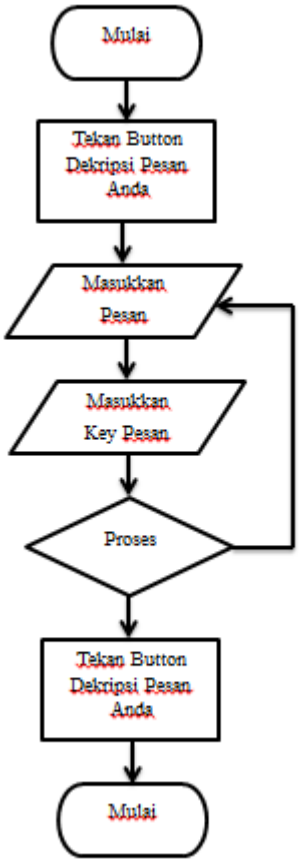
9. Pada halaman embed terdapat form untuk memasukkan gambar yang akan dijadikan media penyembunyian pesan dan isi pesan yang akan disembunyikan
10. Temple pesan yang sebelumnya sudah melalui proses pengacakan ke dalam *form* masukkan pesan dan pilih gambar yang akan digunakan sebagai media penyembunyian dengan ekstensi PNG.
11. Jika pesan sudah terinput selanjutnya simpan gambar agar menjadi *file Stego Image*.

2. Flowchart Ekstraksi Pesan

Flowchart ekstraksi pesan merupakan suatu cara menggambarkan algoritma ekstraksi data sehingga diperoleh data pesan yang telah disisipkan.



Gambar 4. 3 *Flowchart* Ekstraksi Pesan Pada Gambar



Gambar 4. 4 *Flowchart* Ekstraksi Pesan *Hill Cipher*

Berikut ini adalah penjelasan Diagram alir (*flowchart*) dari proses ekstraksi pesan :

1. Dimulai dari tampilan awal ada dua menu utama yaitu menu *encode* dan menu *decode.*
2. Setelah masuk ke menu *decode* akan diarahkan untuk mengunggah citra gambar.
3. Kemudian jika citra gambar tersebut bukan *file Stego Image* maka aka ada *pop up* tidak ada pesan rahasia. Tapi jika *file* tersebut adalah *Stego Image* maka akan secara otomatis menampilkan pesan yang ada di dalam citra gambar tersebut.
4. Lalu setelah berhasil memperoleh pesan yang telah diekstraksi, salin pesan tersebut kemudian klik tombol dekripsi pesan anda.

5. Kemudian masukkan pesan tersebut kedalam *form input* dan masukkan *key* yang sama dengan yang digunakan pada saat enkripsi pesan
6. Jika benar maka akan muncul pesan yang telah diekstraksi pada *form output*,jika huruf pesan berjumlah ganjil maka akan ada tambahan huruf x di akhir kalimat terakhir.

4.3 Perhitungan Manual

4.3.1 Perhitungan Manual Untuk Proses Encode

Perhitungan yang digunakan pada saat proses *hill cipher* dengan mengganti huruf menjadi angka yang telah tersedia dalam tabel lalu dilakukan proses perkalian matriks ordo 2x2 modular 26.

Pada gambar di bawah yaitu Gambar 4.5 adalah tabel yang digunakan untuk mengkonversi huruf ke angka.

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	26

Gambar 4. 5 Tabel Angka *Cipher* yang Digunakan Untuk Pengganti Huruf

Gambar di bawah ini yaitu Gambar 4.6 menggunakan kalimat ‘rama’ sebagai contoh pesan yang akan dienkripsi menggunakan algoritma *Hill Cipher*.

Plainteks: rama		
Matriks kunci:	3	4
	19	11

Gambar 4. 6 *Plaintext* dan Matriks Kunci

Dapat dilihat pada Gambar 4.7 adalah proses pengelompokkan plainteks menjadi matriks 2x2.

Mengelompokkan plainteks menjadi matriks 2.2				
p1=	r		p2=	m
	a			a

Gambar 4. 7 Pengelompokkan *Plaintext* Untuk Perkalian matriks 2x2

Dapat dilihat pada Gambar 4.8 melakukan proses penggantian huruf menjadi angka sesuai dengan tabel yang telah dibuat sebelumnya.

mengganti huruf dengan nomor sesuai tabel sebelumnya				
p1=	17		p2=	12
	0			0

Gambar 4. 8 Mengganti Huruf Menjadi Angka Berdasarkan Tabel *Cipher*

Gambar di bawah yaitu Gambar 4.9 adalah proses perkalian matriks 2x2 modular 26 sebagai rumus untuk melakukan proses enkripsi algoritma *Hill Cipher*.

Perkalian (Ci = ([Pi*[Ki])Mod26)														
p1=	17	.	3	4	mod	26	=	51	mod	26	=	25
	0		19	11				323				11
p2=	12		3	4	mod	26	=	36	mod	26	=	10
	0		19	11				228				20

Gambar 4. 9 Perhitungan Dilakukan Agar Menemukan Hasil Dari *Ciphertext*

Pada gambar di bawah Yaitu Gambar 4.10 akan menunjukan hasil dari perkalian sebelumnya dan akan mendapatkan nilai cipherteks, ini adalah hasil dari proses enkripsi *Hill Cipher* yang sebelumnya kalimat rama menjadi zlku.

Ciphertext				
=	25	11	10	20
=	z	l	k	u

Gambar 4. 10 Hasil Dari *Ciphertext* yang Sudah Didapat

4.3.2 Perhitungan Manual Untuk Proses Decode

Perhitungan yang digunakan pada saat proses hill cipher dengan mengganti huruf menjadi angka yang telah tersedia dalam tabel lalu dilakukan proses perkalian matriks ordo 2x2 modular 26.

Pada gambar di bawah yaitu Gambar 4.11 adalah hasil dari *ciphertext* sebelumnya yang awalnya kalimat rama menjadi zlku setelah melalui proses enkripsi.

Ciphertext				
=	25	11	10	20
=	z	l	k	u

Gambar 4. 11 Ciphertext yang akan didekrip

Pada gambar di bawah yaitu Gambar 4.12 untuk mencari nilai determinan dari matriks kunci yang digunakan untuk enkripsi.

Mencari Nilai Determinan matriks kunci (K)

3	4	=	33	-	76
19	11	=	-43		

Gambar 4. 12 Determinan Dari Nilai Matriks Kunci

Pada gambar di bawah yaitu Gambar 4.13 menunjukan proses untuk dekripsi pesan. Pada gambar di bawa setelah nilai determinan ditemukan maka selanjutnya adalah mencari nilai invers dari matriks kunci yang digunakan untuk enkripsi.

Menghitung invers matriks kunci (K)

K^{-1}=1/det(k) mod 26

$$K^{-1} = 3 * \begin{bmatrix} 11 & -4 \\ -19 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} 33 & -12 \\ -57 & 9 \end{bmatrix} \text{ mod } 26$$

$$= \begin{bmatrix} 7 & 14 \\ 21 & 9 \end{bmatrix}$$

Gambar 4. 13 Invers Matriks Kunci

Pada gambar di bawah yaitu Gambar 4.14 menunjukan setelah mendapatkan hasil dari invers matriks kunci, langkah selanjutnya adalah melakukan perkalian antara *ciphertext* dengan matriks kunci yang telah melalui proses invers. Setelah melakukan perkalian dan mendapatkan hasilnya, maka hasil tersebut dapat dikonversi menjadi huruf sesuai dengan tabel *cipher*. Hasil dari Gambar 4.14 Adalah 17, 0, 12, 0 yang bila dikonversi menjadi rama, sekarang proses dekripsi telah berhasil dilakukan.

Proses dekripsi											
k^{-1}	=	7	14								
		21	9								
Ciphertext	=	25	11	10	20						
	=	z	l	k	u						
c1	=	25		7	14	329	mod	26	=	17	
		11		21	9	624				0	
c2	=	10		7	14	350	mod	26	=	12	
		20		21	9	390				0	
Plaintext	=	17	0	12	0						
		r	a	m	a						

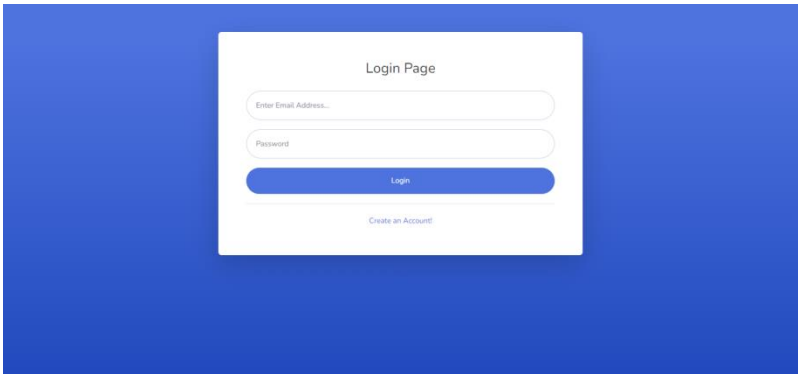
Gambar 4. 14 Proses Perhitungan Untuk *Decrypt* Pesan

4.4 Implementasi

4.4.1 Interface

1. *Login*

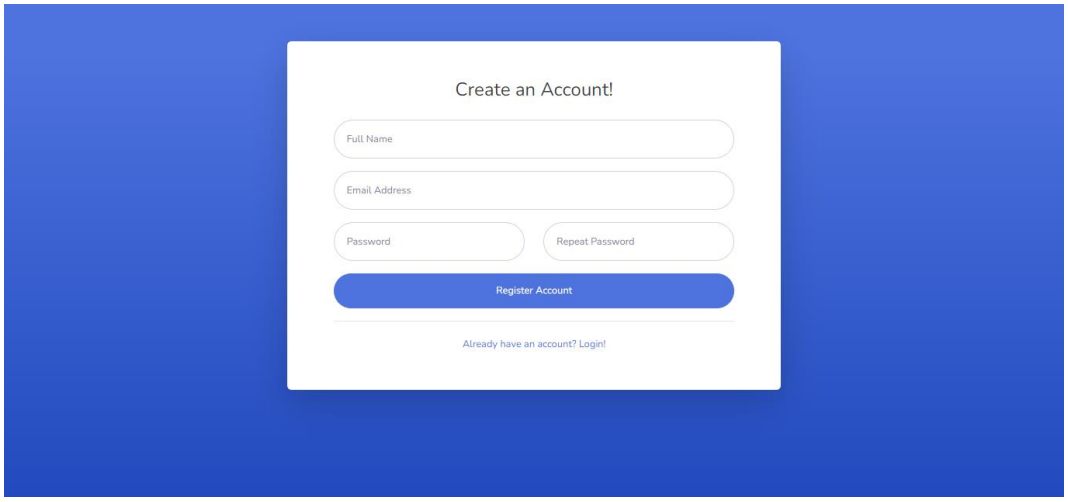
Pada gambar 4.15 halaman login adalah tampilan awal sebelum pengguna masuk ke dalam *website*.



Gambar 4. 15 Fitur *Login Website*

2. Registrasi

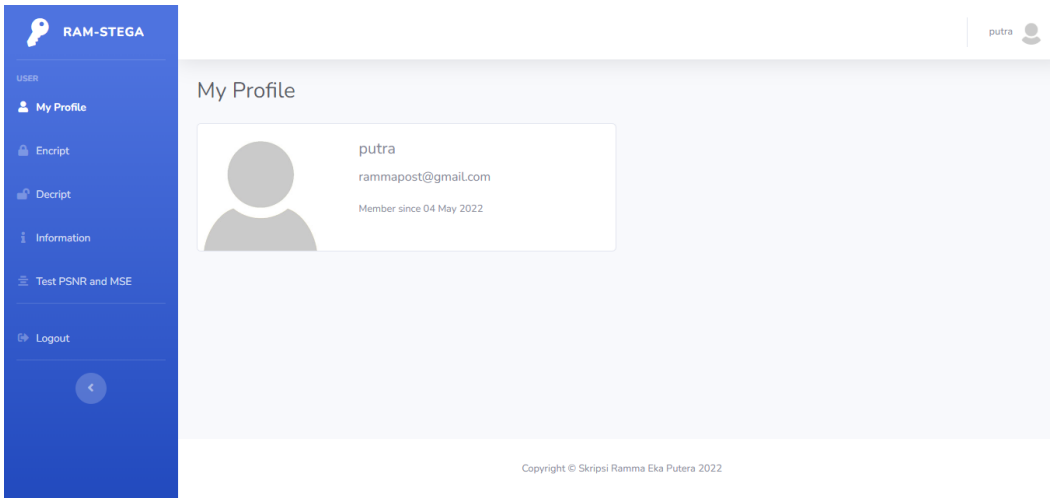
Pada Gambar 4.16 pengguna harus terlebih dahulu melakukan registrasi jika belum memiliki akun agar dapat masuk ke dalam *website*.



Gambar 4. 16 Fitur Registrasi pada *Website*

3. Dashboard

Dashboard adalah tampilan awal dari aplikasi , *Dashboard* terdiri dari fitur *encode*, *decode*, pengujian, informasi dan *PSNR*. Dapat dilihat pada gambar 4.17

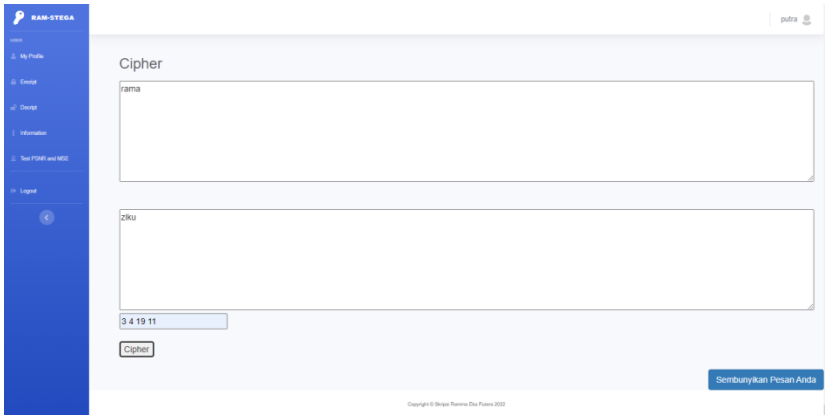


Gambar 4. 17 Tampilan *Dashboard* Saat Pengguna Berhasil *Login*

4. Menu Encode

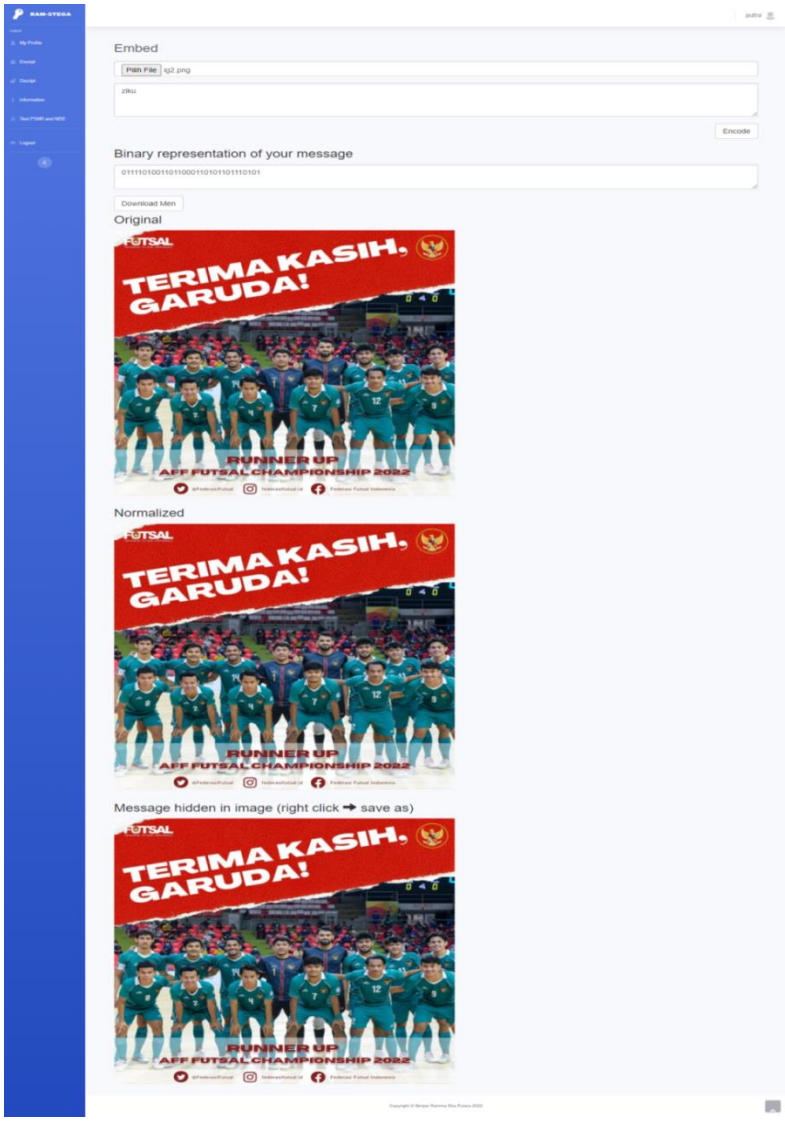
Menu *encode* adalah fitur pengacakan pesan dengan menggunakan algoritma *hill cipher*. Pengguna harus memasukkan pesan yang akan dienkripsi lalu memasukkan key dengan jumlah empat angka (contoh: 2 3 19 11). Apabila key yang dimasukkan tidak sesuai, akan terjadi gagal dalam proses dekrip dikarenakan *key* berfungsi sebagai perkalian matriks, *key* yang dimasukkan dapat digunakan pada proses enkripsi, namun akan tidak dapat digunakan dalam proses dekripsi apabila *key* tersebut tidak dapat diproses menggunakan invers pada proses dekripsi. Jika sudah maka akan muncul pesan yang telah dienkripsi dan dapat disalin sebelum melanjutkan menekan tombol sembunyikan pesan anda. Dapat dilihat pada Gambar 4.18 dan 4.19.

Gambar 4. 18 Tampilan Menu *Encode* Saat Belum Melalui Proses *Cipher*



Gambar 4. 19 Tampilan Menu *Encode* Saat Sudah Melalui Proses *Cipher*

Dapat dilihat pada Gambar 4.20 setelah proses *cipher* berhasil, pengguna dapat menekan tombol sembunyikan pesan anda, lalu pengguna dapat memasukkan gambar beserta pesan yang telah disalin sebelumnya, akan terlihat bilangan biner dari tiap huruf yang telah dimasukkan dan gambar *original*, *normalized*, dan *message hidden in image*. Setelah melakukan semua proses *encode* pengguna dapat mengunduh gambar yang telah diberi pesan di dalamnya dengan menekan *button download.*

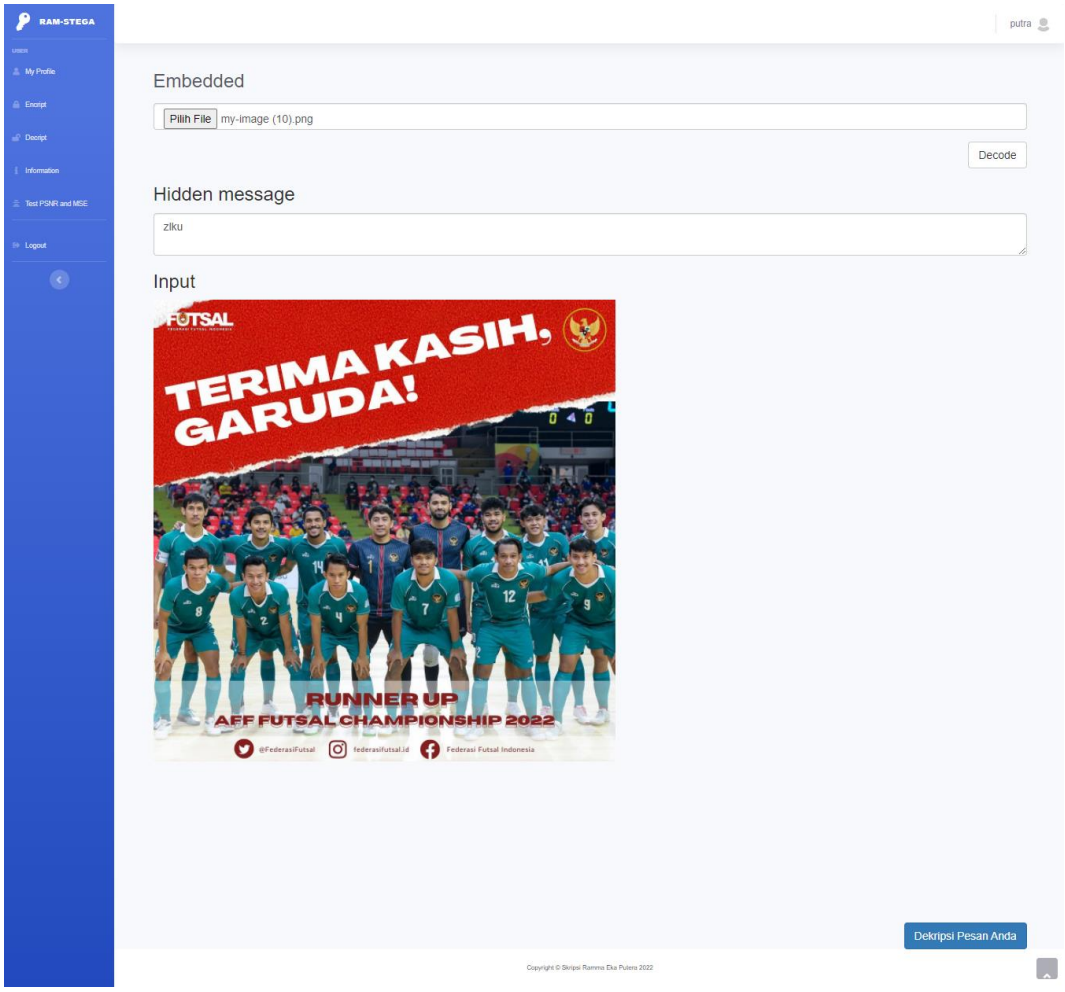


Gambar 4. 20 Tampilan Menu Encode Untuk Menyisipkan Pesan ke Dalam Gambar

5. Menu Decode

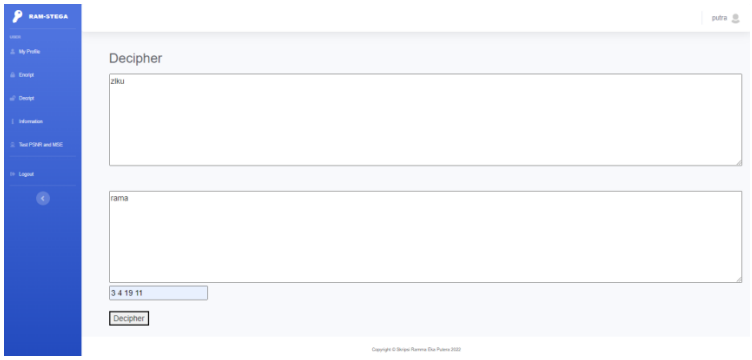
Menu *decode* bertujuan untuk mengekstrak pesan dalam gambar, pada menu ini terdapat 2 proses yaitu ekstrak pesan dari dalam gambar dan ekstrak pesan yang sebelumnya teracak menggunakan *hill cipher* sehingga menjadi sebuah kalimat.

Pada gambar 4.21 pengguna memasukkan gambar yang didalamnya terdapat pesan tersembunyi, jika gambar tersebut memiliki pesan tersembunyi makan akan terlihat pesan di dalamnya.



Gambar 4. 21 Tampilan Menu *Decode* Untuk Ekstraksi Pesan Yang Ada Dalam Sebuah Gambar

Kemudian pengguna dapat melanjutkan untuk menjadi huruf acak tersebut menjadi sebuah kalimat dengan menekan tombol dekripsi pesan anda. Dan setelah itu pengguna memasukan pesan tersebut kedalam *form input* dan memasukkan *key* yang sama saat melakukan proses *encode*, jika berhasil maka akan muncul sebuah kalimat yang sebelumnya berupa huruf acak. Jika isi pesan yang dimasukkan oleh pengguna maka pada proses dekrip akan ditambahkan huruf “x” di akhir pesan. Dapat dilihat pada Gambar 4.22



Gambar 4. 22 Tampilan Menu *Decode* Untuk Ekstraksi Huruf Acak Menjadi Sebuah Kalimat

6. Menu Informasi

Pada gambar di bawah yaitu Gambar 4.23 dalam menu informasi terdapat panduan untuk melakukan proses *encode* dan *decode*, mulai dari mengisi *form* pesan sampai pada ekstraksi pesan.

Gambar 4. 23 Tampilan Menu Informasi Pada *Website*

7. Menu Test PSNR and MSE

Menu Test *PSNR* and *MSE* berfungsi sebagai memperbandingkan kedua citra yaitu citra asli dan juga citra steganografi dimana dari hasil perbandingan tersebut mendapatkan nilai dari *PSNR* dan *MSE* yang menentukan kualitas citra sebelum dan sesudah disisipi data pesan. Pada kasus ini penulis menggunakan gambar dengan nama “ig2.png” sebagai *cover image* dan gambar “my-image(11).png” sebagai *stego image*.

Pada tabel di bawah ini yaitu Tabel 4.1 dapat dilihat nilai nilai piksel *red*, *green*, dan *blue* dan. Kemudian pada Tabel 4.2 adalah pengujian nilai *MSE* dengan gambar yang belum melalui proses steganografi dan yang sudah. Dapat dilihat pada Tabel 4.3 di bawah nilai pengujian *PSNR* yang didapat.

Tabel 4. 1 Jumlah Piksel RGB Pada Gambar "ig2.png"

Cover Image	Pixel Red	Pixel Green	Pixel Blue
ig2.png	244	244	242

Tabel 4. 2 Selisih Piksel Pada Gambar "ig2.png" Setelah Melalui proses MSE

Stego Image	Pixel Red	Pixel Green	Pixel Blue
my-image(11).png	0.50193139835563	0.50381200862527	0.50457237882688

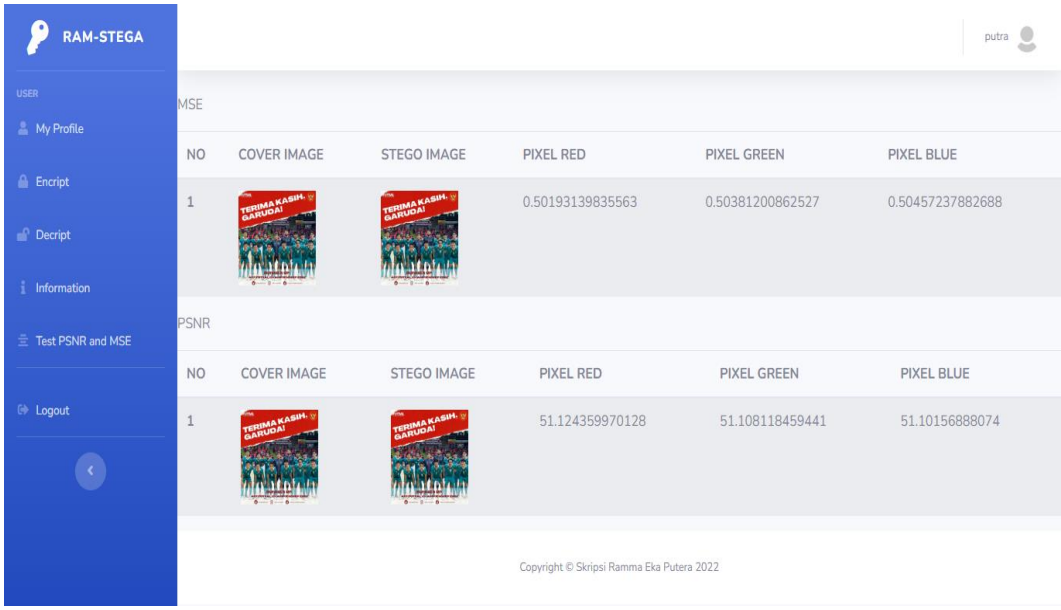
Pada Tabel 4.2 di atas dapat dilihat selisih yang dihasilkan dari piksel RGB terdapat selisih yang nilainya kurang dari satu dengan selisih piksel *red* 0.50193139835563, piksel *green* 0.50381200862527, dan piksel *blue* 0.50457237882688. yang artinya secara kasat mata tidak akan terlihat perbedaan pada *cover image* dan *stego image* karena selisih piksel RGB dari pengujian *MSE* tidak melebihi satu.

Tabel 4. 3 Nilai PSNR yang Didapatkan Pada Gambar "ig2.png"

Stego Image	Pixel Red	Pixel Green	Pixel Blue
my-image(11).png	51.124359970128	51.108118459441	51.10156888074

Pada Tabel 4.3 di atas dapat dilihat rata-rata nilai *PSNR* yang didapatkan dari hasil pengujian gambar “my-image(11).png” ini menunjukan *stego image* sudah cukup baik karena semakin besar nilai *PSNR* akan semakin baik.(Aprilia et al., 2019).

Pada gambar di bawah ini yaitu Gambar 4.24 adalah tampilan pada *website* untuk pengujian nilai *MSE* dan rata-rata *PSNR* yang didapat dari *sample* gambar “ig2.png” sebagai cover image dan gambar “my-image(11).png” sebagai *stego image*.



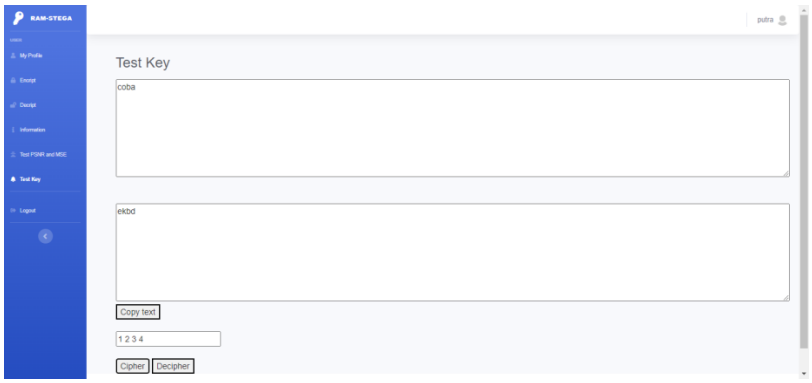
Gambar 4. 24 Hasil Dari Tes MSE dan PSNR

8. Menu Test Key

Menu *test key* adalah fitur yang bertujuan untuk menguji apakah *key* yang digunakan oleh pengguna sudah memenuhi syarat agar dapat dilakukan proses

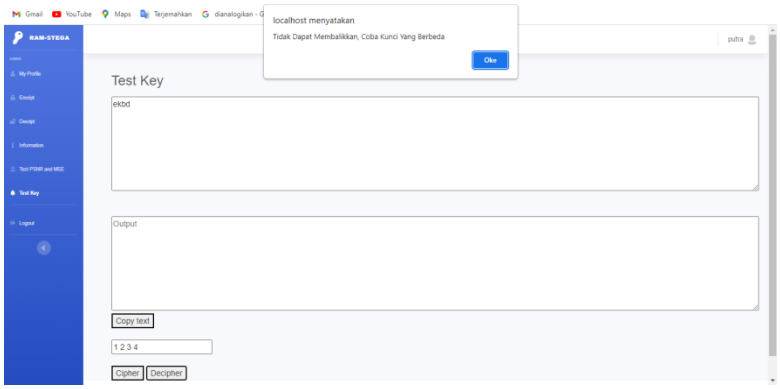
perhitungan matriks invers yang berguna sebagai perkalian dalam proses enkripsi dan dekripsi. Di sini pengguna dapat melakukan percobaan dengan memasukkan pesan acak (contoh: coba) lalu memasukkan key yang berjumlah 4 bilangan bulat positif.

Dapat dilihat pada Gambar 4.25 di bawah menunjukan proses *cipher* berhasil dengan menggunakan *key* 1, 2, 3, dan 4.



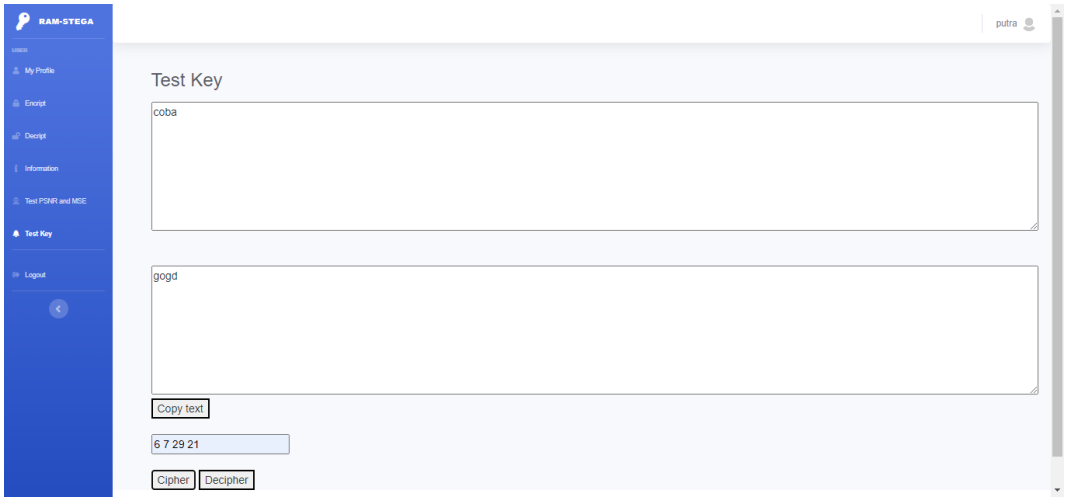
Gambar 4. 25 Proses *Cipher* Berhasil

Dapat dilihat pada Gambar 4.26 di bawah menunjukan proses *decipher* gagal dengan menggunakan key 1, 2, 3, dan 4. Karena *key* yang digunakan tidak dapat dilakukan proses invers pada *decipher*. Maka *key* yang digunakan tidak dapat digunakan dalam proses dekripsi.



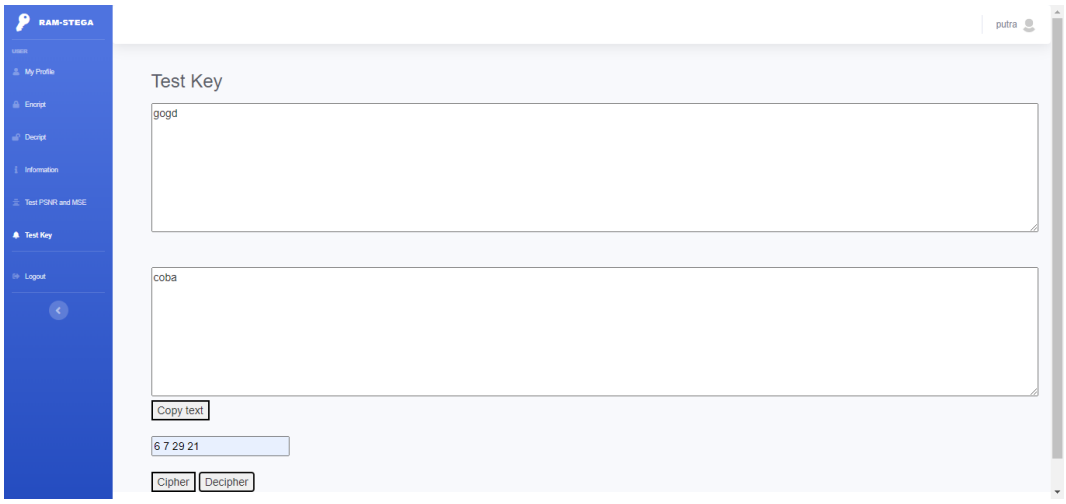
Gambar 4. 26 Proses *Decipher* Gagal

Dapat dilihat pada Gambar 4.27 di bawah menunjukan proses *cipher* berhasil dengan menggunakan key 6, 7, 29, dan 21.



Gambar 4. 27 Proses *Cipher* Berhasil

Dapat dilihat pada Gambar 4.28 di bawah menunjukan proses *decipher* berhasil dengan menggunakan key 6, 7, 29, dan 21. Karena key yang digunakan dapat dilakukan proses invers pada *decipher*. Maka *key* yang digunakan dapat digunakan dalam proses dekripsi.



Gambar 4. 28 Proses *Decipher* Berhasil

4.5 Pengujian

Pengujian merupakan proses untuk menentukan aplikasi tersebut berjalan sesuai kebutuhan. Pengujian sering diasosiasikan dengan pencarian beberapa masalah seperti tidak berfungsinya beberapa fitur , ketidaksempurnaan aplikasi, kesalahan pada kode program yang menyebabkan kegagalan pada eksekusi aplikasi tersebut. Pengujian dilakukan dengan menguji setiap fitur yang ada di dalam aplikasi tersebut.

Pada penelitian ini penulis menggunakan metode *Black box*, yaitu menguji aplikasi dari setiap spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian ini dilakukan untuk mengetahui apakah fungsi fungsi, untuk inputan dan juga hasil keluaran dari aplikasi sesuai dengan

spesifikasi yang dibutuhkan :

a. Menyiapkan *image* untuk sebagai media uji
b. Mengakses aplikasi ram-stega yang sudah dibuat.
c. Melakukan pengujian.
d. Mencatat hasil pengujian.

4.5.1 Pengujian Menu Login

Tabel 4. 4 Tabel Pengujian Menu *Login* Menggunakan Metode *BlackBox* Testing

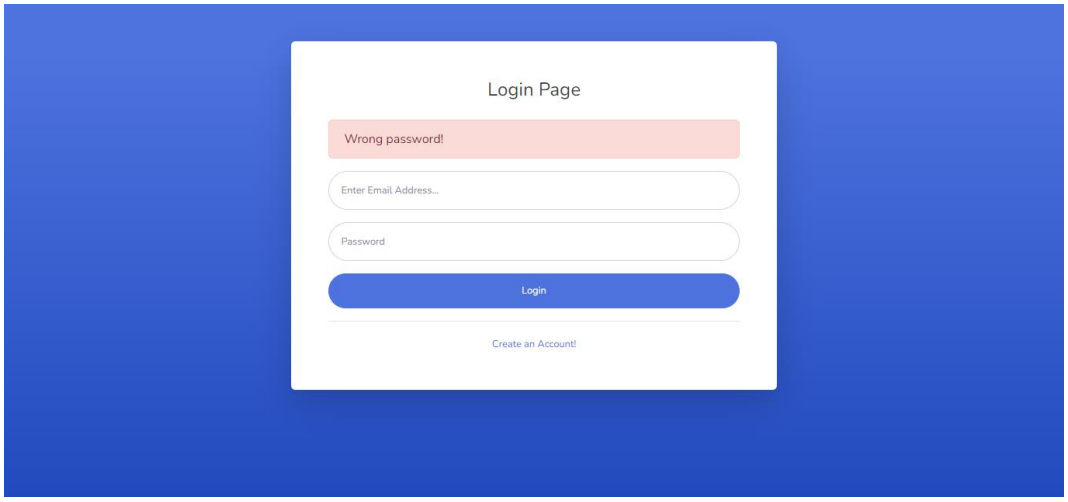
No	Fungsi	Kebutuhan	Hasil
1	Ketika menekan tombol *login* tetapi *password* yang dimasukkan salah	Menampilkan keterangan *password* yang dimasukkan oleh pengguna salah dan tidak dapat melakukan proses *login*	Sukses

Lanjutan Tabel 4.4 Pengujian Menu *Login* Menggunakan Metode *BlackBox* Testing

2	Ketika menekan tombol *login* tetapi *email* yang dimasukkan tidak sesuai	Menampilkan keterangan *email* yang dimasukkan oleh pengguna salah dan tidak dapat melakukan proses *login*	Sukses
3	Ketika menekan tombol *login* tetapi *email* dan *password* belum diisi oleh pengguna	Menampilkan keterangan *email* dan *password* belum dimasukkan oleh pengguna dan tidak dapat melakukan proses *login*	Sukses
4	Ketika menekan tombol *login* dengan memasukkan email dan *password* yang benar dan sudah terdaftar sebelumnya	Pengguna akan dialihkan ke halaman *dashboard* setelah berhasil melakukan *login*	Sukses

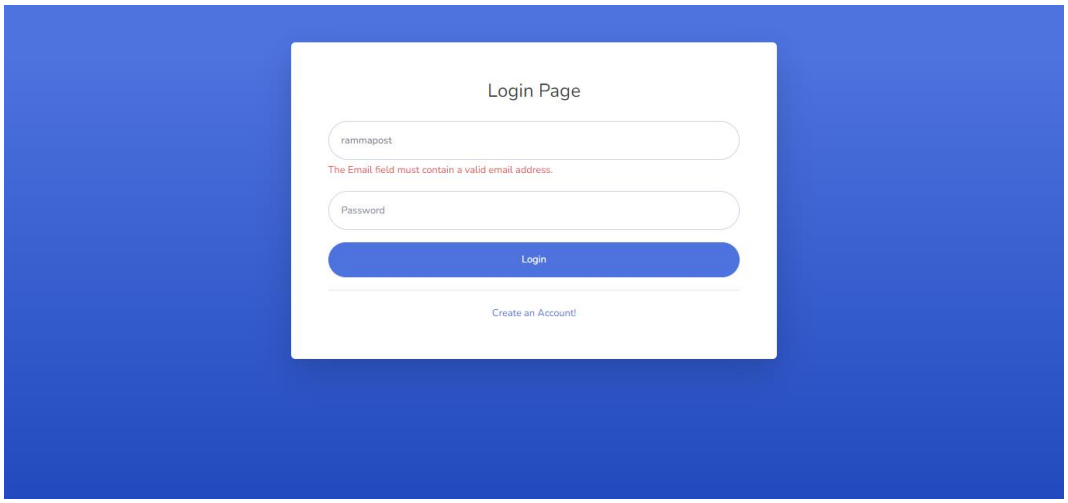
Pada pengujian di Table 4.4 pengujian menu *login* menampilkan fungsi jika *login* berhasil dan gagal serta menampilkan keterangan ketika *login* gagal.

Pada gambar di bawah ini yaitu Gambar 4.29 terjadi kegagalan *login* apabila *password* yang dimasukkan oleh pengguna salah.



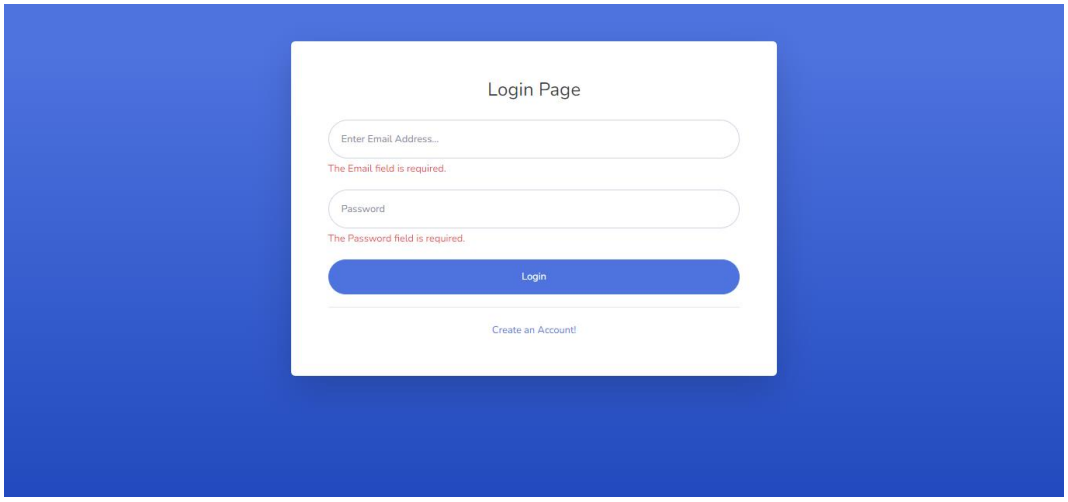
Gambar 4. 29 Proses *Login* Gagal Dengan Keterangan *Password* Salah

Pada gambar di bawah ini yaitu Gambar 4.30 terjadi kegagalan *login* apabila pengguna memasukkan *email* yang tidak sesuai atau salah.



Gambar 4. 30 Proses *login* gagal dengan keterangan email yang dimasukkan salah

Pada gambar di bawah ini yaitu Gambar 4.31 terjadi kegagalan *login* apabila pengguna tidak memasukkan *email* dan *password*.



Gambar 4. 31 Poses *login* gagal dengan keterangan *email* dan *password* harus diisi

4.5.2 Pengujian Menu Registrasi

Tabel 4. 5 Pengujian Menu Registrasi

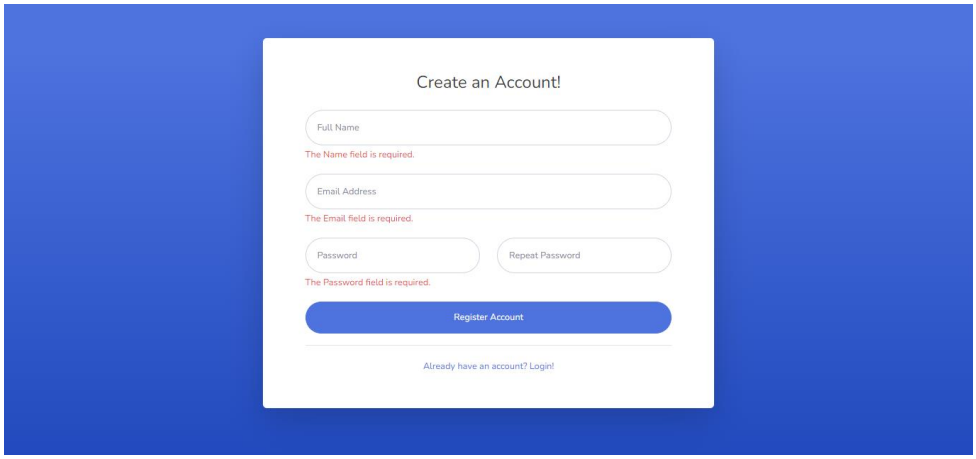
No	Fungsi	Kebutuhan	Hasil
1	Ketika menekan tombol registrasi namun calon pengguna belum memasukkan data apapun	Akan muncul keterangan pengguna belum memasukkan data apapun saat registrasi dan gagal melakukan registrasi	sukses
2	Ketika menekan tombol registrasi namun calon pengguna belum memasukkan nama lengkap	Akan muncul keterangan pengguna belum memasukkan nama lengkap saat registrasi dan gagal melakukan registrasi	sukses

Lanjutan Tabel 4.5 Pengujian Menu Registrasi

3	Ketika menekan tombol registrasi namun calon pengguna belum memasukkan *email*	Akan muncul keterangan pengguna belum memasukkan *email* saat registrasi dan gagal melakukan registrasi	sukses
4	Ketika menekan tombol registrasi namun calon pengguna belum memasukkan *password*	Akan muncul keterangan pengguna belum memasukkan *password* saat registrasi dan gagal melakukan registrasi	sukses
5	Ketika menekan tombol registrasi dan calon pengguna sudah memasukkan semua data yang harus dimasukkan	Pengguna akan langsung dialihkan ke halaman *login* setelah berhasil melakukan proses registrasi	sukses

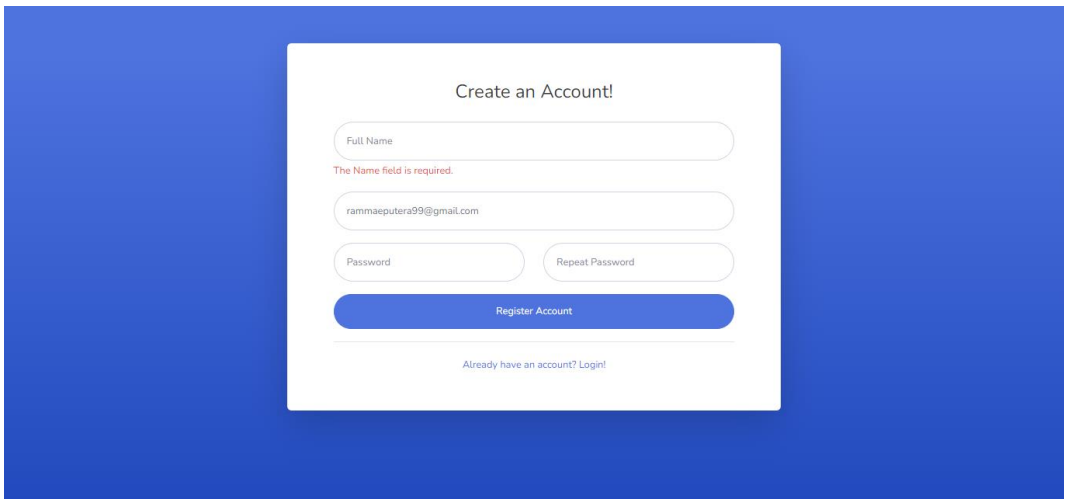
Pada pengujian di Table 4.5 pengujian menu registrasi pengujian fungsi tombol registrasi dan jika data yang dimasukkan salah atau bahkan belum diisi maka akan muncul peringatan, lalu proses registrasi berhasil pengguna akan dialihkan ke halaman *login*.

Pada gambar di bawah ini yaitu Gambar 4.32 Proses registrasi gagal karena calon pengguna belum memasukkan data apapun.



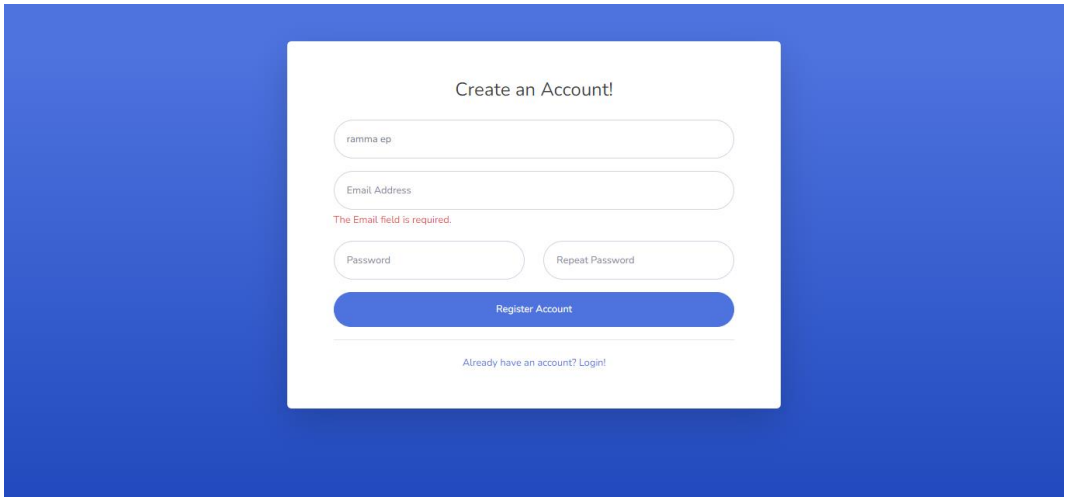
Gambar 4. 32 Proses Registrasi Gagal Dengan Keterangan Harus Mengisi Semua Data Pada Form Yang Disediakan

Pada gambar di bawah ini yaitu Gambar 4.33 proses registrasi gagal karena calon pengguna belum mengisi nama lengkap pada *form* yang telah disediakan.



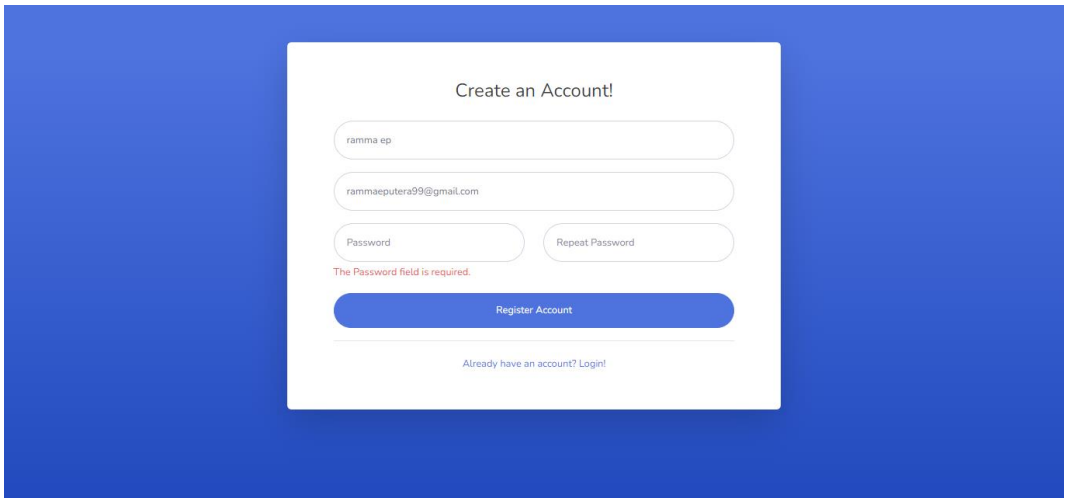
Gambar 4. 33 Proses Registrasi Gagal Dengan Keterangan Nama Lengkap Belum Diisikan

Pada gambar di bawah ini yaitu Gambar 4.34 proses registrasi gagal karena calon pengguna belum mengisi *email* pada *form* yang telah disediakan.



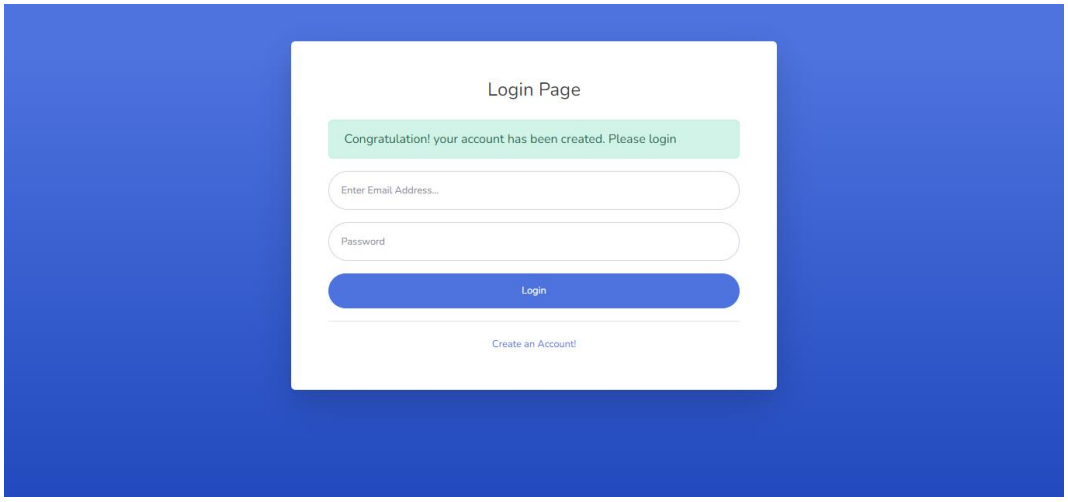
Gambar 4. 34 Proses Registrasi Gagal Dengan Keterangan *Email* Belum Diisikan

Pada gambar di bawah ini yaitu Gambar 4.35 proses registrasi gagal karena calon pengguna belum mengisi password pada form yang telah disediakan.



Gambar 4. 35 Proses Registrasi Gagal Dengan Keterangan *Password* Belum Diisikan

Pada gambar di bawah ini yaitu Gambar 4.36 proses registrasi berhasil dan pengguna akan langsung diarahkan ke halaman *login*.



Gambar 4. 36 Proses Registrasi Berhasil Dan Pengguna kan Diarahkan Ke Halaman *Login*

4.5.3 Pengujian Menu Encode

Tabel 4. 6 Pengujian Menu *Encode*

NO	Fungsi	Kebutuhan	Hasil
1	Ketika menekan tombol sembunyikan pesan anda namun pengguna belum memasukkan data apapun	Menampilkan peringatan pengguna harus mengisi data terlebih dahulu	Sukses
2	Ketika menekan tombol sembunyikan pesan anda namun pengguna belum memasukkan kunci untuk *cipher*	Menampilkan peringatan pengguna harus memasukan kunci cipher terlebih dahulu	Sukses

Lanjutan Tabel 4.6 Pengujian Menu *Encode*

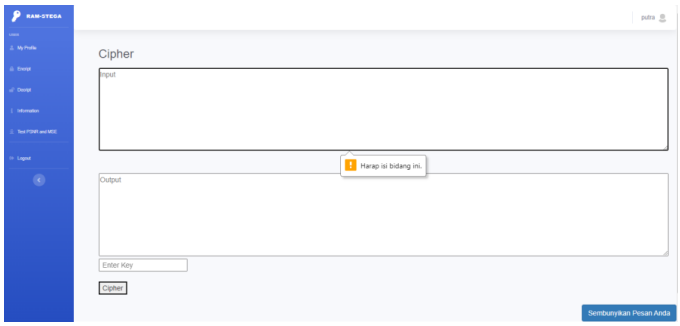
3	Ketika menekan tombol sembunyikan pesan anda dan pengguna sudah memasukkan pesan dan kunci untuk *cipher*	Menampilkan pesan hasil *cipher* dan dapat dilanjutkan ke halaman *extract*	Sukses
4	Ketika menekan tombol *encode* namun pengguna belum memasukkan gambar sebagai media penyisipan pesan	Menampilkan peringatan pengguna belum memasukkan gambar untuk media penyisipan pesan	Sukses
5	Ketika menekan tombol *encode* namun pengguna belum memasukkan gambar dan pesan sebagai media penyisipan pesan	Menampilkan peringatan pengguna belum memasukkan gambar dan pesan untuk media penyisipan pesan	Sukses
6	Ketika menekan tombol *encode* namun pengguna belum memasukkan pesan	Menampilkan peringatan pengguna belum memasukkan pesan	Sukses

Lanjutan Tabel 4.6 Pengujian Menu *Encode*

7	Ketika menekan tombol *encode* dan pengguna sudah memasukkan pesan dan gambar	Menampilkan gambar yang sudah terdapat pesan di dalamnya dan akan muncul tombol *download* gambar	Sukses
8	Ketika menekan tombol *download* setelah berhasil melakukan proses *embedded*	Mengunduh gambar yang sebelumnya sudah melalui proses *embedded*	Sukses

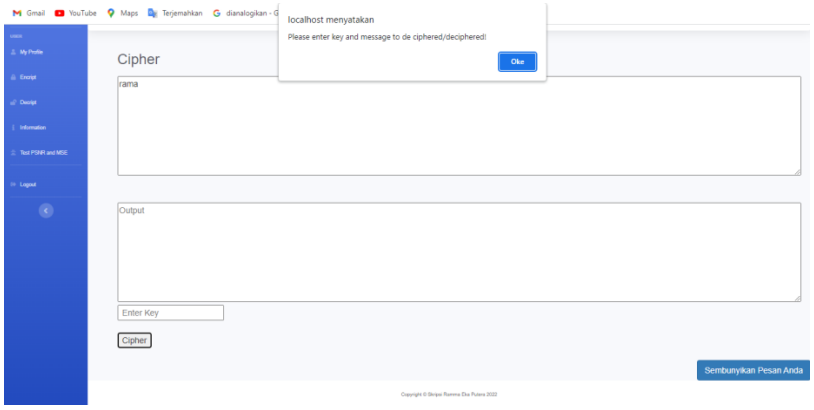
Pada pengujian di Table 4.6 Pengujian menu *encode* menampilkan fungsi *cipher*, fungsi *button* , fungsi menampilkan data hasil *encode* dan juga menampilkan beberapa *alert*.

Pada gambar di bawah ini yaitu Gambar 4.37 pengguna gagal melanjutkan untuk menyisipkan pesan karena belum menuliskan pesan yang akan dilakukan proses enkripsi.



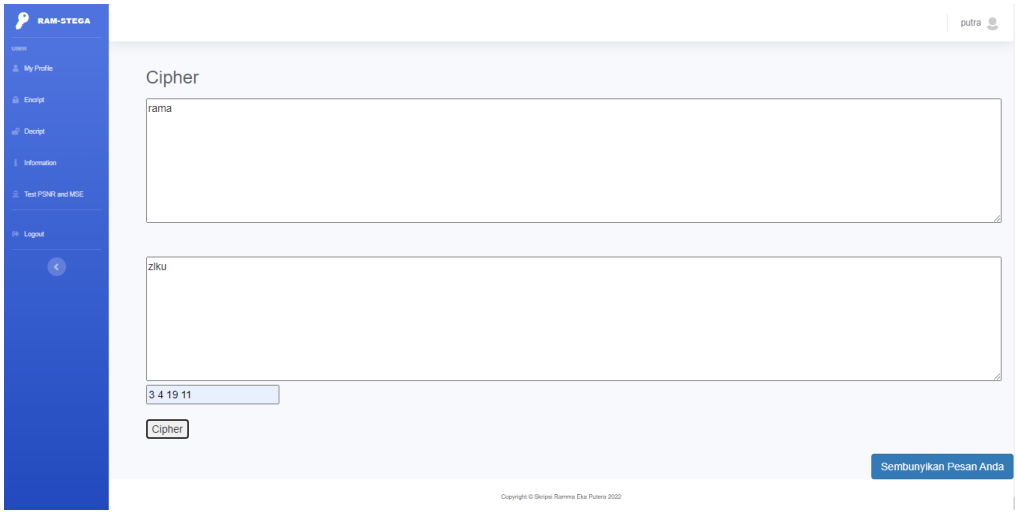
Gambar 4. 37 Proses Enkripsi Gagal Dengan Keterangan Pesan Belum Diisikan

Pada gambar di bawah ini yaitu Gambar 4.38 pengguna gagal melanjutkan untuk menyisipkan pesan karena belum menuliskan kunci yang akan dilakukan dalam proses enkripsi.



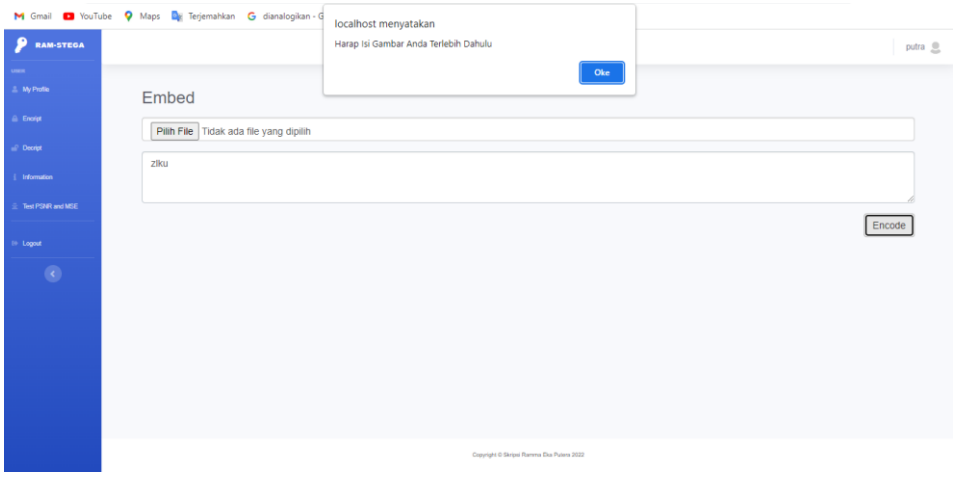
Gambar 4. 38 Proses Enkripsi Gagal Dengan Keterangan Kunci Belum Dimasukkan

Pada gambar di bawah ini yaitu Gambar 4.39 pengguna berhasil melanjutkan untuk menyisipkan pesan dan akan muncul pesan hasil enkripsi.



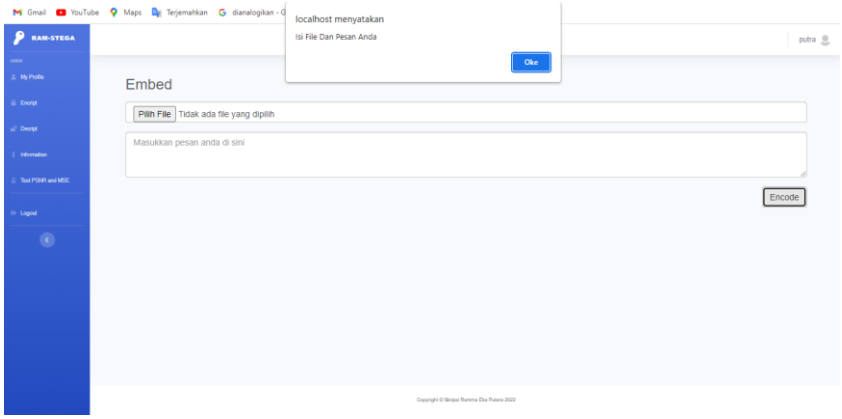
Gambar 4. 39 Proses Enkripsi Berhasil dan Akan Muncul Pesan Hasil Enkripsi

Pada gambar di bawah ini yaitu Gambar 4.40 pengguna berhasil melanjutkan untuk menyisipkan pesan ke dalam gambar namun galal dalam proses penyisipannya karena belum adanya gambar yang tersedia.



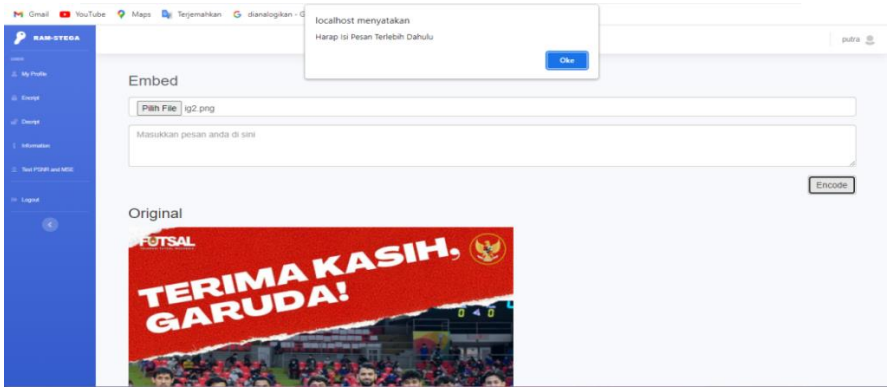
Gambar 4. 40 Proses Penyisipan Pesan Gagal Dengan Keterangan Gambar Belum Dimasukkan

Pada gambar di bawah ini yaitu Gambar 4.41 proses penyisipan gagal apabila terjadi skenario pengguna belum memasukkan gambar dan pesan yang akan disisipkan.



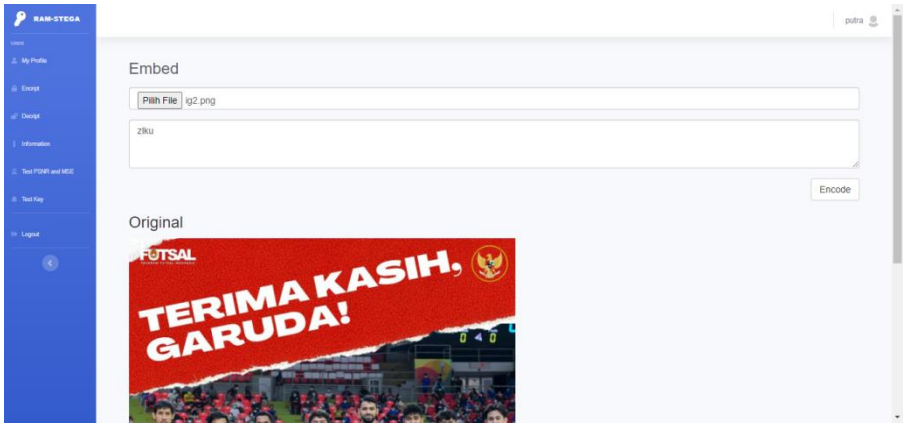
Gambar 4. 41 Proses Penyisipan Pesan Gagal Dengan Keterangan Gambar dan Pesan Belum Dimasukkan

Pada gambar di bawah ini yaitu Gambar 4.42 proses penyisipan gagal apabila terjadi skenario pengguna belum memasukkan pesan yang akan disisipkan.



Gambar 4. 42 Proses Penyisipan Pesan Gagal Dengan Keterangan Pesan Belum Dimasukkan

Pada gambar di bawah ini yaitu Gambar 4.43 proses penyisipan berhasil setelah pengguna memasukkan pesan yang akan disisipkan dan gambar sebagai media untuk menyisipkan pesan



Gambar 4. 43 Proses Penyisipan Berhasil dan Akan Menampilkan Gambar yang Telah Disisipi Pesan

4.5.4 Pengujian Menu Decode

Tabel 4. 7 Pengujian Menu *Decode* Menggunakan Metode *BlackBox Testing*

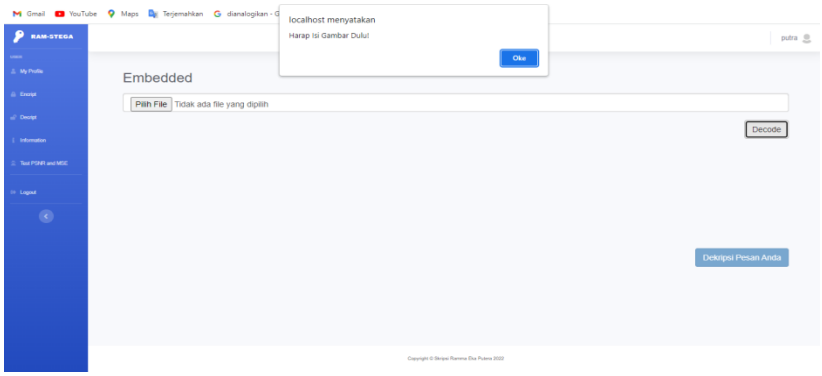
NO	Fungsi	Kebutuhan	Hasil
1	Ketika menekan tombol dekripsi pesan anda namun pengguna belum memasukkan gambar	Menampilkan peringatan untuk memasukkan gambar terlebih dahulu	Sukses
2	Ketika menekan tombol dekripsi pesan anda dan pengguna sudah memasukkan gambar	Menampilkan pesan yang ada dalam gambar	Sukses
3	Ketika menekan tombol decipher namun pengguna belum memasukkan pesan yang akan dideskrip	Menampilkan peringatan untuk pengguna memasukkan pesan untuk dilakukan proses *decipher*	Sukses
4	Ketika menekan tombol *decipher* namun pengguna belum memasukkan kunci yang akan dideskrip	Menampilkan peringatan untuk pengguna memasukkan kunci untuk dilakukan proses *decipher*	Sukses

Lanjutan Tabel 4.7 Pengujian Menu *Decode* Menggunakan Metode *BlackBox Testing*

5	Ketika menekan tombol *decipher* namun pengguna sudah memasukkan pesan dan kunci yang akan dideskrip	Menampilkan pesan yang telah dideskrip yang sebelumnya sudah melalui proses enkripsi	Sukses

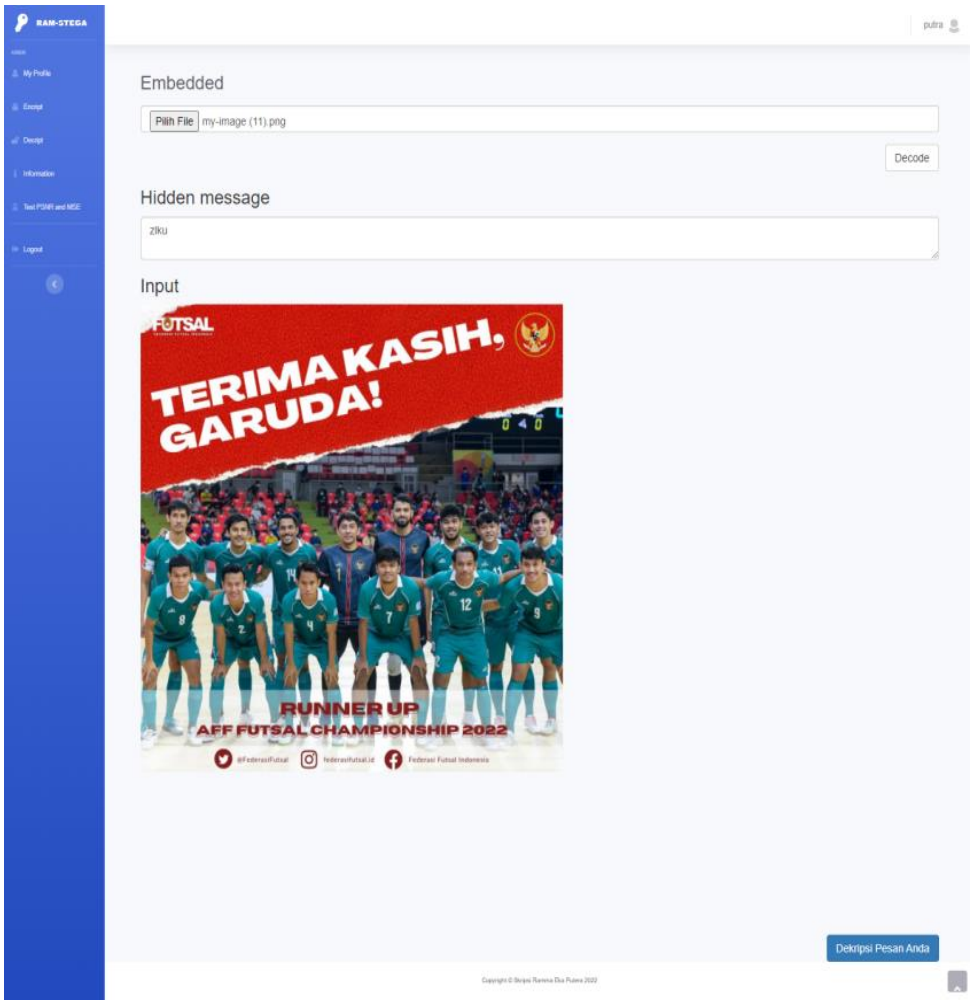
Pada pengujian di Table 4.7 pengujian menu *decode* menampilkan fungsi jika decode berhasil akan menampilkan pesan yang ada dalam gambar dan jika proses *decode* tidak berhasil maka akan keluar keterangan kegagalan *decode* tersebut.

Pada gambar dibawah ini yaitu Gambar 4.44 pengguna akan masuk ke dalam halaman *encode*, yang pertama pengguna dapat melihat pesan yang disisipkan dalam gambar dengan mengunggah gambar yang di dalamnya terdapat sebuah pesan, dalam skenario ini pengguna belum mengunggah gambar dan akan muncul keterangan pengguna harus memasukkan gambar terlebih dahulu.



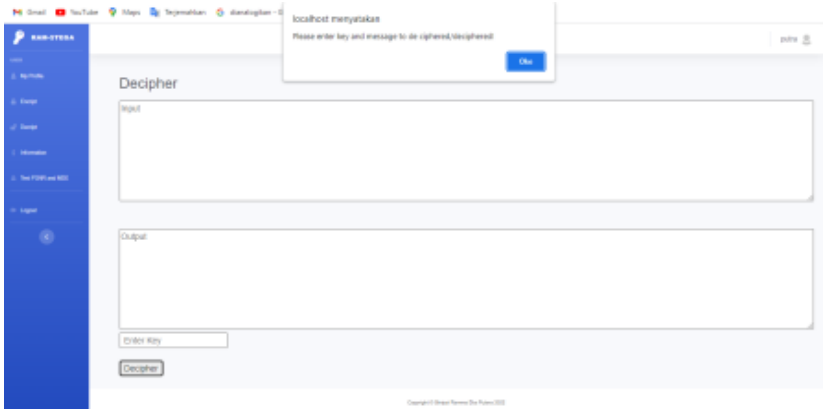
Gambar 4. 44 Proses Extract Gagal Dengan Keterangan Belum Ada Gambar Yang Diunggah

Pada gambar di bawah ini yaitu Gambar 2.45 proses *extract* berhasil setelah pengguna mengunggah gambar dan akan terlihat pesan yang tersembunyi di dalam gambar tersebut.



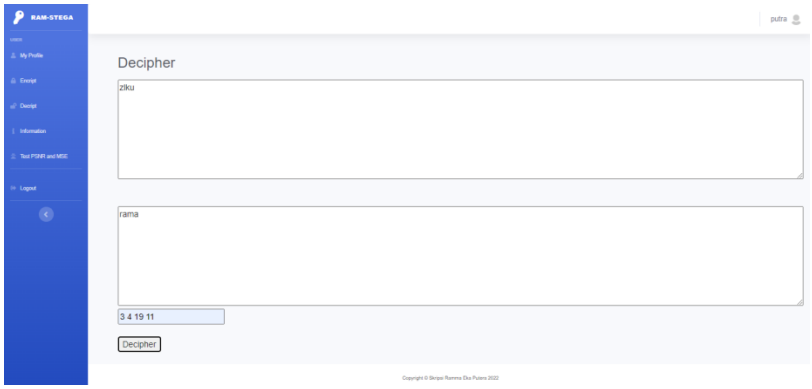
Gambar 4. 45 Proses *Extract* Berhasil Dan Akan Muncul Pesan Yang Tersembunyi Di Dalam Gambar

Pada Gambar di bawah ini yaitu Gmbar 2.46 yaitu pengguna akan dapat masuk ke halaman decipher setelah berhasil melakukan proses *extract*, dalam skenario ini pengguna belum memasukkan pesan dan kunci sehingga proses tidak dapat berjalan.



Gambar 4. 46 Proses *Decipher* Gagal Dengan Keterangan Belum Memasukkan Pesan Dan Kunci

Pada gambar di bawah yaitu Gambar 4.47 proses decipher berhasil setelah pengguna memasukkan pesan dan kunci ke dalam *form* yang sudah disediakan dan akan terlihat pesan yang sebelumnya berupa huruf acak menjadi sebuah kalimat.



Gambar 4. 47 Proses *Decipher* Berhasil Dan Akan Terlihat Pesan Yang Telah Didekrip

4.5.5 Pengujian Menu Informasi dan Test

Tabel 4. 8 Pengujian Menu Informasi dan *Test* Menggunakan Metode *BlackBox Testing*

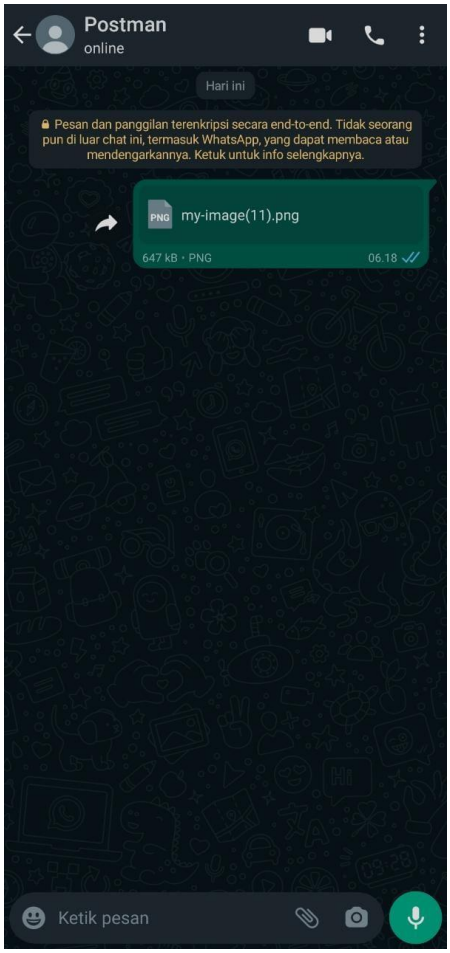
NO	Fungsi	Kebutuhan	Hasil
1	Ketika pengguna menekan menu informasi pada *website*	Menampilkan Video dan informasi tentang langkah-langkah untuk menjalankan proses enkripsi dan dekripsi	Sukses
2	Ketika pengguna menekan menu *Test MSE* dan *PSNR* pada *website*	Menampilkan nilai dari pengujian *MSE* dan *PSNR* pada gambar yang belum melalui proses stegano dan pada gambar yang sudah melalui proses stegano	Sukses

Pada pengujian di Table 4.8 pengujian menu informasi dan test *MSE* dan *PSNR* menampilkan jika pengguna masuk ke menu informasi dan test maka dapat dilihat pada menu informasi tentang cara penggunaan aplikasi dari proses *encrypt* sampai dengan *decrypt*.

4.5.6 Pengujian Pengiriman Gambar

a) Pengujian Melalui Dokumen WhatsApp

Pada metode ini pengiriman gambar menggunakan pengiriman dokumen. didapatkan dari hasil *download* gambar yang dihasilkan aplikasi. Pengiriman ini ditujukan untuk mengetahui apakah terdapat perubahan terhadap gambar yang dikirim menggunakan metode pengiriman dokumen. Selain itu gambar yang dikirimkan dengan metode ini bisa digunakan sebagai bahan untuk aplikasi pengiriman pesan lainnya seperti telegram. Seperti dapat dilihat pada Gambar 4.48 di bawah ini.



Gambar 4. 48 Pengiriman Gambar Stegano Melalui Dokumen WhatsApp

Pada gambar di bawah ini yaitu Gambar 4.49 dan Gambar 4.50 dapat dilihat gembar setelah dikirim melalui whatsapp, tidak terlihat perbedaan antara gambar sebelum dikirim dan sesudah dikirim. Karena tidak ada perubahan yang signifikan sehingga tidak terdapat kecurigaan pada gambar tersebut memiliki sebuah pesan tersembunyi di dalamnya.

Gambar 4. 49 Gambar Stegano Dikirim Melalui Dokumen WhatsApp

Gambar 4. 50 Gambar Stegano Sesudah Dikirim Melalui Dokumen WhatsApp

Tabel 4. 9 Pengujian Menggunakan Dokumen Whatapp

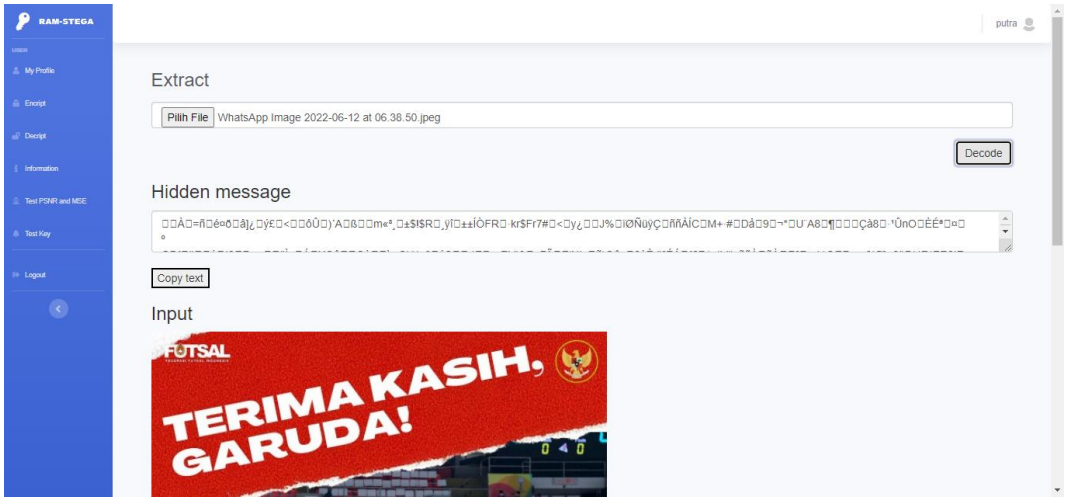
Stego Image	Stego Image Dokumen Whatsapp	Stego Image Size	Stego Image Dokumen Szie
		632 KB	632 KB

Dapat dilihat pada Tabel 4.9 Pengujian pengiriman gambar menggunakan dokumen menghasilkan file yang bisa di *decode* dengan baik. Sehingga pesan yang terdapat dalam gambar dapat terbaca dan tidak ada perubahan yang signifikan. Hal ini membuktikan bahwa tidak ada proses kompres atau modifikasi dari pihak whatsapp terhadap gambar yang dikirim melalui dokumen. Proses percobaan dengan metode ini menghasilkan pesan yang terbaca.

b) Pengujian Melalui Media WhatsApp

Tujuan dari pengiriman gambar menggunakan pengiriman gambar atau media adalah untuk mengetahui apakah terdapat perubahan pada gambar saat dikirim menggunakan pengiriman gambar melalui aplikasi whatsapp. Dari pengujian pengiriman menggunakan gambar atau media didapatkan hasil terdapat perubahan pada gambar. Jadi pesan yang sebelumnya disisipkan tidak dapat terbaca pada pada gambar setelah dikirim menggunakan metode ini. Artinya terdapat sebuah modifikasi file oleh whatsapp. Modifikasi yang dilakukan oleh whatsapp adalah mengkompress file gambar agar menghasilkan file yang berukuran lebih kecil. Hasil percobaan menggunakan metode ini dapat dilihat pada Gambar 4.51 dan 4.52.

Gambar 4. 51 Gambar Stegano Dikirim Menggunakan Media WhatsApp



Gambar 4. 52 Gambar Stegano Dikirim Menggunakan Media WhatsApp

Pada gambar di atas yaitu Gambar 4.52 pesan yang di ekstrak berubah menjadi “À=ñé¤ðâ]¿ý£<ôÛ)´Aßm«³….”. Artinya pesan yang sudah disisipkan sebelumnya hilang dan tidak terbaca. Jadi gambar yang sebelumnya berformat PNG berubah menjadi JPEG setelah dikirim. Hal ini sudah mengindikasikan bahwa pesan yang disisipkan sudah hilang atau tidak bisa terbaca lagi.

Tabel 4. 10 Pengujian Menggunakan Media Whatapp

Stego Image	Stego Image Dokumen Whatsapp	Stego Image Size	Stego Image Media Size
		632 KB	138 KB

Dapat dilihat pada Tabel 4.10 Pengujian pengiriman gambar menggunakan media menghasilkan file yang tidak bisa di *decode* dengan baik. Sehingga pesan yang terdapat dalam gambar tidak dapat terbaca dan ada perubahan yang terjadi. Hal ini membuktikan bahwa ada proses kompres atau modifikasi dari pihak whatsapp terhadap gambar yang dikirim melalui media. Proses percobaan dengan metode ini menghasilkan pesan yang tidak terbaca.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan yang telah dijabarkan pada setiap bab sebelumnya maka penelitian ini dapat ditarik kesimpulan adalah sebagai berikut : dikemas dalam judul "Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma *Hill Cipher* Dan Metode *Least Significant Bit* (*LSB*)." Berkaitan dalam penyisipan pesan dalam gambar, maka rumusan masalah dalam penelitian ini adalah:

1. Aplikasi ini dapat melakukan proses penyisipan data pesan teks ke dalam citra gambar dengan menerapkan algoritma *Hill Cipher* metode *Least Significant Bit*.
2. Aplikasi ini juga berhasil melakukan pengukuran kualitas gambar sebelum dan sesudah melalui proses steganografi. Berdasarkan dari hasil pengujian menggunakan *peak signal-to-noise ratio (PSNR,)* dan *Mean Square Error (MSE)*, citra hasil enkripsi tidak mengalami perubahan yang signifikan tetapi mengalami perubahan pada kapasitas citra yang bertambah pada saat setelah disisipkan data pesan.
3. dari hasil analisis jawaban seluruh responden sebanyak 7 orang terhadap pengujian fungsionalitas program didapatkan hasil 100% SESUAI yang menunjukkan bahwa keseluruhan fitur sistem dapat berjalan sesuai dengan hasil yang diharapkan.
4. Terdapat perubahan file pada pengiriman gambar menggunakan pengiriman media whatsapp dan tidak terjadi perubahan pada gambar yang dikirim menggunakan pengiriman dokumen.
5. Pesan yang disisipkan pada gambar hanya bisa terbaca pada pengiriman gambar menggunakan pengiriman dokumen. Hal ini disebabkan karena pada pengiriman menggunakan dokumen tidak terjadi modifikasi dari pihak whatsapp, sedangkan pada pengiriman file stiker menggunakan pengiriman gambar terjadi perubahan pada gambar yang dilakukan oleh pihak whatsapp.

5.2 Saran

Aplikasi ram-stega ini masih jauh dari kesempurnaan, masih banyak kekurangan untuk menciptakan sebuah aplikasi yang baik tentu perlu dilakukan pengembangan dari sisi manfaat maupun dari sisi kerja sistem, berikut beberapa saran untuk mengembangkan aplikasi ram-stega ini :

Memanfaatkan citra lain selain gambar digital dalam teknik penyisipan data pesan teks.

1. Tidak merubah piksel sama sekali pada gambar yang digunakan sebagai media penyembunyian pesan.
2. Menggunakan media selain gambar dan bisa berupa video agar bit yang digunakan bisa lebih banyak.
3. Pesan yang dimasukkan ke dalam gambar bisa diubah bukan berupa teks, dapat diubah menjadi gambar, video, musik, dll.

DAFTAR PUSTAKA

Aprilia, I., Ariyanti, D., & Izzuddin, A. (2019). Analisa Pengukuran Kualitas Citra Hasil Steganografi. *Seminar Nasional (SENIATI) 2019 "Inovasi Dan Aplikasi Teknologi Berkelanjutan Di Era Revolusi Industri 4.0," 5 (4)*(Technology 4.0), 116–121.

Azlansyah, M., & Setiyono, B. (2019). Penyisipan Pesan pada Citra Digital Menggunakan Metode Least Significant Bit. *Jurnal Sains Dan Seni ITS*, *8*(1). https://doi.org/10.12962/j23373520.v8i1.37658

Download, S., Pack, P. D. F., Algorit, P., Cipher, H., Least, D., Bit, S., & Pa, P. P. (n.d.). *Algoritma hill chiper*.

Laoli, D., Sinaga, B., & Sinaga, A. S. R. M. (2020). Penerapan Algoritma Hill Cipher Dan Least Significant Bit (LSB) Untuk Pengamanan Pesan Pada Citra Digital. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *4*(3), 1. https://doi.org/10.14421/jiska.2020.43-01

Malese, L. P. (2020). *Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit (Lsb). XIII*(November), 96–101. https://doi.org/10.31234/osf.io/8g39h

Noertjahyana, A., Hartono, S., Gunadi, K., Petra, U. K., & Surabaya, J. S. (2012). *Citra Digital Dengan Menggunakan Metode Lsb (Least Significant Bit). 13*(2), 113–121.

Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. (2016). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *10*(1), 20. https://doi.org/10.30872/jim.v10i1.23

Santa, E., & Situmorang, Z. (2018). Algoritma Hill Cipher Untuk Pengamanan Pengiriman File Via E-Mail. *Publikasi Ilmiah Teknologi Informasi Neumann (PITIN)*, 46–50.

Sari, J. I., Sihotang, H. T., & Informatika, T. (2017). Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma Hill Cipher Dan Metode Least Significant Bit (LSB). *Jurnal Mantik Penusa*, *1*(2), 1–8. http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/253

Supardi, S., Alkodri, A. A., & Isnanto, B. (2021). Teknik Steganografi Penyembunyian Pesan Text Rahasia Pada Citra Digital Dengan Metode Least Significant Bit. *Jurnal Sisfotek Global*, *11*(1), 1–5. https://doi.org/10.38101/sisfotek.v11i1.351

Wiryawan, et al, (Univ. Pend. (2019). Steganografi Berdasarkan Metode Least Significant Bit (LSB). *Jurnal Ilmu Komputer Indonesia (JIKI)*, *1*, 34–40.

(Azlansyah & Setiyono, 2019; Download et al., n.d.; Laoli et al., 2020; Malese, 2020; Noertjahyana et al., 2012; Pabokory et al., 2016; Santa & Situmorang, 2018; Sari et al., 2017; Supardi et al., 2021; Wiryawan, et al, 2019)

Aprilia, I., Ariyanti, D., & Izzuddin, A. (2019). Analisa Pengukuran Kualitas Citra Hasil Steganografi. *Seminar Nasional (SENIATI) 2019 "Inovasi Dan Aplikasi Teknologi Berkelanjutan Di Era Revolusi Industri 4.0," 5 (4)*(Technology 4.0), 116–121.

Azlansyah, M., & Setiyono, B. (2019). Penyisipan Pesan pada Citra Digital Menggunakan Metode Least Significant Bit. *Jurnal Sains Dan Seni ITS*, *8*(1). https://doi.org/10.12962/j23373520.v8i1.37658

Download, S., Pack, P. D. F., Algorit, P., Cipher, H., Least, D., Bit, S., & Pa, P. P. (n.d.). *Algoritma hill chiper*.

Laoli, D., Sinaga, B., & Sinaga, A. S. R. M. (2020). Penerapan Algoritma Hill Cipher Dan Least Significant Bit (LSB) Untuk Pengamanan Pesan Pada Citra Digital. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *4*(3), 1. https://doi.org/10.14421/jiska.2020.43-01

Malese, L. P. (2020). *Penyembunyian Pesan Rahasia Pada Citra Digital Dengan Teknik Steganografi Menggunakan Metode Least Significant Bit (Lsb). XIII*(November), 96–101. https://doi.org/10.31234/osf.io/8g39h

Noertjahyana, A., Hartono, S., Gunadi, K., Petra, U. K., & Surabaya, J. S. (2012). *Citra Digital Dengan Menggunakan Metode Lsb (Least Significant Bit). 13*(2), 113–121.

Pabokory, F. N., Astuti, I. F., & Kridalaksana, A. H. (2016). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *10*(1), 20. https://doi.org/10.30872/jim.v10i1.23

Santa, E., & Situmorang, Z. (2018). Algoritma Hill Cipher Untuk Pengamanan Pengiriman File Via E-Mail. *Publikasi Ilmiah Teknologi Informasi Neumann (PITIN)*, 46–50.

Sari, J. I., Sihotang, H. T., & Informatika, T. (2017). Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma Hill Cipher Dan Metode Least Significant Bit (LSB). *Jurnal Mantik Penusa*, *1*(2), 1–8. http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/253

Supardi, S., Alkodri, A. A., & Isnanto, B. (2021). Teknik Steganografi Penyembunyian Pesan Text Rahasia Pada Citra Digital Dengan Metode Least Significant Bit. *Jurnal Sisfotek Global*, *11*(1), 1–5. https://doi.org/10.38101/sisfotek.v11i1.351

Wiryawan, et al, (Univ. Pend. (2019). Steganografi Berdasarkan Metode Least Significant Bit (LSB). *Jurnal Ilmu Komputer Indonesia (JIKI)*, *1*, 34–40.

LAMPIRAN

Bahan yang akan digunakan pada penelitian ini berupa citra digital, adapun beberapa contoh citra digital sebagai berikut:

Lampiran1. 1 Gambar Cover yang Digunakan Untuk Stegano

Lampiran1. 2 Gambar Cover yang Digunakan Untuk Stegano

Lampiran 2 *Source Code Encrypt* dan *Decrypt*

```
// melakukan perhitungan enkripsi dengan mengalikan
plaintext dengan matriks kunci
function encrypt(plaintext, k) {
keys = k.split(" ");
if (plaintext.length % 2 == 1) { plaintext = plaintext +
"x"; }
if (keys.length != 4) { alert("key should consist of 4
integers"); return; }
for (i = 0; i < 4; i++) keys[i] = keys[i] % 26;
ciphertext = "";
for (i = 0; i < plaintext.length; i += 2) {
ciphertext += String.fromCharCode((keys[0] *
(plaintext.charCodeAt(i) - 97) + keys[1] *
(plaintext.charCodeAt(i + 1) - 97)) % 26 + 97);
ciphertext += String.fromCharCode((keys[2] *
(plaintext.charCodeAt(i) - 97) + keys[3] *
(plaintext.charCodeAt(i + 1) - 97)) % 26 + 97);
}
return ciphertext;
}
// melakukan perhitungan dekripsi dengan mengalikan
ciphertext dengan matriks kunci yang sudah melalui
proses invers
function decrypt(ciphertext, k) {
keys = k.split(" ");
if (ciphertext.length % 2 == 1) { alert("ciphertext is
not divisible by 2 (wrong algorithm?)"); return; }
if (keys.length != 4) { alert("key should consist of 4
integers"); return; }
for (i = 0; i < 4; i++) keys[i] = keys[i] % 26;
det = keys[0] * keys[3] - keys[1] * keys[2];
det = ((det % 26) + 26) % 26;
di = 0;
for (i = 0; i < 26; i++) { if ((det * i) % 26 == 1) di
= i; }
if (di == 0) { alert("could not invert, try different
```

```
37)    key"); return; }
38)    ikeys = new Array(4);
39)    ikeys[0] = (di * keys[3]) % 26; ikeys[1] = (-1 * di *
40)    keys[1]) % 26;
41)    ikeys[2] = (-1 * di * keys[2]) % 26; ikeys[3] = di *
42)    keys[0];
43)    for (i = 0; i < 4; i++) { if (ikeys[i] < 0) ikeys[i]
44)    += 26; }
45)    plaintext = "";
46)    for (i = 0; i < ciphertext.length; i += 2) {
47)    plaintext += String.fromCharCode((ikeys[0] *
48)    (ciphertext.charCodeAt(i) - 97) + ikeys[1] *
49)    (ciphertext.charCodeAt(i + 1) - 97)) % 26 + 97);
50)    plaintext += String.fromCharCode((ikeys[2] *
51)    (ciphertext.charCodeAt(i) - 97) + ikeys[3] *
52)    (ciphertext.charCodeAt(i + 1) - 97)) % 26 + 97);
53)    }
54)    return plaintext;
55)    }
56)    // mendapatkan pesan dan kunci yang dimasukkan oleh
57)    pengguna dan mengacaknya
58)    function cipherButtonFunction() {
59)    var enteredKey =
60)    document.getElementById('enteredKey').value.toLowerCas
61)    e().replace(/[^0-9 ]/g, "");
62)    var message =
63)    document.getElementById("inputMessage").value.toLowerC
64)    ase().replace(/[^a-z]/g, "");
65)    if (enteredKey == "" || message == "") {
66)    alert("Please enter key and message to de
67)    ciphered/deciphered!");
68)    return;
69)    }
70)    var result = encrypt(message, enteredKey);
71)    document.getElementById("result").value = result;
72)    }
73)    // mendapatkan pesan dan kunci yang dimasukkan oleh
74)    pengguna dan menguraikannya
```

```
75)    function decipherButtonFunction() {
76)    var enteredKey =
77)    document.getElementById('enteredKey').value.toLowerCas
78)    e().replace(/[^0-9 ]/g, "");
79)    var message =
80)    document.getElementById("inputMessage").value.toLowerC
81)    ase().replace(/[^a-z]/g, "");
82)    if (enteredKey == "" || message == "") {
83)    alert("Please enter key and message to de
84)    ciphered/deciphered!");
85)    return;
86)    }
87)    var result = decrypt(message, enteredKey);
88)    document.getElementById("result").value = result;
89)    }
90)    function cops() {
91)    var copyText = document.getElementById("result");
92)    copyText.select();
93)    copyText.setSelectionRange(0, 99999);
94)    navigator.clipboard.writeText(copyText.value);
95)    alert("Copied the text: " + copyText.value);
96)    }
```

Lampiran 3 Souerce Code Embed dan Extract

```
1)   $('button.encode, button.decode').click(function (event) {
2)   event.preventDefault();
3)   });
4)   //mengambil gambar yang akan didekripsi
5)   function previewDecodeImage() {
6)   var file =
      document.querySelector('input[name=decodeFile]').files[0];
7)   previewImage(file, ".decode canvas", function () {
8)   $(".decode").fadeIn();
9)   });
10)  }
11)  //mengambil gambar yang akan dienkripsi
12)  function previewEncodeImage() {
```

```
13) var file =
     document.querySelector("input[name=baseFile]").files[0];
14) $(".images .nulled").hide();
15) $(".images .message").hide();
16) previewImage(file, ".original canvas", function () {
17) $(".images .original").fadeIn();
18) $(".images").fadeIn();
19) });
20) }
21) //mendeteksi gambar yang sudah dimasukkan
22) function previewImage(file, canvasSelector, callback) {
23) var reader = new FileReader();
24) var image = new Image;
25) var $canvas = $(canvasSelector);
26) var context = $canvas[0].getContext('2d');
27) if (file) {
28) reader.readAsDataURL(file);
29) }
30) reader.onloadend = function () {
31) image.src = URL.createObjectURL(file);
32) image.onload = function () {
33) $canvas.prop({
34) 'width': image.width,
35) 'height': image.height
36) });
37) context.drawImage(image, 0, 0);
38) callback();
39) }
40) }
41) }
42) //melakukan penyembunyian pesan ke dalam gambar
43) function encodeMessage() {
44) // validasi pesan apakah memenuhi syarat untuk melakukan
     proses stegano
45) var a = document.getElementById("validasi_encode");
46) var filesLength = a.files.length;
47) var pesan = document.getElementById("pesan").value;
48) if (pesan.length == '0' && filesLength == '0') {
```

```
49)  alert('Isi File Dan Pesan Anda');
50)  } else if (filesLength != '0' && pesan.length == '0') {
51)  alert('Harap Isi Pesan Terlebih Dahulu');
52)  } else if (pesan.length != '0' && filesLength == '0') {
53)  alert('Harap Isi Gambar Anda Terlebih Dahulu');
54)  } else {
55)  $(".error").hide();
56)  $(".binary").hide();
57)  var text = $("textarea.message").val();
58)  var $originalCanvas = $('.original canvas');
59)  var $nulledCanvas = $('.nulled canvas');
60)  var $messageCanvas = $('.message canvas');
61)  var originalContext = $originalCanvas[0].getContext("2d");
62)  var nulledContext = $nulledCanvas[0].getContext("2d");
63)  var messageContext = $messageCanvas[0].getContext("2d");
64)  var width = $originalCanvas[0].width;
65)  var height = $originalCanvas[0].height;
66)  // periksa apakah gambarnya cukup besar untuk menyembunyikan
      pesan
67)  if ((text.length * 8) > (width * height * 3)) {
68)  $(".error")
69)  .text("Text too long for chosen image....")
70)  .fadeIn();
71)  return;
72)  }
73)  $nulledCanvas.prop({
74)  'width': width,
75)  'height': height
76)  });
77)  $messageCanvas.prop({
78)  'width': width,
79)  'height': height
80)  });
81)  // normalisasikan gambar asli
82)  var original = originalContext.getImageData(0, 0, width,
      height);
83)  var pixel = original.data;
84)  for (var i = 0, n = pixel.length; i < n; i += 4) {
```

```
85)  for (var offset = 0; offset < 3; offset++) {
86)  if (pixel[i + offset] % 2 != 0) {
87)  pixel[i + offset]--;
88)  }
89)  }
90)  }
91)  nulledContext.putImageData(original, 0, 0);
92)  // mengubah pesan menjadi angka biner
93)  var binaryMessage = "";
94)  for (i = 0; i < text.length; i++) {
95)  var binaryChar = text[i].charCodeAt(0).toString(2);
96)  // Pad with 0 until the binaryChar has a lenght of 8 (1 Byte)
97)  while (binaryChar.length < 8) {
98)  binaryChar = "0" + binaryChar;
99)  }
100) binaryMessage += binaryChar;
101) }
102) $('.binary textarea').text(binaryMessage);
103) // memasukkan angka biner ke gambar
104) var message = nulledContext.getImageData(0, 0, width,
      height);
105) pixel = message.data;
106) counter = 0;
107) for (var i = 0, n = pixel.length; i < n; i += 4) {
108) for (var offset = 0; offset < 3; offset++) {
109) if (counter < binaryMessage.length) {
110) pixel[i + offset] += parseInt(binaryMessage[counter]);
111) counter++;
112) }
113) else {
114) break;
115) }
116) }
117) }
118) messageContext.putImageData(message, 0, 0);
119) $(".binary").fadeIn();
120) $(".images .nulled").fadeIn();
121) $(".images .message").fadeIn();
```

```
122) }
123) };
124) //melakukan proses enkripsi terhadapat gambar yang di
      dalamnya terdapat pesan
125) function decodeMessage() {
126) var a = document.getElementById("validasi");
127) var filesLength = a.files.length;
128) if (filesLength == '0') {
129) alert('Harap Isi Gambar Dulu!');
130) } else {
131) var $originalCanvas = $('.decode canvas');
132) var originalContext = $originalCanvas[0].getContext("2d");
133) var original = originalContext.getImageData(0, 0,
      $originalCanvas.width(), $originalCanvas.height());
134) var binaryMessage = "";
135) var pixel = original.data;
136) for (var i = 0, n = pixel.length; i < n; i += 4) {
137) for (var offset = 0; offset < 3; offset++) {
138) var value = 0;
139) if (pixel[i + offset] % 2 != 0) {
140) value = 1;
141) }
142) binaryMessage += value;
143) }
144) }
145) //menampilkan pesan yang ada di dalam gambar
146) var output = "";
147) for (var i = 0; i < binaryMessage.length; i += 8) {
148) var c = 0;
149) for (var j = 0; j < 8; j++) {
150) c <<= 1;
151) c |= parseInt(binaryMessage[i + j]);
152) }
153) output += String.fromCharCode(c);
154) }
155) //menampilkan huruf dan menghapus simbol piksel
156) var temp = [];
157) for (i = 0; i < output.length; i++) {
```

```
158) if (output[i] != '\x00') {
159) temp.push(output[i]);
160) }
161) }
162) var hasil = temp.join("");
163) $('.tombol').removeAttr('disabled');
164) $('.binary-decode textarea').text(hasil);
165) $('.binary-decode').fadeIn();
166) }
167) };
168) function Download() {
169) var canvas = document.getElementById("download");
170) image = canvas.toDataURL("image/png").replace("image/png",
      "image/octet-stream");
171) var link = document.createElement('a');
172) link.download = "my-image.png";
173) link.href = image;
174) link.click();
175) }
176) function cop() {
177) var copyText = document.getElementById("salin");
178) copyText.select();
179) navigator.clipboard.writeText(copyText.value);
180) }
```

Lampiran Pengujian Sistem Menggunakan Blackbox Testing

SURAT PERNYATAAN PENGUJIAN VALIDASI

“RAM-STEGA”

Saya yang bertanda tangan di bawah ini sebagai responden pengujian sistem menggunakan blackbox testing :

1. Brian Vidyanjaya
2. Mohammad Marsa Kamal Setiawan
3. Mohammad Rizki Yanuarianto
4. Rizki Arisandhi Pramana
5. Bayu Agil Prananda
6. Indyra Ayu Wijayanti
7. Fadhila Dwi Kurniawan

Menyatakan dengan sebenar benarnya bahwa aplikasi telah dilakukan pengujian sesuai dengans scenario yang tercantum. Proses pengujian dilakukan secara tatap muka. Penguji dapat melakukan demo terhadap sistem penyembunyian pesan. Rentang waktu pengujian yaitu 9 Mei 2022 sampai 11 Mei 2022.

(Mohammad Rizki Yanuarianto) (Mohammad Marsa Kamal Setiawan)

(Brian Vidyanjaya) (Bayu Agil Prananda)

(Rizki Arisandhi Pramana)

(Indyra Ayu Wijayanti)

(Fadhila Dwi Kurniawan)