ANALISIS PENERAPAN METODE STEGANOGRAFI *HISTOGRAM SHIFTING* SEBAGAI ALTERNATIF OTENTIKASI PADA APLIKASI WEB HOSTING

SKRIPSI

Oleh:

Irsandy Maulana Satya Viddin

NIM 162410101094

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2020

ANALISIS PENERAPAN METODE STEGANOGRAFI *HISTOGRAM SHIFTING* SEBAGAI ALTERNATIF OTENTIKASI PADA APLIKASI WEB HOSTING

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan mendapat gelar Sarjana Komputer

Oleh:

Irsandy Maulana Satya Viddin

162410101094

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS ILMU KOMPUTER

UNIVERSITAS JEMBER

2020

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Sri Wiyandari dan Ayahanda Hasan Basri
2. Saudara perempuan Putri Tama Hasandy Satya Viddin
3. Guru-guruku sejak taman kanak-kanak hingga perguruan tinggi
4. Teman-teman Infinity angkatan 2016
5. Almamater Fakultas Ilmu Komputer Universitas Jember

MOTO

"*Bismillah*"

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Irsandy Maulana Satya Viddin

NIM : 162410101094

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Analisis Penerapan Metode Steganografi *Histogram Shifting* Sebagai Alternatif Otentikasi pada Aplikasi Web Hosting", adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika kemudian hari pernyataan ini tidak benar.

Jember, 10 September 2020

Yang menyatakan,

Irsandy Maulana Satya Viddin

NIM 162410101094

SKRIPSI

ANALISIS PENERAPAN METODE STEGANOGRAFI *HISTOGRAM SHIFTING* SEBAGAI ALTERNATIF OTENTIKASI PADA APLIKASI WEB HOSTING

Oleh:

Irsandy Maulana Satya Viddin

NIM 162410101094

Pembimbing:

Dosen Pembimbing Utama : Drs.Antonius C. P., M.App.Sc., Ph.D.

Dosen Pembimbing Pendamping : Diksy Media F. S.Kom., M.Kom

PENGESAHAN PEMBIMBING

Skripsi berjudul "Analisis Penerapan Metode Steganografi *Histogram Shifting* Sebagai Alternatif Otentikasi pada Aplikasi Web Hosting" telah disetujui dan disahkan pada:

Hari, tanggal : Rabu, 23 September 2020

Tempat : Fakultas Ilmu Komputer Universitas Jember

Disetujui oleh:

Dosen Pembimbing Utama	Dosen Pembimbing Pendamping
Drs.Antonius C. P., M.App.Sc., Ph.D.	Diksy Media F., S.Kom., M.Kom
NIP. 196909281993021001	NRP. 760016853

PENGESAHAN PENGUJI

Skripsi berjudul "Analisis Penerapan Metode Steganografi *Histogram Shifting* Sebagai Alternatif Otentikasi pada Aplikasi Web Hosting" telah disetujui dan disahkan pada:

Hari, tanggal : Rabu, 23 September 2020

Tempat : Fakultas Ilmu Komputer Universitas Jember

Tim Penguji:

Penguji I,	Penguji II,
Yanuar Nurdiansyah, ST,. M.Cs.	Qurrota A'yuni A.R., S.Pd., M.Sc.
NIP. 198201012010121004	NRP. 760018029

Mengesahkan:

Dekan Fakultas Ilmu Komputer

Prof. Dr. Saiful Bukhori, ST., M.Kom

NIP. 196811131994121001

RINGKASAN

Analisis Penerapan Metode Steganografi *Histogram Shifting* Sebagai Alternatif Otentikasi pada Aplikasi Web Hosting; Irsandy Maulana Satya Viddin, 162410101094; 2020: 79 halaman; Fakultas Ilmu Komputer Universitas Jember.

Otentikasi merupkan proses yang umum pada kebanyakan sistem informasi atau aplikasi saat ini. Proses ini bertujuan untuk mengetahui apakah seseorang memang benar memiliki hak untuk mengakses sistem atau aplikasi. Bentuk otentikasi yang banyak digunakan adalah otentikasi menggunakan kata sandi berbasis teks. Meski banyak digunakan, metode tersebut memiliki kelemahan karena mengharuskan seseorang untuk membuat kata sandi yang tidak mudah ditebak dan sekaligus mengingatnya. Hal ini dapat membuat seseorang menggunakan kata sandi yang sama untuk banyak akun. Karenanya, dibutuhkan alternatif metode otentikasi yang tidak membebani ingatan sesorang.

Histogram Shifting merupakan salah satu metode steganografi *reversible data hiding* dengan media citra. Metode ini menggunakan histogram citra untuk mendapatkan nilai *peak* dan *zero point*. Kedua nilai tersebut digunakan pada proses pergeseran histogram untuk menyisipkan pesan rahasia. Kedua nilai juga digunakan kembali ketika pesan rahasia diekstrak dan histogram digeser kembali ke awalnya.

Modul otentikasi yang dikembangkan terdiri dari fitur registrasi, *log in*, dan pemulihan kata sandi atau citra. Fitur registrasi menerapkan metode *Histogram Shifting* untuk menyisipkan kredensial pengguna ke dalam citra. Citra stego hasil penyisipan selanjutnya dapat digunakan untuk masuk ke akun pengguna. Fitur *log in* mengekstrak kredensial pengguna dari citra stego untuk dicocokkan dengan data di basis data. Jika data kredensial yang diekstrak sesuai dengan data di basis data, maka pengguna diotentikasi masuk ke akunya. Fitur pemulihan citra dapat digunakan untuk memperbarui kata sandi dan citra stego pengguna. Pengguna diharuskan memasukkan email dan tanggal lahir yang digunakan ketika melakukan registrasi. Email berisi URL pemulihan akan dikimkan ke alamt email pengguna

jika data pengguna ditemukan di basis data. Setelah membuka URL tersebut, pengguna dapat memperbarui kata sandi dan citra stego milikinya.

Pengujian dilakukan terhadap citra stego dan waktu eksekusi pada proses registrasi dan *log in*. Pengujian kualitas citra (*fidelity*) menghasilkan nilai PSNR rata-rata sebesar 52,52 dB, yang mana berarti citra memiliki kualitas tinggi. Pengujian kemampuan pemulihan (*recovery*) atau ekstraksi menunjukkan kesepuluh citra yang diuji mampu digunakan untuk masuk ke akun pengguna. Hal ini berarti kredensial pengguna berhasil diekstrak dengan tepat. Pengujian ketahanan (*robustness*) pesan rahasia terhadap berbagai teknik manipulasi citra menunjukkan bahwa pesan rahasia tidak tahan dengan semua teknik manipulasi yang dicoba. Citra yang telah dimanipulasi tidak dapat digunakan untuk masuk ke akun pengguna. Sementara untuk pengujian waktu eksekusi menghasilkan waktu rata-rata 0,265 detik untuk proses regsitrasi, dan 0,119 detik untuk proses *log in*. Pengujian kedua proses dilakukan sebanyak masing-masing 10 kali, menggunakan citra beresolusi 600 x 600 piksel.

PRAKATA

Puji Syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul "Analisa Penerapan Metode Steganografi *Histogram Shifting* Sebagai Alternatif Otentikasi pada Aplikasi Web Hosting". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan Pendidikan Strata Satu (S1) pada Fakultas Ilmu Komputer Universitas Jember.

Penyusunan skripsi ini tidak lepas dari dukungan berbagai pihak . Oleh karena itu penulis menyampaikan terima kasih kepada:

1. Prof. Dr. Saiful Bukhori, ST., M.Kom selaku Dekan Fakultas Ilmu Komputer Universitas jember.
2. Drs.Antonius C. P., M.App.Sc., Ph.D selaku Dosen Pembimbing Pertama dan Diksy Media F. S.Kom., M.Kom selaku Dosen Pembimbing Pendamping yang telah meluangkan waku, pikiran, dan perhatian dalam penulisan skripsi.
3. Seluruh Bapak dan Ibu dosen beserta civitas akademika di Fakultas Ilmu Komputer Universitas Jember.
4. Ibunda Sri Wiyandari dan Ayahanda Hasan Basri untuk segala bentuk kasih sayang, doa, dan dukungan yang senantiasa diberikan.
5. Saudara perempuan Putri Tama Hasandy SV yang selalu mendukung saya
6. Teman-teman Infinity angkatan 2016
7. Semua pihak yang tidak dapat disebutkan satu persatu

Dengan harapan bahwa penelitian ini nantinya terus berlanjut dan berkembang, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 10 September 2020

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Otentikasi merupkan proses yang umum pada kebanyakan sistem informasi atau aplikasi saat ini. Proses ini bertujuan untuk mengetahui apakah seseorang atau suatu entitas memang benar memiliki hak untuk mengakses sistem atau aplikasi tersebut. Menurut Suo, Zhu, & Owen (2005) otentikasi dapat dikelompokkan menjadi tiga tipe, yaitu *token-based authentication* (contohnya kartu ATM), *biometric-based authentication* (contohnya pemindai sidik jari), dan *knowledge-based authentication* (contohnya kata sandi berbasis teks). Tipe-tipe otentikasi tersebut dapat dikombinasikan, seperti contohnya pada otentikasi di mesin ATM yang mana kartu ATM dan PIN dibutuhkan untuk memverifikasi pemilik akun. Dibandingkan dengan tipe atau metode otentikasi lain, kata sandi berbasis teks merupakan bentuk otentikasi yang paling banyak digunakan (Idrus, et al. 2013).

Kata sandi berbasis teks memiliki keunggulan tersendiri dibandingkan metode otentikasi lainnya. Keunggulan tersebut terletak pada kemudahan penggunaan dan biaya yang rendah (Bonneau, et al. 2012). Penggunaan kata sandi berbasis teks dalam otentikasi umumnya cukup dengan mengetikkan kata sandi pengguna pada kolom masukkan yang tersedia. Jika kata sandi sesuai dengan data yang disimpan sistem, maka pengguna akan terotentikasi. Selain kemudahannya, kata sandi berbasis teks juga lebih murah dibandingkan metode lain seperti otentikasi biometri yang membutuhkan peralatan khusus untuk mengenali seorang pengguna.

Di balik keunggulannya, metode otentikasi menggunakan kata sandi berbasis teks memiliki kelemahan tersendiri. Metode otentikasi ini mengharuskan seseorang untuk membuat kata sandi yang tidak mudah ditebak dan sekaligus mengingatnya. Apabila seseorang memiliki banyak akun digital, seperti akun media sosial, email, *e-commerce, internet banking* dan sebagainya, maka seseorang akan kesulitan untuk mengingat kata sandi setiap akun tersebut. Hal ini dapat membuat seseorang menggunakan kata sandi yang sama untuk banyak akun. Penelitian yang dilakukan oleh Wash, et al. (2016) menunjukkan bahwa sebanyak 134 partisipan

menggunakan ulang password yang sama di rata-rata 9 situs berbeda. Penelitian lainnya oleh Florencio dan Herley (2007) terhadap setengah juta pengguna internet menunjukkan bahwa pengguna rata-rata memiliki 6,5 kata sandi yang masing-masing digunakan di 3,9 situs berbeda. Sebagai tambahan, diperkirakan juga bahwa 1,5% pengguna Yahoo lupa akan kata sandi mereka setiap bulannya. Penggunaan password yang sama untuk banyak akun tentunya mengurangi keamanan seseorang, karena apabila satu kata sandi dapat diketahui orang lain, maka akun-akun terkait dapat diserang dengan mudah. Melihat kelemahan metode otentikasi ini, maka dibutuhkan suatu alternatif otentikasi yang tidak membebani ingatan seseorang. Hal ini juga untuk mengurangi resiko penggunaan kata sandi yang sama berulang kali.

Steganografi merupakan ilmu yang mempelajari tentang penyembunyian pesan pada suatu media sedemikian hingga keberadaannya tidak diketahui. Cox, et al. (2008) menyebutkan "steganografi adalah praktek dalam mengubah sebuah karya secara tak terdeteksi guna menanamkan suatu pesan". Steganografi berfokus pada bagaimana menyembunyikan pesan atau informasi pada suatu media (gambar, suara, atau video) sehingga keberadaannya tidak dapat ditangkap oleh indra manusia. Salah satu metode steganografi yang dapat diterapkan pada media citra atau gambar adalah *Histogram Shifting*. Metode ini merupakan metode *reversible data hiding*, yang mana mampu mengembalikan citra asli setelah pesan rahasia diambil kembali. *Histogram Shifting* memanfaatkan tingkat kecerahan (*grey level*) dengan frekuensi atau jumlah piksel terendah pada histogram citra untuk menyisipkan pesan rahasia (Ni, et al. 2006).

Berdasarkan penjelasan di atas, penelitian ini bermaksud untuk menerapkan metode steganografi *Histogram Shifting* sebagai alternatif otentikasi kata sandi berbasis teks. Selain itu, penelitian ini juga bertujuan untuk menguji kualitas citra hasil steganografi (citra stego), ketahanan (*robustness*) pesan rahasian yang disisipkan, dan kemampuan pengungkapan (*recovery*) pesan rahasia. Pengujian lain yang akan dilakukan yaitu terhadap waktu eksekusi untuk proses registrasi dan *log in*. Pesan rahasia yang akan disisipkan ke citra sebelumnya dienkripsi terlebih dahulu untuk meningkatkan keamanan pesan. Obyek yang digunakan pada penelitain ini adalah aplikasi web hosting yang sedang dikembangkan oleh

kelompok riset di Fakultas Ilmu Komputer Universitas Jember. Sebagai batasan, penelitian ini hanya mengembangkan modul otentikasi tanpa mengembangkan aplikasi web hosting keseluruhan. Metode *Histogram Shifting* dipilih karena memiliki kompleksitas komputasi yang rendah (Ni, et al. 2006), sehingga sesuai untuk proses otentikasi seperti registrasi dan *log in*. Selain itu, metode ini juga mampu menghasilkan citra dengan nilai PSNR yang relatif tinggi, rata-rata sebesar 48 dB (Ni, et al. 2006). Pengembangan alternatif ini diharapkan dapat mempermudah pengguna dalam melakukan otentikasi, karena pengguna tidak harus mengingat kata sandi baru ketika mendaftarkan akun.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan di atas, rumusan masalah pada penelitin ini adalah sebagai berikut:

1. Bagaimana mekanisme alternatif otentikasi pada aplikasi web hosting dengan menggunakan metode steganografi *Histogram Shifting* ?
2. Bagaimana solusi untuk mengatasi permasalahan ketika citra yang telah disisipi pesan rahasia rusak atau hilang ?
3. Bagaimana melakukan pengujian terhadap citra stego berdasarkan kriteria kualitas citra, ketahanan (*robustness*) pesan rahasia, dan kemampuan pengungkapan (*recovery*) pesan rahasia ?
4. Berapa waktu eksekusi rata-rata pada proses registrasi dan *log in* ?

1.3 Tujuan

Adapun tujuan yang ingin dicapai dari penelitian ini yaitu:

1. Untuk mengetahui bagaimana mekanisme alternatif otentikasi pada aplikasi web hosting dengan menerapkan metode steganografi *Histogram Shifting*.
2. Untuk mengetahui bagaimana mengembangkan fitur pemulihan (*recovery*) citra stegano milik pengguna yang rusak atau hilang

3. Untuk mengetahui bagaimana melakukan pengujian terhadap citra stego berdasarkan kriteria kualitas citra, ketahanan (*robustness*) pesan rahasia, dan kemampuan pengungkapan (*recovery*) pesan rahasia.
4. Untuk mengetahui waktu eksekusi rata-rata pada proses registrasi dan *log in*.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian ini antara lain:

1. Bagi Akademis

   Penelitian ini diharapkan mampu memberi informasi dan masukkan tambahan terkait topik yang diangkat bagi pembaca teruatama untuk Fakultas Ilmu Komputer Universitas Jember.
2. Bagi Peneliti

   Mampu menambah pengetahuan dalam penerapan steganografi di bidang otentikasi.

1.5 Batasan Masalah

Beberapa batasan masalah pada penelitian ini antara lain:

1. Media yang digunakan untuk menyembunyikan informasi rahasia adalah media citra atau gambar dengan ekstensi .jpeg atau .png
2. Informasi yang disembunyikan berupa teks
3. Penelitian yang dilakukan hanya sebatas mengembangkan modul otentikasi (registrasi, *log in*, dan pemulihan citra), tanpa mengembangkan aplikasi web hosting keseluruhan
4. Pengembangan modul otentikasi berbasis web dengan menggunakan *framework* Laravel
5. Enkripsi dan dekripsi pesan rahasia menggunakan fungsi enkripsi dan dekripsi bawaan dari *framework* Laravel

1.6 Sistematika Penulisan

Sistematika penulisan pada skripsi ini adalah sebagai berikut

1. Pendahuluan

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, dan sistematika terkait penelitian yang dilakukan

2. Tinjauan Pustaka

   Bab ini memaparkan tinjauan mengenai kajian teori yang berkaitan dengan penelitian yang akan dilakukan.

3. Metodologi Penelitian

   Bab ini terdiri dari jenis penelitian yang digunakan dan tahapan penelitian yang meliputi analisi kebutuhan, implementasi, dan pengujian.

4. Implementasi

   Bab ini memaparkan mengenai penerapan metode yang digunakan dalam penelitian ini sesuai dengan analisi kebutuhan.

5. Hasil dan Pembahasan

   Bab ini menerangkan tentang hasil pengujian seperti yang dijelaskan pada bab metodologi penelitian. Selain itu, bab ini juga membahas mengenai hasil pengujian guna menjawab rumusan masalah terkait pengujian di bab awal.

6. Penutup

   Bab ini terdiri dari kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Bab ini akan memaparkan tinjauan mengenai kajian teori yang berkaitan dengan penelitian yang akan dilakukan. Teori-teori yang digunakan diambil dari literatur buku dan juga jurnal. Teori yang dipaparkan diharapkan dapat memberi gambaran mengenai penelitain ini. Berikut ini adalah penjelasan untuk masing-masing teori

2.1 Otentikasi

Otentikasi merupakan proses untuk memastikan identitas seseorang atau suatu entitas. Mare, Baker dan Gummeson (2016) menggunakan istilah *authenticators* yang merujuk pada "cara untuk membuktikan bahwa kita adalah orang yang berhak mengakses sumber daya terbatas dalam kehidupan kita". Otentikasi dalam suatu sistem informasi merujuk pada proses untuk memverifikasi apakah seseorang atau suatu entitas memang benar memiliki hak untuk mengakses sistem informasi tersebut.

Dalam dunia digital, otentikasi merupakan proses yang penting untuk menjaga kepemilikan suatu akun. Berbagai layanan di dunia maya mengharuskan penggunanya untuk membuat akun pada layanan tersebut. Tujuannya agar pengguna dapat menggunakan layanan tersebut sesuai keingiann atau kebutuhan mereka. Akun-akun tersebut dapat memuat informasi-informasi sensitif terkait pemilik akun ataupun layanan yang digunakan. Contohnya seperti nomor rekening, nomor kartu kredit, nomor telepon, alamat rumah, dan lain sebagainya. Berbagai innformasi sensitif tersebut sudah seharusnya dilindungi dan hanya boleh diakses oleh pemilik asli akun. Karenanya, proses otentikasi memiliki peran yang penting dalam menjaga keamanan akun seseorang.

Menurut Suo, Zhu dan Owen (2005) otentikasi sendiri dapat dibagi menjadi tiga area utama, yaitu:

a. Otentikasi Berbasis Pengetahuan (*Knowledge-based*)

Otentikasi jenis ini merupakan yang paling banyak digunakan, terdiri dari kata sandi berbasis teks (*text-based*) dan berbasis gambar (*picture-based*). Kata sandi berbasis gambar dapat dibagi lagi menjadi *recognition-based* dan *recall-based*.

b. Otentikasi Berbasis Token (*Token-based*)

Contoh otentikasi ini seperti *smart card, key card,* kartu ATM. Otentikasi jenis ini terkadang dipadukan dengan otentikasi berbasis pengetahuan (*knowledge-based*) untuk meningkatkan keamanan. Contohnya pada kartu ATM yang umumnya digunakan bersamaan dengan nomor PIN.

c. Otentikasi Berbasis Biometri (*Biometric-based*)

Termasuk dalam otentikasi ini adalah pemindai sidik jari, *iris scan*, dan *facial recognition*. Kelemahan dari otentikasi ini terletak pada biaya penggunaannya yang relatif tinggi, serta kecepatan identifikasi yang terkadang rendah. Meski begitu, otentikasi jenis ini mampu memberikan keamanan yang tinggi.

2.2 Steganografi

Steganografi berfokus pada teknik penyembunyian pesan atau informasi rahasia. Secara bahasa, kata steganografi berasal dari bahasa Yunani *steganos*, yang berarti "tersembunyi", dan *graphia* yang berarti "tulisan" (Fridrich 2010). Menurut Andono, Sutojo dan Muljono (2017) steganografi merupakan seni untuk menyembunyikan pesan di dalam media digital sedemikian rupa, sehingga orang lain tidak menyadari ada suatu pesan di dalam media tersebut. Dapat dikatakan bahwa steganografi adalah metode penyembunyian pesan dalam suatu media, sehingga keberadaan pesan tersebut tidak diketahui orang lain. Media yang dapat digunakan sebagai penampung pesan rahasia adalah media citra atau gambar, suara, dan video.

Terdapat dua proses utama dalam steganografi, yaitu proses penyisipan pesan (*embedding*) dan proses pengambilan pesan kembali (*extraction*). Proses

embedding dilakukan dengan menyisipkan pesan atau informasi pada media pembawa (*cover*). Proses *embedding* akan menghasilkan media yang berisi pesan rahasia. Proses *extacion* dilakukan untuk mendapatkan kembali pesan atau informasi rahasia dari media pembawa yang telah disisipi pesan sebelumnya. Proses *embedding* dan *extraction* terkadang dilakukan dengan tambahan masukkan berupa kunci (*key*) untuk meningkatkan keamanan steganografi.

Penerapan steganografi kebanyakan dilakukan untuk mengamankan data atau informasi penting dengan cara menyembunyikannya ke dalam suatu media pembawa. Penelitian yang dilakukan oleh Nurdiansyah dan Riftana (2017) menerapkan metode steganografi *Least Significant Bit* (LSB) pada sistem pemberkasan arsip. Pada penelitain tersebut, metode steganografi diterapkan dengan cara menyisipkan informasi rahasia berupa arsip ke dalam citra digital. Sebelum penyisipan, arsip dienkripsi menggunakan algotirma *Twofish* guna meningkatkan keamanan arsip jika sewaktu-waktu ada pihak tak berwenang yang mencoba mengekstrak arsip tersebut. Steganografi dapat pula diterapkan untuk mengirimkan informasi rahasia dari satu pihak ke pihak lain. Caranya dengan menyembunyikan informasi rahasia ke dalam media pembawa, kemudian mengirimkan media tersebut ke pihak yang berhak mengetahuinya.

Ada beberapa kriteria yang harus diperhatikan dalam menggunakan teknik steganografi. Menurut Andono, Sutojo dan Muljono (2017), beberapa kriteria yang perlu diperhatikan dalam penerapan steganografi pada media citra, antara lain:

a. Fidelity

Mutu media citra penampung tidak jauh berubah setelah penambahan pesan rahasia, sehingga media pembawa masih terlihat baik atau mirip dengan aslinya. Pengamat tidak mengetahui jika di dalam media pembawa tersebut terdapat pesan rahasia.

b. Robustness

Pesan yang disembunyikan harus tahan (*robust*) terhadap berbagai operasi manipulasi yang dilakukan pada media pembawa. Contoh manipulasi

seperti pengubahan kontras, penajaman, pemampatan, rotasi, perbesaran gambar, pemotongan (*cropping*), enkripsi dan sebagainya.

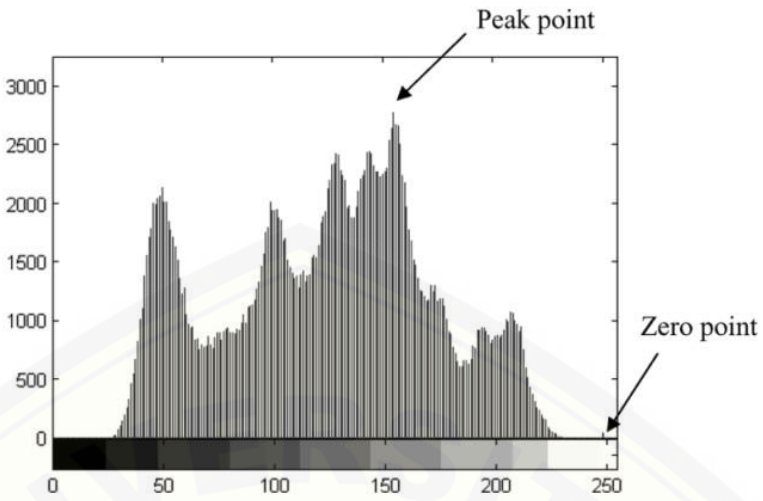
c. Recovery

Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*). Karena tujuan steganografi adalah penyembunyian informasi, maka sewaktu-waktu pesan rahasia di dalam media pembawa harus dapat diambil kembali untuk digunakan lebih lanjut.

2.3 *Histogram Shifting*

Histogram Shifting merupakan salah satu metode steganografi *reversible data hiding* dengan media citra. Hal ini berarti metode ini dapat mengembalikan media pembawa asli setelah pesan rahasia diekstrak atau diambil. *Histogram Shifting* memanfaatkan tingkat kecerahan (*grey level*) dengan frekuensi piksel terendah pada histogram citra untuk menyisipkan pesan rahasia (Ni, et al. 2006). Histogram citra sendiri adalah representasi frekuensi atau jumlah piksel terhadap tingkat kecerahan pada suatu gambar. Sumbu horizontal pada histogram terdiri atas tingkat kecerahan citra, yaitu dari nilai 0 hingga 255, dan sumbu vertikal menunjukkan frekuensi atau jumlah piksel untuk tingkat kecerahan tertentu. Gambar 2.1 menunjukkan contoh histogram suatu citra.

Informasi yang terdapat pada histogram citra dapat digunakan dalam proses penyisipan dan ekstraksi pesan rahasia. Melalui histogram gambar, dapat diketahui tingkat kecerahan dengan frekuensi piksel tertinggi, disebut *peak point*, dan frekuensi terendah, disebut *zero point*. Nilai *peak point* dan *zero point* ini dimanfaatkan untuk menggeser tingkat kecerahan di antara kedua nilai tersebut, pada histogram citra, ke arah nilai *zero point*. Penggeseran bertujuan untuk menyediakan ruang bagi penyisipan pesan rahasia. Penggeseran tingkat kecerahan dilakukan dengan menambah atau mengurangi satu poin tingkat kecerahan. Setelah proses ekstraksi, tingkat kecerahan yang sebelumnya diubah dapat dikembalikan dengan melakukan penambahan atau pengurangan sedemikian hingga tingkat kecerahan kembali ke nilai aslinya.



Gambar 2.1 contoh histogram citra (Ni, et al. 2006)

Tahapan proses penyisipan dan ekstraksi pesan rahasia pada metode *Histogram Shifting* dijelaskan berikut ini (Ni, et al. 2006) :

- Proses Penyisipan
  1. Pertama, buat histogram dari citra yang akan dijadikan media pembawa pesan rahasia
  2. Dari histogram citra yang telah dibuat, tentukan nilai *peak point* dan *zero point*. Kedua nilai ini harus berada di antara rentang 0 sampai 255
  3. Jika *zero point* memiliki frekeunsi lebih dari 0, maka simpan informasi tingkat kecerahan yang ada di samping (kanan atau kiri) *zero point*. Informasi ini dapat disimpan dengan menambahkannya pada bit pesan rahasia. Informasi ini disebut *overhead inforamation*, dan akan digunakan pada proses ekstraksi.
  4. Asumsikan nilai *peak point* lebih kecil dari *zero point*, tambahkan semua piksel dengan tingkat kecerahan di antara *peak* dan *zero* dengan 1 poin. Ini akan menyebabkan histogram citra bergeser ke kanan sebesar 1 poin. Untuk kasus *peak poin* lebih besar dari *zero point*, maka dilakukan pengurangan 1 poin sehingga histogram bergeser ke kiri.

5. Untuk nilai *peak point* lebih kecil dari *zero point*. Pindai citra sekali lagi, jika menemukan piksel dengan tingkat kecerahan sama dengan *peak poin*, periksa nilai bit pesan rahasia yang akan disisipkan. Jika nilai bit tersebut adalah "1", maka tingkat kecerahan pada piksel tersebut ditambah 1 poin. Jika nilai bit adalah "0", maka tingkat kecerahan piksel tersebut tidak diubah. Untuk nilai *peak point* lebih besar dari *zero point*, maka dilakukan pengurangan 1 poin.

- Proses Ekstraksi

  Proses ekstraksi pada *Histogram Shifting* membutuhkan informasi nilai *peak point* dan *zero point* pada proses penyisipan sebelumnya. Diasumsikan nilai *peak point* lebih kecil dari *zero point*, tahapan ekstraksi yaitu:

  1. Pindai citra stego dengan urutan yang sama ketika proses penyisipan. Jika ditemukan piksel dengan tingkat kecerahan sama dengan *peak point* + 1, maka ekstrak nilai bit "1". Jika ditemukan tingkat kecerahan sama dengan *peak point*, maka ekstrak nilai bit "0".
  2. Pindai citra lagi, jika menemukan piksel dengan tingkat kecerahan di antara *peak* dan *zero point*, kurangi tingkat kecerahan tersebut dengan 1 poin.
  3. Jika terdapat *overhead information* pada pesan yang diekstrak, kembalikan tingkat kecerahan di samping *zero point*.

  Untuk *peak point* lebih besar dari *zero point*, maka penambahan diubah menjadi pengurangan begitu juga sebaliknya.

Melalui penjelasan proses penyisipan dan ekstraksi di atas, diketahui metode *Histogram Shifting* memiliki kompleksitas yang cukup rendah. Contohnya pada tahap penyisipan, proses yang dilakukan kebanyakan terletak pada pembuatan histogram, penentuan *peak* dan *zero point*, pemindaian (*scanning*) tingkat kecerahan pada setiap piksel, dan penambahan atau pengurangan nilai keceerahan piksel dengan satu poin. Waktu pemrosesan akan sangat bergantung pada ukuran atau resolusi citra yang digunakan, karena dalam

metode ini dilakukan beberapa kali pemindaian tingkat kecerahan pada setiap piksel. Dalam penelitian Ni, et al. (2006) disebutkan, untuk citra *greyscale* dengan ukuran 512 x 512, waktu penyisipan yang dibutuhkan adalah 100 ms.

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

Peak Signal-to-Noise Ratio merupakan metode yang dapat digunakan untuk mengukur tingkat kualitas citra stego. Al-Najjar dan Soong (2012) menyebutkan bahwa "pengukruan *Signal-to-Noise Ratio* (SNR) adalah estimasi kualitas citra hasil rekonstruksi dibandingkan dengan citra aslinya". Metode PSNR dapat digunakan untuk mengetahui kualitas citra stego yang telah disisipi pesan rahasia dengan cara membandingkannya terhadap citra asli sebelum penyisipan. Satuan yang digunakan pada perhitungan PSNR adalah desibel (dB). Jika nilai PSNR lebih besar dari 36 dB, maka citra asli dan citra hasil rekonstruksi atau steganografi akan terlihat sama (Banarjee, Bhattacharyya dan Sanyal 2013). Persamaan untuk menghitung PSNR yaitu:

$$PSNR = 10 \log \frac{s^2}{MSE} \tag{2.1}$$

Dengan s = 255 untuk citra 8-bit.

Mean Square Error (MSE) dihitung menggunakan persamaan (2.2) di bawah ini:

$$MSE = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} e(m,n)^2 \tag{2.2}$$

Dimana N x M adalah ukuran resolusi gambar, dan e(m,n) adalah perbedaan error antara citra asli dan hasil rekonstruksi. e(m,n) dihitung dengan mengurangi tingkat kecerahan piksel (m,n) citra asli dengan citra hasil rekonstruksi atau citra stego.

Metode PSNR memanfaatkan nilai MSE yang mana merupakan nilai tingkat *error* atau perbedaan antara piksel-piksel pada citra asli dan citra hasil rekonstruksi. Nilai PSNR berbanding terbalik dengan MSE, yang berarti jika nilai MSE semakin besar maka nilai PSNR akan semakin kecil, begitu pula

sebaliknya. Tingginya nilai MSE menunjukkan banyaknya perbedaan tingkat kecerahan antara piksel-piksel pada citra asli dan citra hasil rekonstruksi. Tingginya tingkat perbedaan tersebut akan membuat citra hasil rekonstruksi semakin berbeda dengan aslinya. Dapat dikatakan bahwa semakin tinggi nilai MSE, semakin rendah kualitas atau kemiripan citra hasil rekonstruksi terhadap citra aslinya.

BAB 3. METODOLOGI PENELITIAN

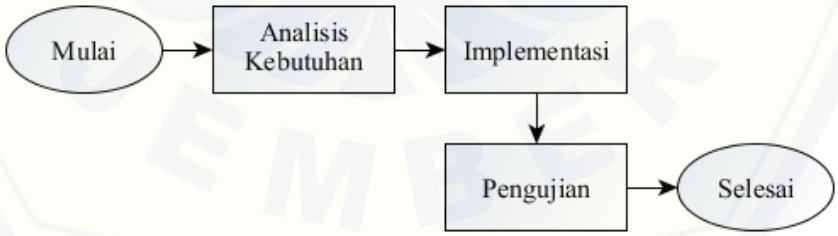
Bab ini akan memaparkan tahapan-tahapan dalam penelitian guna menjawab permasalahan yang ada pada bab awal. Penjelasan tahapan penelitian ini diharapkan juga dapat membuat pengerjaan tugas akhir lebih fokus dan terarah. Bagian ini terdiri dari jenis penelitian yang digunakan dan tahapan penelitian yang akan dilaksankan. Tahapan penelitiannya sendiri terdiri dari analisi kebutuhan, implementasi, dan pengujian.

3.1 Jenis Penelitian

Penelitian ini menggunakan jenis penelitain kuantitatif. Penelitian dengan pendekatan kuantitatif merupakan penelitain yang menggunakan aspek pengukuran, perhitungan rumus, dan kepastian data numerik di hampir semua tahapan penelitiannya (Musianto 2002). Jenis penelitian ini digunakan karena pada penelitian ini dilakukan perhitungan nilai *Peak Signal-to-Noise Ratio* (PSNR) untuk mengukur kualitas citra stego.

3.2 Tahapan Penelitian

Penelitian ini dilaksanakan dalam beberapa tahap seperti ditunjukkan pada Gambar 3.1 di bawah ini



Gambar 3.1 tahapan penelitian

3.3 Analisis Kebutuhan

Tahap pertama ini bertujuan untuk mengumpulkan kebutuhan fungsional dari modul otentikasi yang akan dibuat nantinya. Kebutuhan fungsional yang

dimaksud adalah fitur-fitur yang terdapat pada modul otentikasi ini. Analisi kebutuhan dilakukan secara deskriptif dengan menjelaskan masing-masing fitur. Setiap fitur yang dijabarkan akan disertai dengan data masukkan (*input*), proses pengolahan data, dan keluaran (*output*) dari proses yang dilakukan.

Fitur yang dikembangkan dalam modul otentikasi ini terdiri dari fitur registrasi, *log in*, dan pemulihan kata sandi (citra). Penelitian ini hanya mengembangkan modul otentikasi saja, sehingga hanya ketiga fitur tersebut yang dibuat. Berikut merupakan penjelasan dari ketiga fitur yang ada pada modul otentikasi ini:

a. Registrasi (*Sign Up*)

Fitur registrasi digunakan untuk mendaftarkan akun pengguna baru ke aplikasi *web hositng*. Pada fitur ini, pengguna diharuskan mengisi data diri dan menyertakan gambar atau citra yang nantinya digunakan sebagai media penampung kredensial pengguna. Citra digital yang digunakan harus berekstensi .jpg atau .png. Data yang digunakan sebagai kredensial adalah email dan kata sandi. Proses enkripsi email dan kata sandi pengguna akan menggunakan fungsi enkripsi bawaan dari *framework* Laravel. Proses penyisipan (*embedding*) kredensial akan dilakukan menggunakan metode steganografi *Histogram Shifting*. Setelah pengguna berhasil melakukan pendaftaran, citra stego akan dikembalikan ke pengguna untuk digunakan dalam proses masuk ke aplikasi (*Log In*).

Tabel 3.1 Masukkan, proses, dan keluaran fitur Registrasi (Sign Up)

Masukkan	**Proses**	**Keluaran**
- Email - Nama - Kata sandi - Tanggal lahir - Jenis Kelamin - No HP - Citra digital (.jpg atau .png)	- Penyimpanan data pengguna ke basis data - Enkripsi kredensial - Penyisipan kredensial terenkripsi ke citra digital menggunakan metode *Histogram Shifting*	- Citra stego (.png) yang telah disisipi kredensial pengguna - Pengguna masuk ke akun miliknya

b. *Log In*

Fitur *Log In* digunakan untuk masuk ke akun pengguna. Data masukkan yang diperlukan adalah citra stego yang didapat ketika registrasi. Proses ekstraksi akan dilakukan untuk mendapat data kredensial pengguna dalam bentuk yang terenkripsi. Kredensial yang telah diekstrak akan dikembalikan ke bentuk aslinya terlebih dahulu menggunakan fungsi dekripsi bawaan dari *framework* Laravel. Setelah didapatkan kredensial pengguna, dilakukan pencocokan dengan data yang disimpan pada basis data. Jika kredensial sesuai dengan data di basis data, maka pengguna dapat masuk ke akun miliknya.

Tabel 3.2 Masukkan, proses, dan keluaran fitur Log In

Masukkan	**Proses**	**Keluaran**
- Citra stego (.png)	- Ekstraksi kredensial dari citra stego menggunakan metode *Histogram Shifting* - Dekripsi kredensial - Pencocokan kredensial dengan data di basis data	- Pengguna masuk ke akun miliknya

c. Pemulihan Citra

Fitur pemulihan diperuntukan bagi pengguna yang kehilangan citra stego atau ketika citra stego mengalami kerusakan sehingga tidak dapat digunakan untuk *Log In*. Pengguna diharuskan memasukkan email dan tanggal lahir ketika ingin memulihkan citra stego. Email dan tanggal lahir akan dicocokan terlebih dahulu dengan data di basis data. Jika sesuai, maka URL pemulihan akan dikirim ke email pengguna. Setelah membuka URL tersebut, pengguna akan diminta untuk memasukkan kata sandi dan citra baru untuk menggantikan kata sandi dan citra lama. Proses enkripsi dan penyisipan kredensial akan dilakukan seperti ketika pengguna melakukan registrasi. Setelah selesai, pengguna dapat masuk ke akun miliknya lagi, dan citra stego baru juga akan diberikan ke pengguna untuk *Log In* selanjutnya.

Tabel 3.3 Masukkan, proses, dan keluaran fitur Pemulihan Kata Sandi/Citra

Masukkan	Proses	Keluaran
- Email - Tanggal Lahir - Kata sandi baru - Citra digital baru (.jpg, .png)	- Pencocokan email dan tanggal lahir dengan data di basis data - Pengiriman URL pemulihan ke email pengguna - Enkripsi kredensial - penyisipan kredensial terenkripsi ke dalam citra digital menggunakan metode *Histogram Shifting*	- Citra stego (.png) baru yang telah disisipi kredensial pengguna - Pengguna masuk ke akun miliknya

3.4 Implementasi

Tahap implementasi dilakukan untuk mengembangkan modul otentikasi dengan menerapkan metode *Histogram Shifting* sesuai analisis kebutuhan yang dijelaskan sebelumnya. Tahap ini akan menerapkan langsung metode steganografi ke dalam kode program. Kode program yang ditulis akan berbasis web dengan memanfaatkan *framework* Laravel versi 5.8. Penulisan kode program dilakukan menggunakan aplikasi *Sublime Text* 3. Basis data yang digunakan pada pengembangan modul otentikasi ini adalah DBMS *Mysql*.

3.5 Pengujian

Tahap pengujian terdiri dari beberapa pengujian, antara lain pengujian citra hasil setaganografi, pengujian waktu eksekusi proses regsitrasi dan *log in*, serta pengujian pengaruh enkripsi kredensial. Pengujian-pengujian tersebut dijelaskan sebagai berikut

a. Pengujian Citra Stego

Tahap Pengujian ini dilakukan untuk menguji citra hasil steganografi (citra stego) berdasarkan beberapa kriteria pengujian. Kriteria pengujian yang digunakan adalah kualitas citra, ketahanan (*robustness*) pesan rahasia, dan

kemampuan pengungkapan (*recovery*) pesan rahasia. Ketiga kriteria pengujian tersebut dijelaskan sebagai berikut:

1. Kualitas Citra

   Pengujian ini dilakukan untuk mengetahui kualitas citra yang dihasilkan dari proses steganografi menggunakan metode *Histogram Shifting*. Pengujian ini menggunakan metode *Peak Signal-to-Noise Ratio* (PSNR) seperti yang dijelaskan pada bab tinjauan pustaka. Standar nilai PSNR agar citra stego terlihat sama dengan citra aslinya adalah 36 dB (Banarjee, Bhattacharyya dan Sanyal 2013). Pengujian akan dilakukan pada beberapa citra dengan dimensi atau resolusi yang berbeda.

2. Ketahanan (*robustness*)

   Pengujian ini dilakukan untuk mengetahui ketahanan pesan rahasia yang telah disisipkan ke dalam suatu citra terhadap berbagai teknik manipulasi citra (*image processing*). Beberapa teknik manipulasi citra yang digunakan yaitu pemotongan (*cropping*), rotasi (*rotate*), pengubahan ukuran citra (*resize*), dan pengubahan kontras warna (*contrast adjustment*). Pengujian ini akan menerapkan teknik-teknik tersebut pada citra hasil setganografi, lalu pesan rahasia akan coba diekstrak kembali dengan memanfaatkan fitur *log in*. Jika citra stego yang telah dimanipulasi dapat digunakan untuk masuk ke akun pengguna, maka pesan rahasia tahan terhadap teknik manipulasi yang dilakukan.

3. Kemampuan Pengungkapan (*recovery*)

   Pengujian ini dilakukan guna mengetahui apakah pesan rahasia yang telah disisipkan pada citra stego dapat diungkap atau diekstrak kembali. Pengujian ini akan memanfaatkan fitur *log in* untuk melakukan ekstraksi pesan rahasia berupa kredensial pengguna. Jika citra stego dapat digunakan untuk masuk ke akun pengguna, maka pesan rahasia berhasil diungkap atau diekstrak. Ketika citra stegano tidak dapat diekstrak, karena terdapat kerusakan pada citra, maka pengguan dapat menggunakan fitur pemulihan citra untuk mendapatkan citra stegano baru.

b. Pengujian Waktu Eksekusi

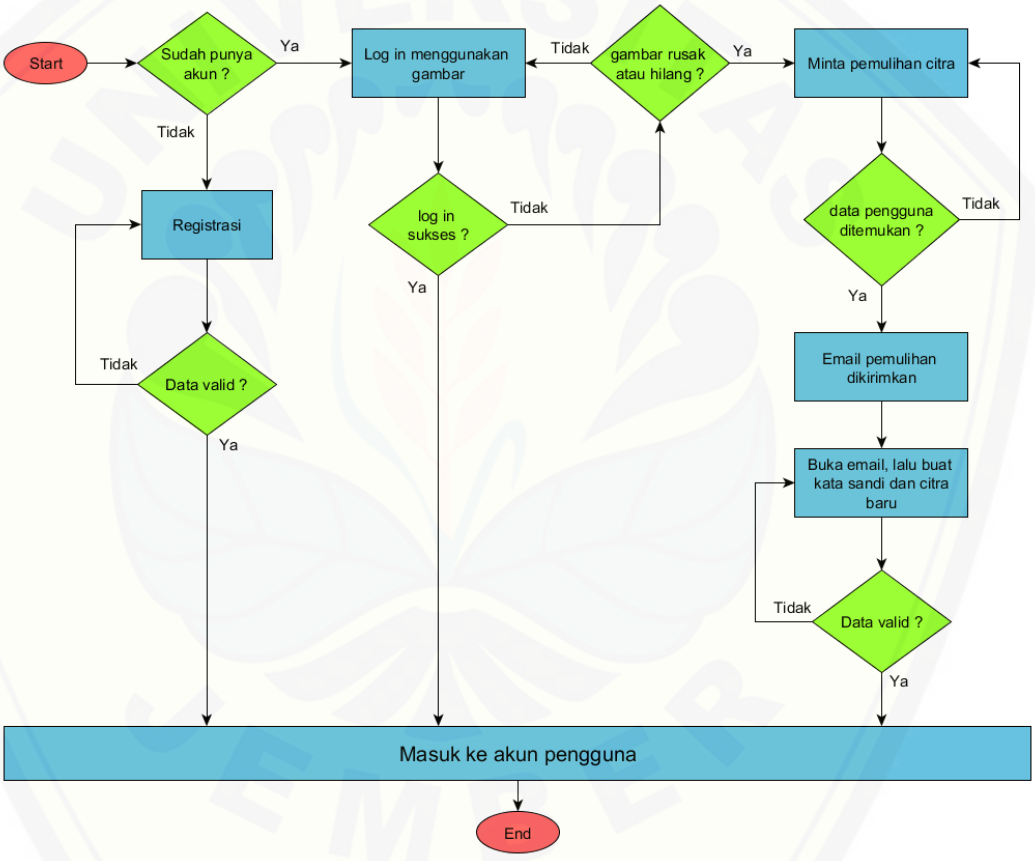
Pengujian ini bertujuan untuk mengetahui waktu eksekusi rata-rata pada proses registrasi dan *log in*. Proses yang diuji adalah proses yang terjadi pada bagian server (*server-side*), yaitu pada kode program PHP yang dijalankan di server. Pengujian ini dilakukan dengan menghitung lamanya kode program suatu proses dijalankan atau dieksekusi dalam satuan detik. Pengujian akan dilakukan beberapa kali menggunakan citra yang sama untuk mengetahui rata-rata waktu eksekusi masing-masing proses.

c. Pengujian Pengaruh Enkripsi

Pengujian ini bertujuan guna mengetahui pengaruh proses enkripsi dan dekripsi kredensial pengguna terhadap keamanan akun pengguna. Pengujian akan terdiri dari dua skenario, yang petama menerapkan proses enkripsi dan dekripsi dalam modul otentikasi, sementara skenario kedua tidak menerapkan kedua proses tersebut. Pada masing-masing skenario pengujian tersebut akan dibuat sebuah akun baru dengan kata sandi yang sama. Kemudian, kredensial terdaftar yang digunakan untuk membuat akun baru tersebut disisipkan pada citra lain yang berbeda dengan citra pada pembuatan akun sebelumnya. Penyisipan kredensial terdaftar tidak menerapkan proses enkripsi. Citra stego hasil penyisipan tanpa enkripsi kemudian dicoba digunakan untuk masuk ke akun pengguna menggunakan fitur *log in*. Percobaan masuk ke akun pengguna dilakukan pada kedua skenario pengujian. Melalui pengujian ini, dapat diketahui apakah proses enkripsi dan dekripsi dapat memberi perlindungan pada akun pengguna dalam kasus kredensial pengguna diketahui orang lain dan digunakan untuk membuat citra stego lain.

BAB 4. IMPLEMENTASI

Bab ini akan menjelaskan mengenai penerapan metode steganografi *Histogram Shifting* dalam kode program sesuai dengan analisis kebutuhan. Penerapan kode program berbasis web dengan menggunakan *framework* Laravel versi 5.8. Penjelasan implementasi dilengkapi dengan diagram alir (*flowchart*) untuk setiap fitur.



Gambar 4.1 *userflow* diagram untuk modul otentikasi yang dikembangkan

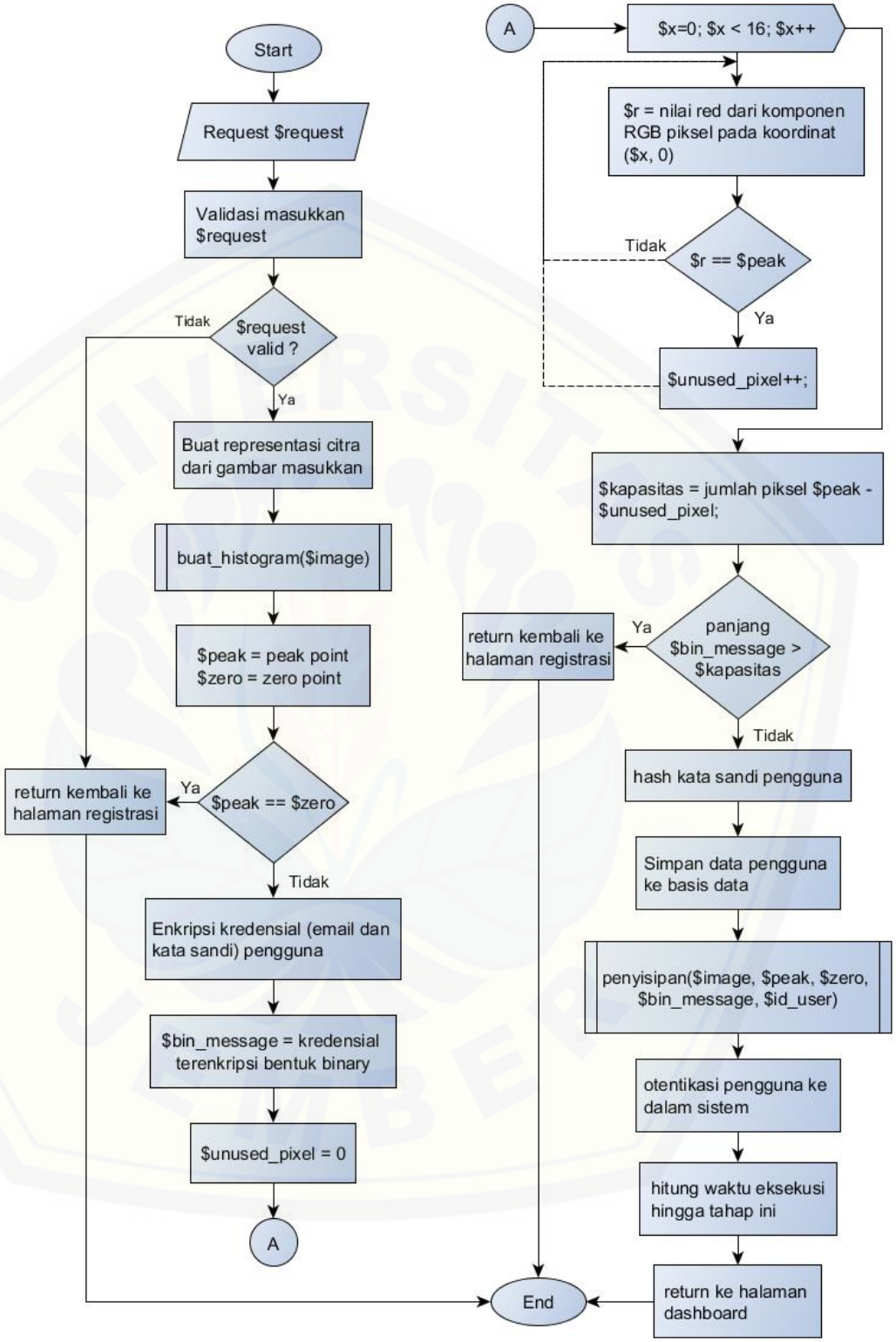
Modul otentikasi yang dikembangkan terdiri dari tiga fitur, yaitu fitur registrasi, *log in*, dan pemulihan citra. Gambar 4.1 di atas menunjukkan *userflow* diagram atau alur kerja pada modul otentikasi ini. Pengguna baru yang belum memiliki akun dapat menggunakan fitur registrasi untuk membuat akun baru.

Setelah mengisi formulir registrasi, pengguna akan langsung diarahkan ke halaman *dashboard* miliknya. Di halaman tersebut, pengguna dapat mengunduh citra stego untuk digunakan ketika *log in* ke akun miliknya. Pengguna yang telah memiliki akun dapat masuk ke akun miliknya menggunakan fitur *log in*. Fitur *log in* hanya memerlukan masukkan berupa gambar atau citra digital yang diperoleh setelah melakukan registrasi. Jika citra benar dan tidak mengalami kerusakan, maka pengguna dapat langsung masuk ke halaman *dashboard*. Namun jika citra pengguna tidak dapat digunakan untuk masuk ke akun miliknya, maka pengguna dapat memanfaatkan fitur pemulihan citra untuk memperbarui kata sandi dan citra miliknya. Fitur pemulihan citra membutuhkan masukkan email dan tanggal lahir pengguna yang digunakan ketika mendaftarkan akun. Jika email dan tanggal lahir yang dimasukkan sesuai dengan data di basis data, maka sistem akan mengirimkan email berisi URL pemulihan citra ke alamat email pengguna. Setelah membuka URL yang dikirmkan, pengguna diharuskan memasukkan kata sandi dan citra baru. Setelah berhasil, maka pengguna akan diarahkan ke halaman *dashboard* miliknya, dimana pengguna dapat mengunduh citra stego baru untuk digunakan masuk ke akunnya lain waktu.

4.1 Registrasi (*Sign Up*)

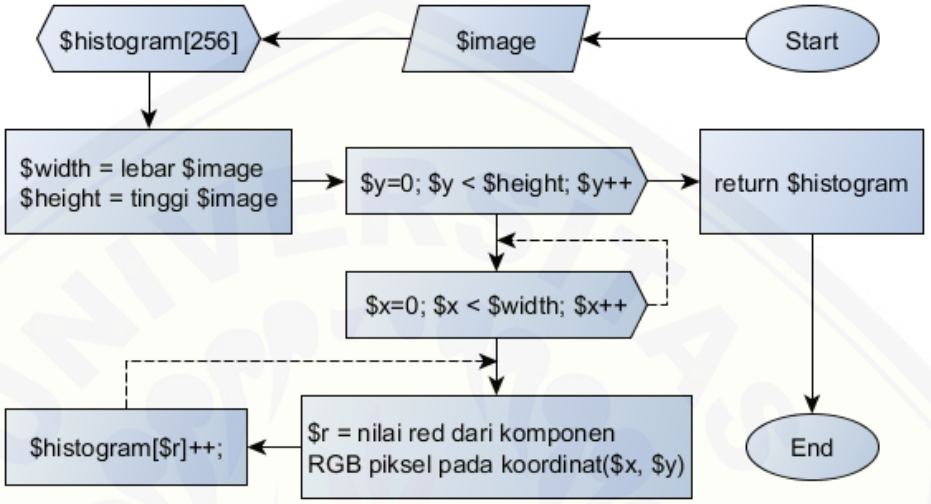
Fitur registrasi digunakan untuk mendaftarkan akun pengguna baru. Proses pada fitur regstrasi terdiri dari beberapa tahap, mulai dari validasi data pengguna hingga pengembalian citra hasil stegano ke pengguna melalui fungsi unduh citra stegano. Proses registrasi dimulai dari fungsi *store_user* yang ditunjukkan oleh *flowchart* pada Gambar 4.2 di bawah.

Proses regsitrasi diawali dengan memvalidasi data pengguna yang dikirim ke server. Proses validasi memanfaatkan fungsi bawaan *validate* dari Laravel. Tahap validasi memastikan data yang dimasukkan oleh pengguna telah sesuai dengan ketetapan sistem. Contohnya seperti citra yang harus berekstensi .jpeg atau .png. Jika terdapat data pengguna yang tidak sesuai dengan standar validasi, maka program akan mengembalikan ke halaman registrasi. Jika data telah sesuai, maka akan dilanjutkan ke tahap berikutnya.



Gambar 4.2 *flowchart* fungsi *store_user*

Citra yang diunggah pengguna selanjutnya dibuatkan representasinya agar citra dapat diolah atau dimanipulasi lebih lanjut. Representasi citra selanjutnya digunakan untuk membuat histogram. *Flowchart* pembuatan histogram citra dapat dilihat pada Gambar 4.3 di bawah.



Gambar 4.3 *flowchart* fungsi pembuatan histogram

Fungsi pembuatan histogram memerlukan parameter berupa representasi citra, yang mana pada *flowchart* ditunjukkan dengan variabel $image. Langkah pertama pembuatan histogram adalah membuat *array* $histogram dengan panjang 256. Panjang ini sesuai dengan nilai tingkat kecerahan (*grey level*) yang bernilai mulai dari 0 hingga 255. Selanjutnya simpan lebar dan tinggi citra untuk digunakan ketika memindai citra. Pemindaian citra dilakukan dengan membuat dua perulangan (*loop*) sesuai lebar dan tinggi citra. Untuk setiap piksel yang dipindai, simpan nilai tingkat kecerahan komponen warna merah pada variabel $r. Pada penelitian ini, komponen warna yang digunakan untuk menyisipkan pesan rahasia adalah warna merah (*Red*), sehingga untuk tahap selanjutnya akan terus menggunakan komponen warna ini. Selanjutnya, elemen *array* $histogram dengan indeks sama dengan nilai $r akan ditambahkan satu nilai. Sehingga pada akhirnya, indeks *array* $histogram akan menunjukkan tingkat kecerahan, dan

nilai elemen *array* merupakan jumlah piksel untuk masing-masing tingkat kecerahan tersebut. Fungsi ini kemudian mengembalikan *array* $histogram.

Tahap berikutnya yaitu menentukan nilai *peak* dan *zero point*. Kedua nilai ini akan digunakan ketika menyisipkan pesan rahasia ke citra penampung nantinya. *Peak point* adalah nilai tingkat kecerahan dengan jumlah piksel paling banyak, dan *zero point* adalah kebalikannya. Kedua nilai tersebut terdapat pada indeks array histogram. *Peak point* ditunjukkan oleh indeks *array* dengan nilai elemen tertinggi, dan *zero point* ditunjukkan oleh indeks *array* dengan nilai elemen terendah. Pengecekan akan dilakukan untuk memastikan nilai *peak* dan *zero point* tidak sama, karena penyisipan tidak dapat dilakukan jika kedua nilai sama. Sistem akan mengembalikan ke halaman registrasi ketika menjumpai citra dengan nilai *peak* dan *zero point* yang sama.

Berikutnya, kredensial pengguna yang terdiri dari email dan kata sandi dienkripsi terlebih dahulu. Kredensial ini berperan sebagai pesan rahasia yang nantinya disisipkan ke citra. Kredensial pengguna dienkripsi terlebih dahulu sebelum disipkan ke citra pembawa. Proses enkripsi menggunakan fungsi bawaan Laravel *encrypt.* Enkripsi dilakukan agar kredensial pengguna lebih aman jika suatu saat orang lain mencoba mengekstraknya. Pesan rahasia terenkripsi kemudian diubah ke bentuk binary untuk kebutuhan penyisipan.

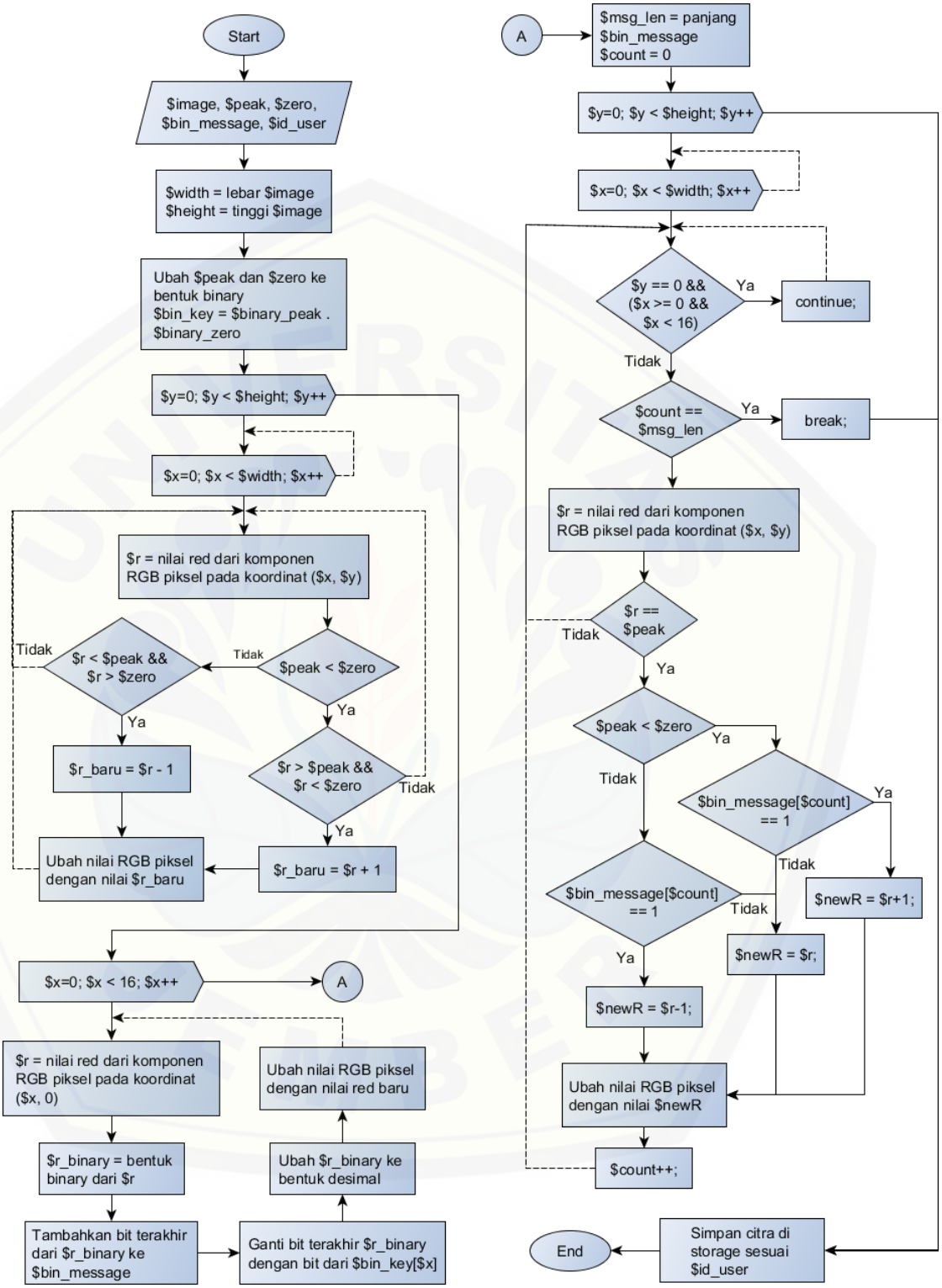
Sebelum penyisipan pesan rahasia, citra akan diperiksa dahulu apakah memiliki kapasitas yang cukup. 16 piksel pertama pada citra akan digunakan untuk menyisipkan nilai *peak* dan *zero point*, sehingga jika pada 16 piksel pertama tersebut terdapat piksel dengan nilai komponen warna merah sama dengan *peak point*, maka piksel tersebut tidak dapat digunakan untuk menyisipkan pesan rahasia. pertama, pindai 16 piksel pertama secara horizontal dan dapatkan nilai komponen warna merah untuk setiap piksel tersebut. Jika nilai warna merah sama dengan *peak point*, maka tambahkan nilai 1 pada variabel $unused_pixel. Variabel ini akan menyimpan banyaknya piksel *peak point* yang tidak dapat diguankan untuk menyisipkan pesan rahasia. Lalu, kurangkan jumlah piksel *peak point* dengan nilai variabel $unused_ pixel. Variabel $kapasitas berisi jumlah piksel yang dapat digunakan untuk menyisipkan pesan rahasia.

Dilakukan pengecekan apakah panjang bit pesan rahasia lebih besar dari pada nilai variabel $kapasitas. Jika bit pesan rahasia lebih panjang, maka citra tidak dapat digunakan dan sistem akan mengembalikan ke halaman registrasi.

Sebelum data pengguna disimpan, akan dilakukan *hash* terlebih dahulu pada kata sandi. Proses *hash* memanfaatkan fungsi bawaan *hash* dari Laravel. Fungsi tersebut menggunakan metode *hash* bcrypt. Setelah itu, data pengguna akan disimpan ke basis data. Id pengguna baru akan didapatkan dan digunakan untuk keperluan penyisipan nantinya.

Langkah berikutnya, dilakukan penyisipan pesan rahasia menggunakan fungsi *penyisipan*. Fungsi *penyisipan* membutuhkan parameter representasi citra, *peak point*, *zero point*, bit pesan rahasia, dan id pengguna baru. Keseluruhan alur proses penyisipan dapat dilihat pada *flowchart* Gambar 4.4 di bawah.

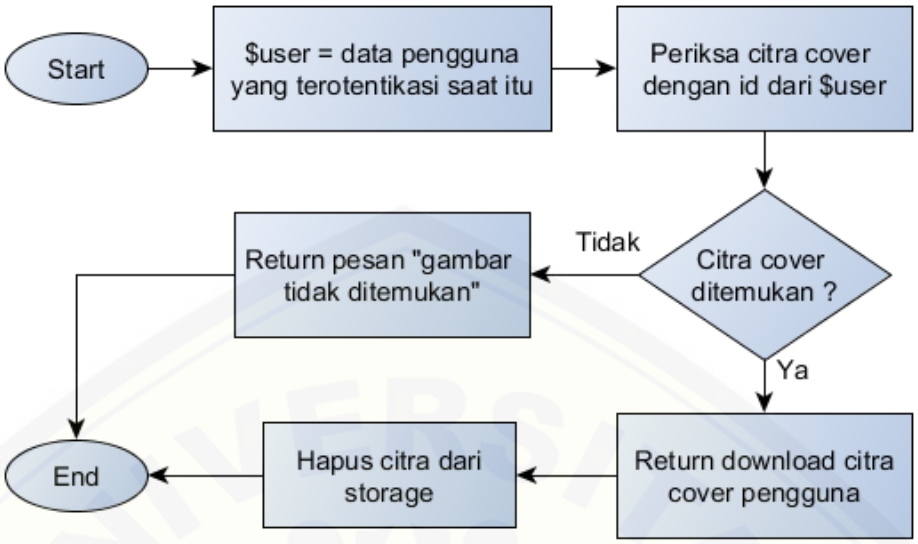
Langkah pertama pada fungsi penyisipan adalah mendapatkan lebar dan tinggi citra. Kedua nilai tersebut akan digunakan untuk memindai citra nantinya. Lalu, ubah nilai *peak* dan *zero point* ke bentuk binary dan gabungkan menjadi satu *string*. Variabel $bin_key menyimpan gabungan nilai binary *peak* dan *zero point*. Selanjutnya, lakukan penggesaran histogram dengan cara memindai citra dan simpan nilai komponen warna merah untuk suatu piksel pada variabel $r. Untuk *peak point* lebih kecil dari *zero point*, jika nilai $r berada di antara *peak* dan *zero point*, tambahkan 1 poin pada nilai $r dan simpan sebagai nilai *red* baru. Untuk *peak point* lebih besar dari *zero point*, jika nilai $r berada di antara *peak* dan *zero point*, kurangi 1 poin pada nilai $r dan simpan sebagai nilai *red* baru. Ubah nilai RGB piksel dengan nilai *red* baru tadi. Selanjutnya, pindai 16 piksel horizontal pertama untuk menyisipkan nilai variable $bin_key. Penyisipan nilai binary *peak* dan *zero point* dilakukan untuk kebutuhan proses ekstraksi nantinya. Dapatkan nilai komponen warna merah suatu piksel, lalu ubah nilai tersebut ke bentuk binary dan simpan pada variabel $r_binary. Tambahkan nilai bit terakhir $r_binary ke bit pesan rahasia $bin_message, lalu ganti nilai bit terakhir tadi dengan bit dari variable $bin_key[$x].



Gambar 4.4 *flowchart* fungsi penyisipan

Ubah nilai $r_binary ke bentuk desimal kembali. Kemudian ubah nilai RGB piksel saat itu dengan nilai desimal baru tadi. Langkah selanjutnya, simpan panjang bit pada $bin_message dan buat variabel $count untuk nantinya menetukan batasan perulangan. Penyisipan pesan rahasia dimulai dengan memindai citra dari awal. 16 piksel horizontal pertama akan dilewati karena sudah digunakan sebelumnya untuk meyisipkan nilai *peak* dan *zero point*. Simpan nilai komponen warna merah suatu piksel pada variabel $r. Untuk nilai $r sama dengan *peak point*, jika nilai *peak point* kurang dari *zero point*, periksa apakah nilai bit pada $bin_message untuk indeks $count saat itu sama dengan 1. Jika benar, maka tambahkan nilai 1 point pada variabel $r dan simpan ke variabel $newR. Sebaliknya, jika nilai *peak point* lebih besar dari *zero point* dan nilai bit pada $bin_message untuk indeks $count saat itu sama dengan 1, maka kurangi nilai 1 poin pada variabel $r dan simpan ke variabel $newR. Untuk nilai bit $bin_message sama dengan 0, tidak perlu mengubah nilai $r. Ubah nilai RGB piksel saat itu dengan nilai komponen warna merah menggunakan nilai $newR. Tambahkan nilai $count dengan 1 poin. Setelah itu, simpan citra yang telah disisipkan pesan rahasia ke *storage* sesuai nomor id pengguna pada variabel $id_user. Citra disimpan dengan ekstensi .png. Sebagai catatan, Proses penyispan ini tidak disertai dengan penyimpanan *overhead information* yang mana berguna untuk mengembalikan citra asli pada proses ekstraksi. Proses ekstraksi nantinya hanya sebatas pengambilan pesan rahasia kembali, tanpa mengembalikan citra asli. Hal ini dilakukan untuk mempersingkat waktu eksekusi proses penyisipan dan ekstraksi.

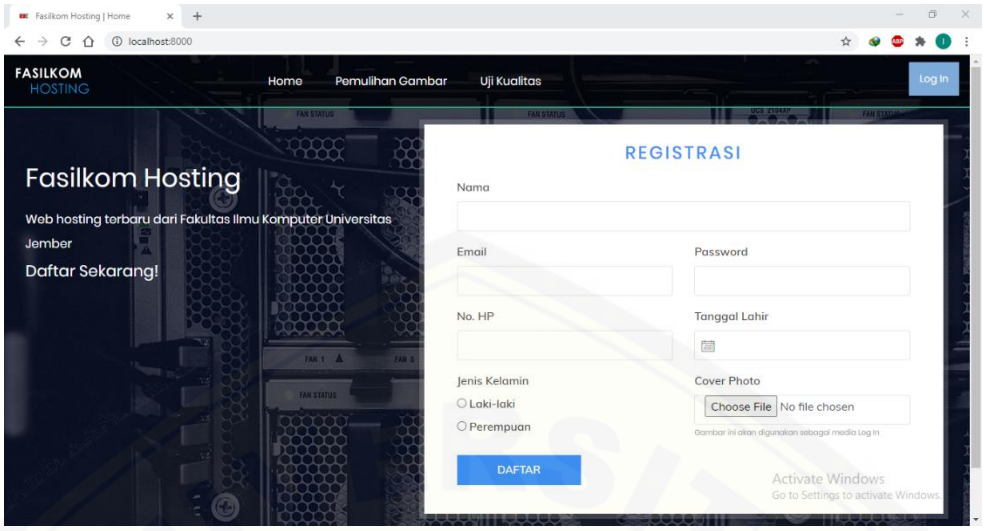
Setelah kredensial pengguna berhasil disisipkan, pengguna lalu diotentikasi ke sistem. Sebelum di arahkan ke halaman *dashboard,* dilakukan perhitungan lama waktu eksekusi proses registrasi tersebut. Perhitungan dimulai dari terbentuknya *request* pada server hingga sebelum diarahkan ke halaman *dashboard*. Perhitungan waktu ini dilakukan terkait kebutuhan untuk pengujian eksekusi waktu nantinya.



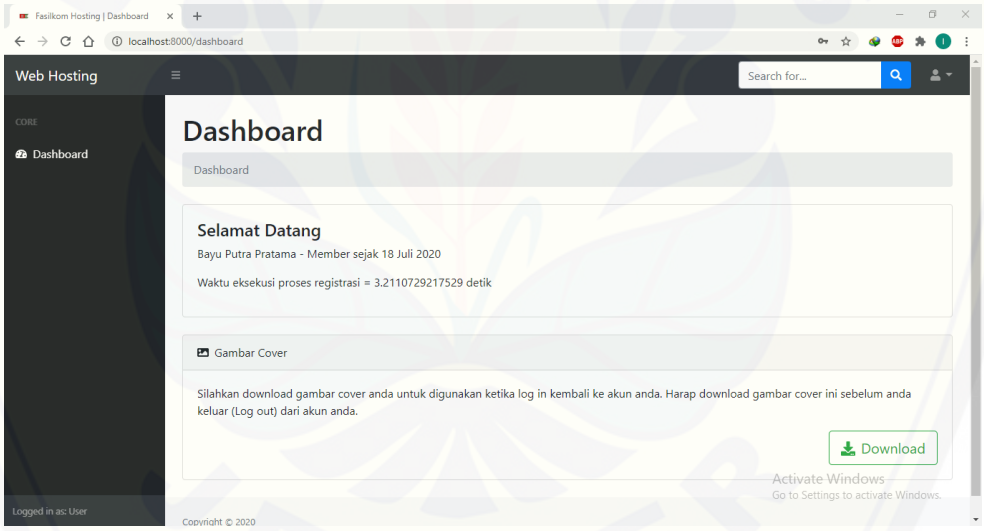
Gambar 4.5 *flowchart* fungsi *download_cover*

Pengguna dapat mengunduh citra stego miliknya melalui tombol "Download" pada halaman *dashboard* ketika baru saja mendaftarkan akun. Fungsi *download_cover* akan dijalankan ketika tombol "Download" diklik. Gambar 4.5 di atas menunjukkan *flowchart* fungsi *download_cover*. Langkah pertama, dapatkan data pengguna yang terotentikasi saat itu. Lalu, periksa citra stego di *storage* sesuai id pengguna saat itu. Jika ditemukan citra dengan id pengguna tersebut, maka sistem akan mengirim respon berupa *file* citra stego ke *browser* pengguna. Jika citra tidak ditemukan, maka fungsi ini akan mengembalikan pesan eror. Setelah diunduh, citra akan dihapus dari *storage*.

Berikut merupakan cuplikan layar halaman-halaman yang ada pada fitur registrasi. Gambar 4.6 menunjukkan halaman regsitrasi yang berisi formulir pendaftaran. Sementara Gambar 4.7 menunjukkan halaman *dashboard* pengguna baru setelah melakukan registrasi.



Gambar 4.6 cuplikan layar halaman registrasi

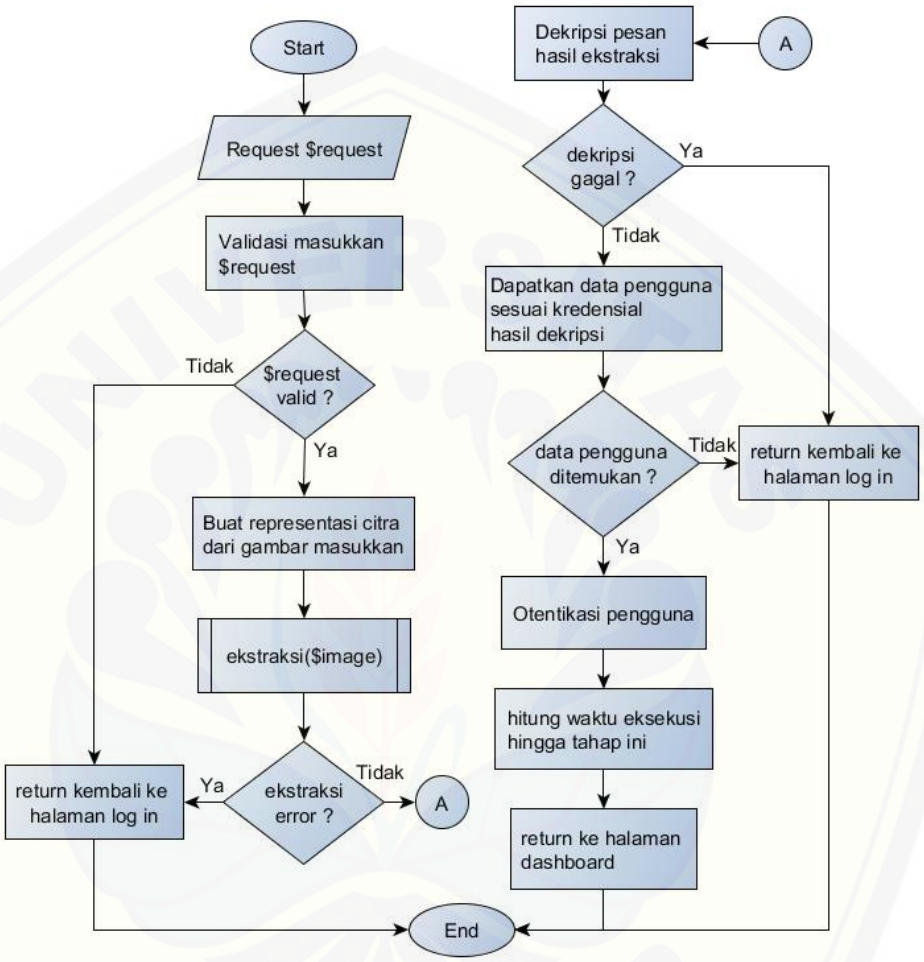


Gambar 4.7 cuplikan layar halaman *dashboard* setelah registrasi

4.2 *Log In*

Fitur *log in* digunakan untuk masuk ke akun pengguna. Kredensial pengguna yang disispkan ke dalam citra akan diekstrak dan digunakan untuk mengotentikasi pengguna. Proses ekstraksi pesan rahasia pada fitur ini hanya sebatas mengambil pesan yang sebelumnya disisipkan, tanpa mengembalikan citra asli. Hal tersebut dilakukan untuk mempercepat proses otentikasi

pengguna. Fungsi *check_login* merupakan fungsi yang menjalankan fitur *log in*. Gambar 4.8 di bawah menunjukkan *flowchart* fungsi *check_login*.

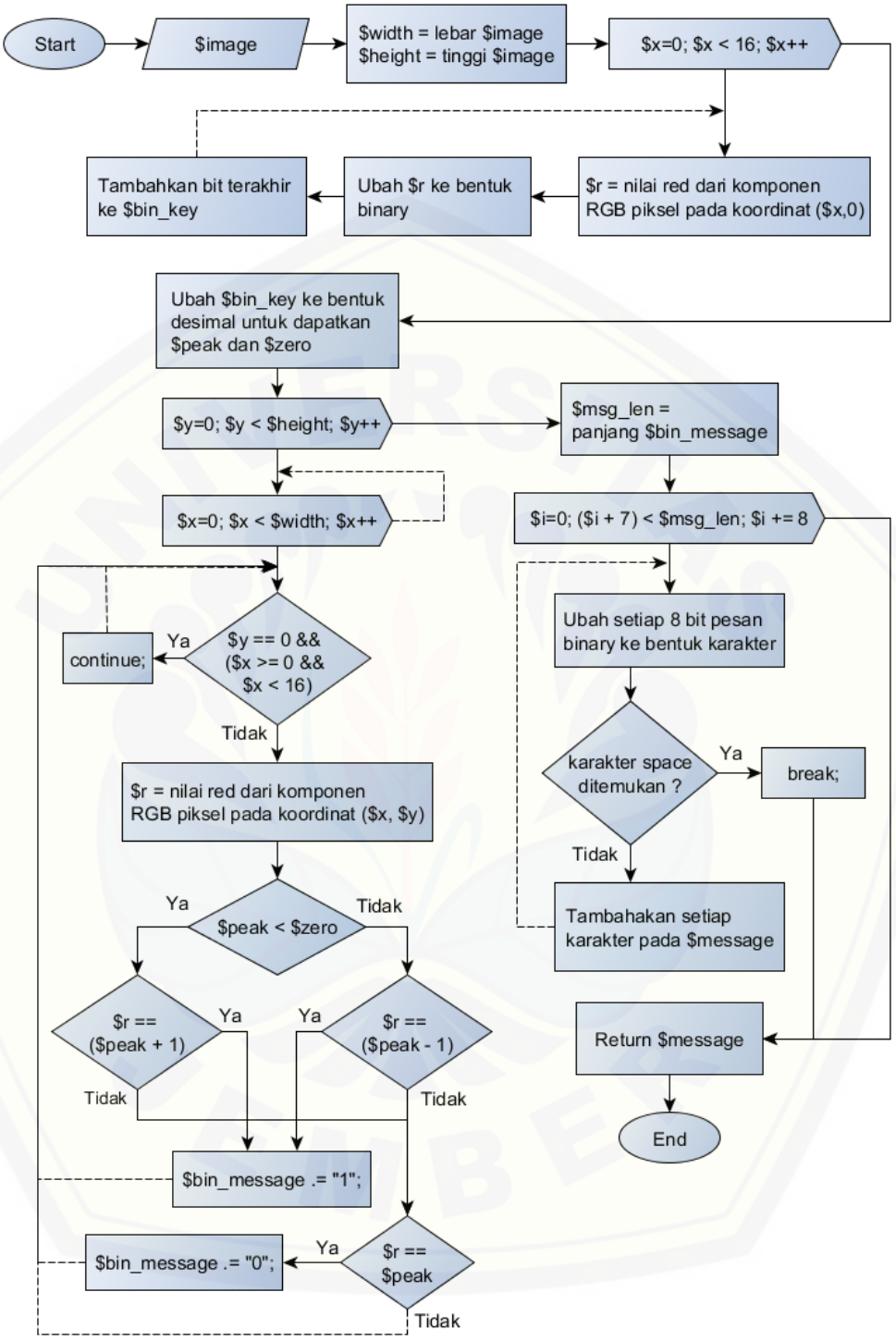


Gambar 4.8 *flowchart* fungsi *check_login*

Proses *log in* diawali dengan melakukan validasi terhadap *file* citra yang diunggah pengguan. Validasi dilakukan dengan memanfaatkan fungsi bawaan *validate* dari Laravel. Valdiasi yang dilakukan yaitu untuk memastikan *file* yang diunggah adalah benar *file* citra dengan ekstensi .png. Jika citra yang diunggah tidak sesuai dengan standar validasi, maka program akan mengembalikan ke tampilan *log in*. Setelah divalidasi, citra akan dibuatkan representasinya untuk selanjutnya diolah atau dimanipulasi.

Tahap berikutnya adalah ekstraksi pesan rahasia. Tahap ini menggunakan fungsi *ekstraksi* untuk mendapatkan pesan rahasia berupa kredensia terenkripsi milik pengguna. Pengecekan dilakukan setelah ekstraski untuk memastikan tidak terjadi kesalahan dalam proses ekstraksi. Fungsi ekstraksi membutuhkan parameter berupa representasi citra yang ditunjukkan oleh variabel $image pada *flowchart*. Keseluruhan tahapan fungsi ektraksi dapat dilihat pada Gambar 4.9 dibawah.

Fungsi ekstraksi diawali dengan mendapatkan lebar dan tinggi citra untuk keperluan pemindaian citra nantinya. Selanjutnya, ekstrak nilai *peak* dan *zero point* dengan memindai 16 piksel horizontal pertama. Dapatkan nilai komponen warna merah untuk setiap piksel dan simpan pada variabel $r. Ubah nilai variabel $r ke bentuk binary, lalu tambahkan nilai bit terakhir ke variabel $bin_key. Ubah nilai binary $bin_key ke bentuk desimal. 8 bit pertama merupakan bit nilai *peak point*, sementara 8 bit selanjutnya adalah bit nilai *zero point*. Selanjutnya, pindai keseluruhan citra untuk mengekstrak bit pesan rahasia, kecuali 16 piksel horizontal pertama. Gunakan dua perulangan (*looping*) dengan batasan lebar dan tinggi citra untuk memindai citra. Dapatkan nilai komponen warna merah suatu piksel dan simpan pada varaibel $r. Untuk *peak point* lebih kecil dari *zero point*, jika $r sama dengan nilai *peak point* + 1, maka ekstrak nilai bit "1" dan tambahkan pada variabel $bin_message. Untuk *peak point* lebih besar dari *zero point*, jika $r sama dengan nilai *peak point* - 1, maka ekstrak nilai bit "1" dan tambahkan pada variabel $bin_message. Jika nilai $r sama dengan *peak point*, maka ekstrak nilai bit "0" dan tambahkan pada variabel $bin_message. Selanjutnya, ubah pesan rahasia dari bentuk binary ke bentuk teks atau *string*. Lakukan perulangan untuk mengambil setiap 8 bit pesan rahasia, kemudia ubah ke bentuk karakter (*char*). 8 bit pesan rahasia akan membentuk 1 karakter. Jika karakter yang didapat sama dengan karakter spasi, maka hentikan perulangan. Tambahkan setiap karakter ke varaibel $message, yang mana akan berisikan kredensial pengguna dalam bentuk terenkripsi pada akhirnya. Fungsi ekstraksi akan megembalikan nilai varaibel $message tersebut.

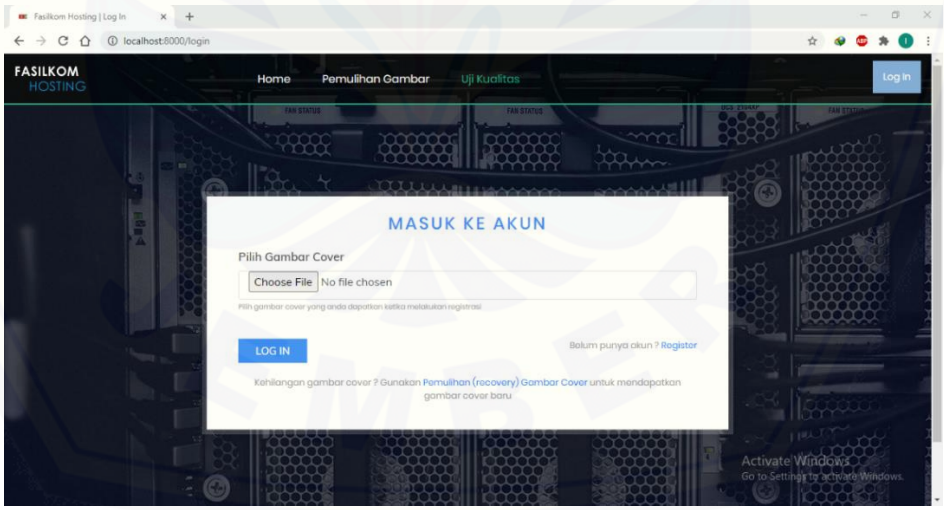


Gambar 4.9 *flowchart* fungsi ekstraksi

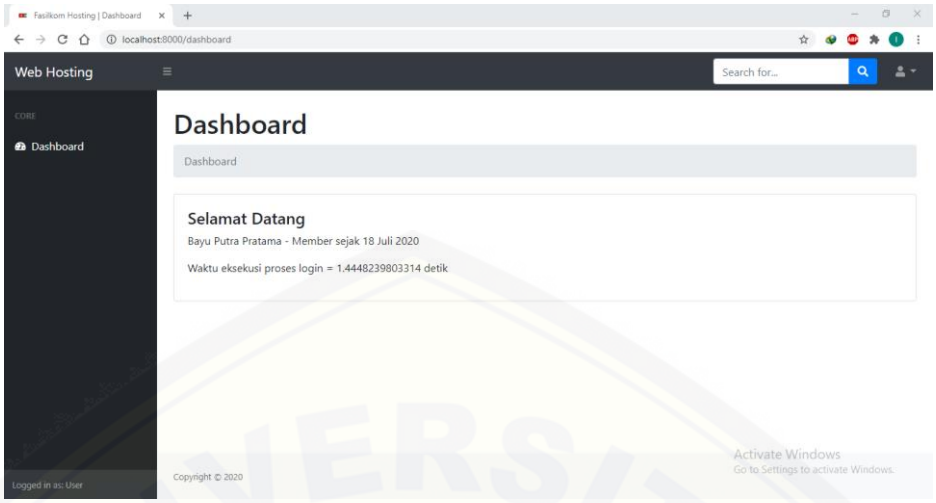
Kredensial yang telah diekstrak selanjutnya didekripsi menggunakan fungsi Laravel *decrypt*. Jika terjadi kesalahan selama proses dekripsi, maka sistem akan mengarahkan ke halaman *log in* dengan pesan eror. Hasil dekripsi berupa email dan kata sandi yang dipisahkan oleh spasi dalam bentuk teks atau *string*.

Email dan kata sandi yang telah didapatkan selanjutnya dicocokkan dengan data di basis data. Jika ditemukan data pengguna yang sesuai, maka pengguna akan diotentikasi ke sistem. Namun jika data pengguna tidak ditemukan, maka sistem akan mengembalikan ke halaman log in dengan pesan eror. Selanjutnya hitung lamanya waktu eksekusi proses *log in*, dimulai sejak terjadinya *request* pada server hingga sebelum diarahkan ke halaman dashboard. Perhitungan waktu ini dilakukan terkait kebutuhan untuk pengujian eksekusi waktu nantinya.

Berikut merupakan cuplikan layar untuk halaman-halaman pada fitur *log in*. Gambar 4.10 di bawah menunjukkan cuplikan layar tampilan halaman *log in*. Gambar 4.11 menujukkan halaman *dashboard* pengguna setelah masuk ke akunnya melalui fitur *log in*.



Gambar 4.10 cuplikan layar halaman log in



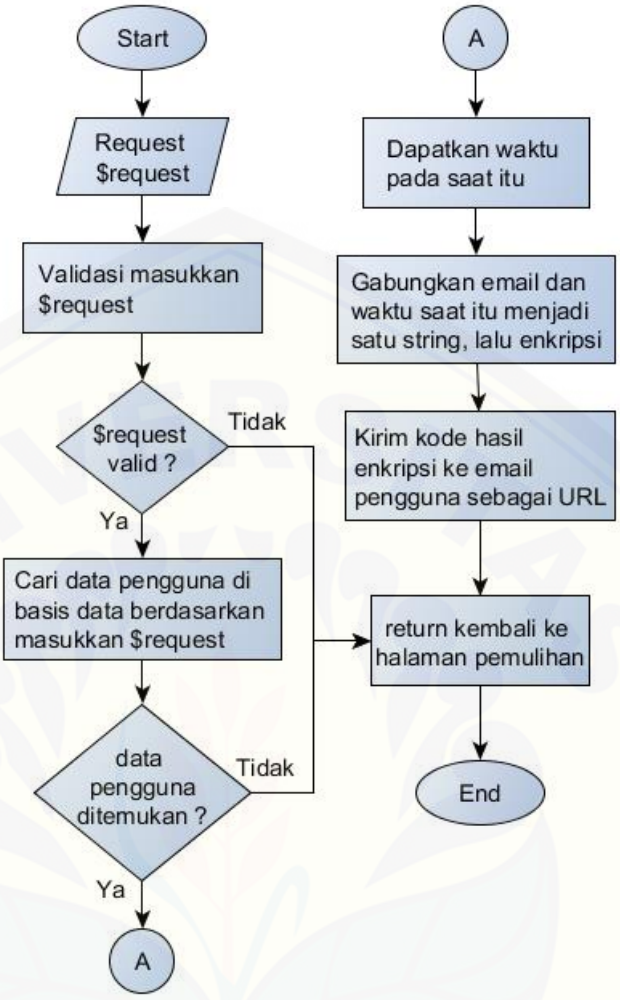
Gambar 4.11 cuplikan layar halaman dashboard setelah log in

4.3 Pemulihan Citra

Fitur pemulihan kata sandi digunakan ketika citra milik pengguna hilang atau mengalami kerusakan sehingga tidak dapat digunakan masuk ke akun pengguna. Proses pada fitur pemulihan citra terdiri dari beberapa tahap, mulai dari pencocokan email dan tanggal lahir, hingga penyisipan kredensial baru pengguna ke citra baru. Fungsi *kirim_email_pemulihan* merupakan fungsi pertama pada tahapan proses pemulihan ini. Gambar 4.12 di bawah menunjukkan *flowchart* fungsi tersebut.

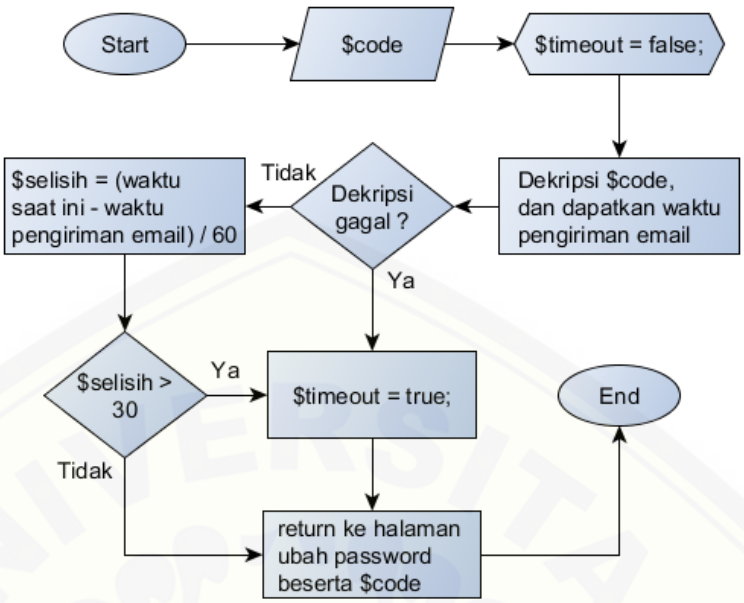
Langkah pertama pada fungsi *kirim_email_pemulihan* adalah memvalidasi email dan tanggal lahir yang dimasukkan pengguna. Validasi dilakukan dengan memanfaatkan fungsi bawaan *validate* dari Laravel. Jika email dan tanggal lahir tidak sesuai dengan standar validasi, maka sistem akan mengembalikan ke tampilan pertama pemulihan citra. Jika data telah sesuai, maka akan dilanjutkan ke tahap berikutnya.

Selanjutnya, data pengguna dicari berdasarkan email dan tanggal lahir pada basis data. Jika data pengguna tidak ditemukan, maka sistem akan mengembalikan ke halaman pemulihan citra dengan pesan eror. Jika data ditemukan, maka proses akan dilanjutkan ke tahap berikutnya.



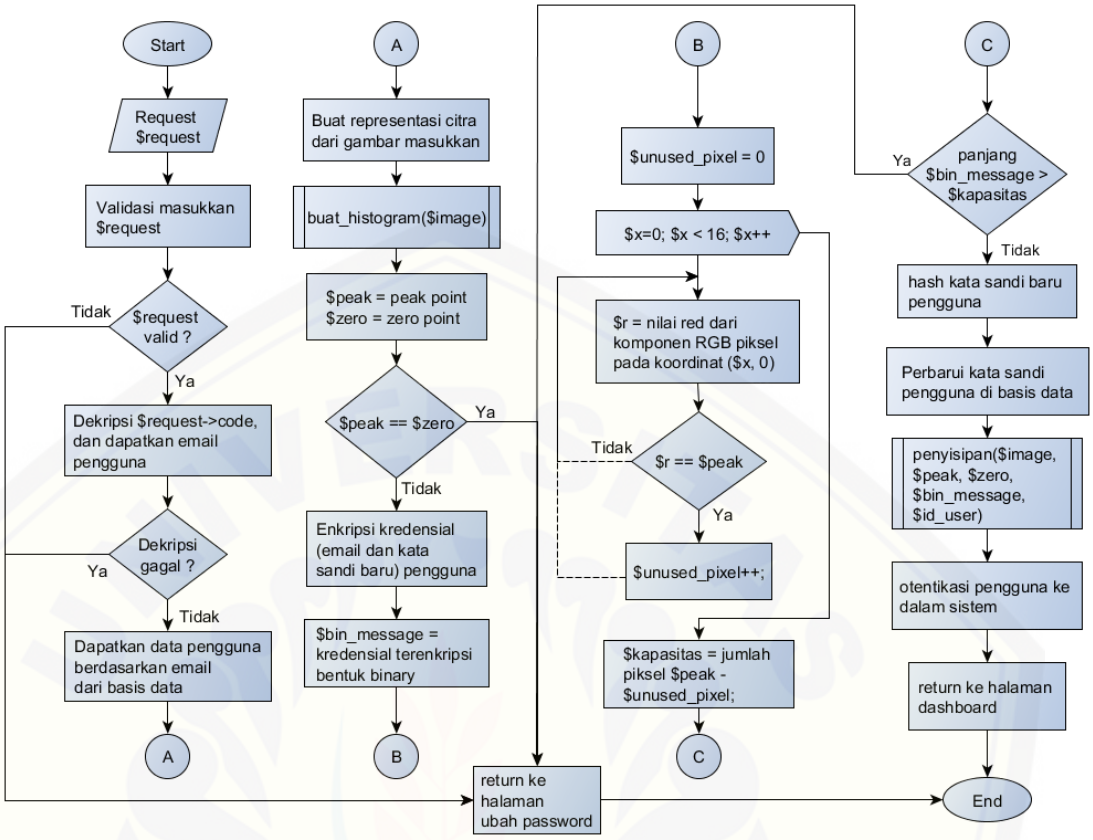
Gambar 4.12 *flowchart* fungsi *kirim_email_pemulihan*

Setelah data pengguna diperoleh, dapatkan waktu saaat ini dalam bentuk *timestamp*. Lalu, gabungkan waktu saat ini dengan email pengguna menjadi satu *string.* Nilai waktu tersebut akan digunakan nantinya untuk menghitung masa berlaku URL pemulihan. Lakukan enkripsi pada *string* gabungan waktu dan email. Kirim email berisi kode hasil enkripsi tersebut dalam bentuk URL ke alamat email pengguna. Arahkan kembali ke halaman pemulihan citra dengan pesan sukses. Pengguna diharuskan membuka URL yang telah dikirmkan untuk melanjutkan proses pemulihan citra.



Gambar 4.13 *flowchart* fungsi *reset_cover*

Ketika pengguna membuka URL pemulihan yang dikirim melalui email, sistem akan memeriksa terlebih dahulu apakah URL tersebut masih berlaku. Tahap ini dilakukan pada fungsi *reset_cover* seperti ditunjukkan oleh *flowchart* pada Gambar 4.13 di atas. Variabel $code berisikan kode terenkripsi yang disertakan pada URL pemulihan. Pertama, variabel $timeout diberi nilai *false*. Lalu dilakukan dekripsi pada $code menggunakan fungsi Laravel *decrypt*. Jika kode tidak dapat didekripsi, nilai variabel $timeout diubah menjadi *true*, lalu diarahkan ke halaman ubah *password* dengan pesan eror. Jika dekripsi berhasil, hitung selisih waktu dengan mengurangi waktu saat ini dengan waktu pengiriman email pemulihan, lalu bagi dengan 60. Batas waktu URL pemulihan adalah 30 menit, jika lebih dari itu maka ubah nilai variabel $timeout menjadi *true*. Hal ini akan menyebabkan pengguna tidak bisa memulihkan data sandi pada tampilan berikutnya, dan diharuskan membuat email pemulihan baru. Jika selisih waktu masih di bawah 30 menit, maka di halaman berikutnya pengguna dapat memasukkan kata sandi dan citra baru untuk mendapatkan citra stego baru. Variabel $code juga dikutkan ke halaman ubah password.



Gambar 4.14 *flowchart* fungsi *update_cover*

Setelah pengguna memasukkan citra dan kata sandi baru pada halaman ubah *password*, sistem akan menjalankan fungsi *update_cover* untuk memperbarui kata sandi dan citra stego pengguna. Gambar 4.14 munjukkan *flowchart* fungsi *update_cover*. Pertama dilakukan validasi terhadap citra dan kata sandi yang dimasukkan. Jika citra atau kata sandi tidak sesuai standar validasi, maka sistem akan mengembalikan ke halaman ubah *password* beserta pesan eror. Selain itu, kode terenkripsi yang dikirmkan melalui email juga diikutsertakan higga tahap ini. Jika kode ini tidak ditemukan pada tahap ini, maka kemungkinan terjadi kesalahan, dan pengguna diharuskan membuat permintaan pemulihan baru. Kode yang disertakan selanjutnya didekripsi untuk mendapatkan alamat email pengguna. Jika proses dekripsi gagal, maka sistem akan mengarahkan kembali

ke halaman ubah *password* dengan pesan eror. Email yang didapat selanjutnya digunakan untuk mengambil data pengguna dari basis data.

Citra yang diunggah pengguna selanjutnya dibuatkan representasinya agar citra dapat diolah atau dimanipulasi lebih lanjut. Representasi citra selanjutnya digunakan untuk membuat histogram menggunakan fungsi *buat_histogram* seperti pada Gambar 4.3. Selanjutnya tentukan nilai *peak* dan *zero point* dari histogram yang telah dibuat. Kedua nilai ini akan digunakan ketika menyisipkan pesan rahasia ke citra penampung nantinya. Kedua nilai tersebut terdapat pada indeks *array* histogram. *Peak point* ditunjukkan oleh indeks *array* histogram dengan nilai elemen tertinggi, dan *zero point* ditunjukkan oleh indeks *array* histogram dengan nilai elemen terendah. Pengecekan akan dilakukan untuk memastikan nilai *peak* dan *zero point* tidak sama, karena penyisipan tidak dapat dilakukan jika kedua nilai sama. Sistem akan mengembalikan ke halaman ubah *password* jika *peak* dan *zero point* bernilai sama.

Berikutnya, kredensial pengguna yang terdiri dari email dan kata sandi baru dienkripsi terlebih dahulu. Kredensial ini berperan sebagai pesan rahasia yang akan disisipkan. Proses enkripsi menggunakan fungsi bawaan Laravel *encrypt.* Enkripsi dilakukan agar kredensial pengguna lebih aman jika suatu saat orang lain mencoba mengekstraknya. Pesan rahasia terenkripsi kemudian diubah ke bentuk binary untuk kebutuhan penyisipan.

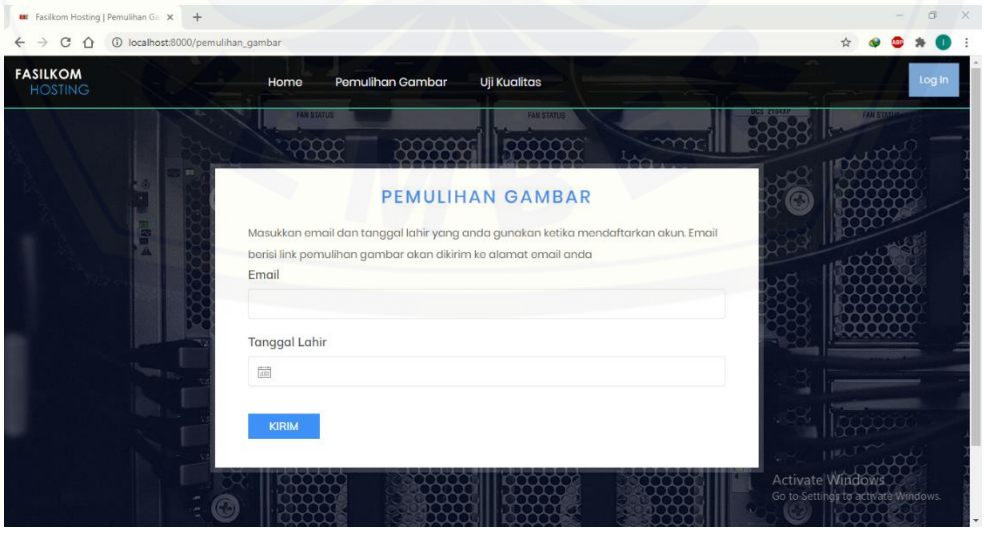
Sebelum penyisipan pesan rahasia, citra akan diperiksa dahulu apakah memiliki kapasitas yang cukup. 16 piksel pertama pada citra akan digunakan untuk menyisipkan nilai *peak* dan *zero point*, sehingga jika pada 16 piksel pertama tersebut terdapat piksel dengan nilai komponen warna merah sama dengan *peak point*, maka piksel tersebut tidak dapat digunakan untuk menyisipkan pesan rahasia. pertama, pindai 16 piksel pertama secara horizontal dan dapatkan nilai komponen warna merah untuk setiap piksel tersebut. Jika nilai warna merah sama dengan *peak point*, maka tambahkan nilai 1 pada variabel $unused_pixel. Variabel ini akan menyimpan banyaknya piksel *peak point* yang tidak dapat diguankan untuk menyisipkan pesan rahasia. Lalu, kurangkan jumlah piksel *peak point* dengan nilai variabel $unused_ pixel. Variabel $kapasitas

berisi jumlah piksel yang dapat digunakan untuk menyisipkan pesan rahasia. Dilakukan pengecekan apakah panjang bit pesan rahasia lebih besar dari pada nilai variabel $kapasitas. Jika bit pesan rahasia lebih panjang, maka citra tidak dapat digunakan dan sistem akan mengembalikan ke halaman ubah *password*.

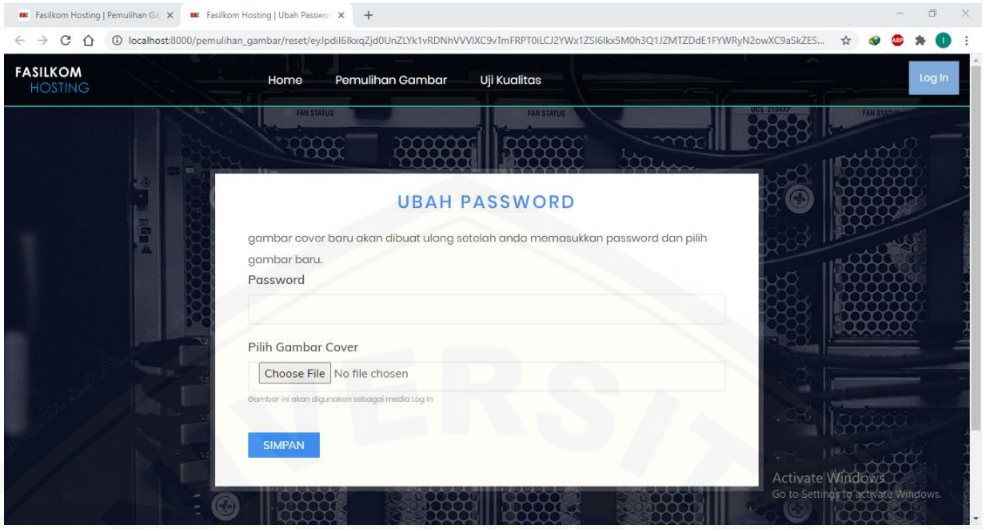
Selanjutnya akan dilakukan *hash* pada kata sandi baru pengguna. Proses *hash* memanfaatkan fungsi bawaan *hash* dari Laravel. Fungsi tersebut menggunakan metode *hash* bcrypt. Setelah itu, data pengguna akan diperbarui menggunakan kata sandi baru yang telah di*hash* tersebut.

Langkah berikutnya, dilakukan penyisipan pesan rahasia menggunakan fungsi *penyisipan*. Fungsi tersebut membutuhkan parameter representasi citra, *peak point*, *zero point*, bit pesan rahasia, dan id pengguna. Gambar 4.4 menunjukkan *flowchart* fungsi *penyisipan*. Setelah itu, pengguna akan diotentikasi masuk ke sistem dan diarahkan ke halaman *dashboard*. Citra stego baru pengguna dapat diunduh di halaman *dashbord* melalui tombol "Download". Ketika tombol tersebut diklik, fungsi *download_cover* (Gambar 4.5) akan dijalankan untuk mengunduh citra stego.

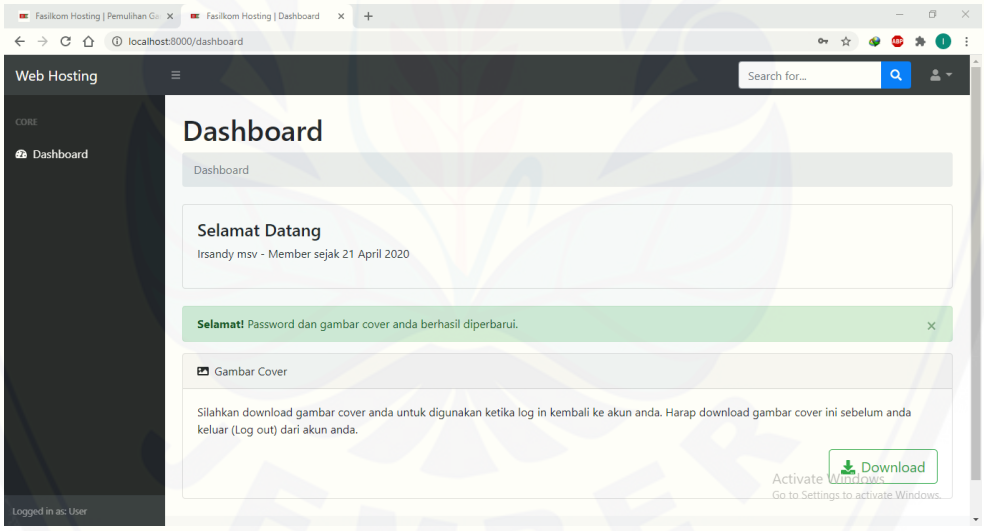
Berikut merupakan cuplikan layar tampilan halaman pemulihan citra (Gambar 4.15), halaman ubah *password* (Gamabr 4.16), dan halaman *dashboard* pengguna setelah memperbarui citra (Gambar 4.17).



Gambar 4.15 cuplikan layar halaman pemulihan gambar (citra)



Gambar 4.16 cuplikan layar halaman ubah *password*



Gambar 4.17 cuplikan layar halaman *dashboard* setelah pemulihan citra

BAB 6. PENUTUP

Bab ini merupakan bagian akhir dari laporan skripsi yang terdiri dari kesimpulan dan saran. Kesimpulan ditulis berdasarkan tahapan penelitian yang telah dilaksanakan. Saran diberikan sebagai acuan untuk penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan tahapan penelitian yang telah dilaksanakan, berikut merupakan kesimpulan yang didapatkan pada penelitian ini

1. Mekanisme alternatif otentikasi dilakukan dengan menyisipkan kredensial pengguna ke dalam citra pembawa, lalu citra stego yang telah disisipi kredensial tersebut dapat digunakan untuk masuk ke akun pengguna. Sebelum disisipkan, kredensial yang terdiri dari email dan kata sandi dienkripsi terlebih dahulu. Proses enkripsi ini dilakukan untuk menjaga keamanan kredensial dan akun pengguna. Penyisipan kredensial dilakukan pada komponen warna merah (*Red*) suatu piksel. Selain kredensial, nilai *peak* dan *zero point* juga disisipkan pada 16 piksel horizontal pertama. Penyisipan dilakukan pada fitur registrasi ketika membuat akun baru. Ketika digunakan untuk masuk ke akun pengguna melalui fitur *log in*, kredensial pada Citra stego akan diekstrak terlebih dahulu. Kredensial yang telah diekstrak kemudian didekripsi untuk mendapatkan kredensial asli pengguna. Selanjtunya, dilakukan proses otentikasi pengguna menggunakan kredensial asli tersebut. Proses penyisipan dan ekstraksi kredensial dilakukan menggunakan meode *Hisotgram Shifting*. Dibandingkan metode otentikasi menggunakan kata sandi berbasis teks, penggunaan citra sebagai media otentikasi dapat mempermudah seorang pengguna karena tidak harus mengingat kata sandi yang dibuat ketika melakukan registrasi.
2. Fitur pemulihan citra digunakan untuk memperbarui kata sandi dan citra stego pengguna yang rusak atau hilang. Pengguna diharuskan memasukkan email dan tanggal lahir yang digunakan ketika melakukan registrasi. Setelah

itu, akan dilakukan proses pengecekan data pengguna di basis data berdasarkan data masukkan tadi. Jika data pengguna ditemukan, email berisi URL pemulihan citra akan dikirim ke akun pengguna. URL tersebut memiliki batas waktu 30 menit untuk digunakan. Ketika pengguna membuka URL yang dikirmkan, akan dilakukan proses pengecekan masa berlaku URL. Jika masa berlaku URL telah habis, maka pengguna diharuskan membuat ulang permintaan pemulihan citra. Namun jika masa berlaku URL masih ada, pengguna selanjutnya diarahkan ke halaman ubah kata sandi. Di halaman tersebut, pengguna diharuskan memasukkan kata sandi dan citra baru. Tahap selanjutnya dilakukan penyisipan kredensial baru ke dalam citra baru sama seperti pada proses registrasi. Sebelum penyisipan dilakukan, data pengguna akan diperbarui dengan kata sandi baru. Setelah proses penyisipan selesai, pengguna diotentikasi masuk ke akun miliknya dan dapat mengunduh citra stego baru untuk digunakan ketika akan masuk kembali ke akunnya.

3. Pengujian terhadap citra hasil steganografi (citra stego) dilakukan berdasarkan tiga kriteria, yaitu kualitas citra, kemampuan pengungkapan (*recovery*) pesan rahasi, dan ketahanan (*robustness*) pesan rahasia. Kualitas citra diukur mengguankan metode PSNR. Melalui pengujian terhadap 10 citra stego, didapatkan nilai rata-rata PSNR sebesar 52,52 dB. Nilai ini lebih tinggi dari standar nilai 36 dB yang digunakan pada penelitian ini, dan dapat dikatakan bahwa metode *Histogram Shifting* yang diterapkan mampu menghasilkan citra dengan kualitas tinggi yang mirip dengan aslinya. Pengujian kemampuan pengungkapan atau ekstraksi menunjukkan bahwa kesepuluh citra stego dari pengujian sebelumnya dapat digunakan untuk masuk ke akun pengguna. Sehingga dapat dikatakan bahwa metode *Histogram Shifting* yang diterapkan mampu mengekstrak pesan rahasia dengan tepat. Pengujian ketahanan pesan rahasia menunjukkan citra yang telah dimanipulasi tidak dapat digunakan untuk masuk ke akun pengguna. Dapat dikatakan bahwa pesan rahasia tidak tahan terhadap semua teknik manipulasi yang diterapkan pada citra pembawa. Hal tersebut dikarenakan

kebanyakan teknik manipulasi yang diterapkan mempengaruhi jumlah atau tingkat kecerahan piksel-piksel suatu citra. Berubahnya jumlah atau tingkat kecerahan piksel-piksel menyebabkan pesan rahasia mengalami kerusakan.

4. Proses registrasi dan *log in* membutuhkan waktu eksekusi di bawah setengah detik. Hasil pengukuran menunjukkan waktu eksekusi rata-rata proses registrasi adalah 0,265 detik. Sementara waktu eksekusi rata-rata proses *log in* adalah 0,119 detik. Pengujian dilakukan sebanyak 10 kali untuk masing-masing proses.

6.2 Saran

Penerapan metode steganografi *Histogram Shifting* sebagai alternatif otentikasi masih mempunyai kekurangan. Penulis menyarankan pengembagan penelitian lebih lanjut guna memperbaiki atau menyempurnakan kekurangan tersebut. Berikut merupakan saran untuk penelitian selanjutnya

1. Citra stego pada penelitian ini masih rentan terhadap manipulasi citra yang dapat merusak pesan rahasia di dalamnya. Penelitian berikutnya dapat menerapkan metode steganografi yang mampu menghasilkan citra dengan ketahanan (*robustness*) lebih baik terhadap berbagai teknik manipulasi.
2. Pengujian kecepatan waktu eksekusi dalam penelitian ini hanya dilakukan menggunakan satu citra. Penelitian selanjutnya dapat melakukan pengujian terhadap banyak citra dengan reoslusi yang beragam guna mengetahui secara nyata pengaruh resolusi citra terhadap waktu eksekusi.

DAFTAR PUSTAKA

Al-Najjar, Yusra A. Y., dan Dr. Der Chen Soong. 2012. "Comparison of Image Quality Assessment: PSNR, HVS, SSIM, UIQI." *International Journal of Scientific & Engineering Research* 3 (8).

Andono, Pulung Nurtantio, T. Sutojo, dan Muljono. 2017. *Pengolahan Citra Digital*. Yogyakarta: Penerbit ANDI.

Banarjee, Indradip, Souvik Bhattacharyya, dan Gautam Sanyal. 2013. "Hiding & Analyzing Data in Image Using Extended PMM." *International Conference on Computational Intelegence: Modeling Techniques and Applications (CIMTA)*. Elsevier Ltd. 157-166.

Bonneau, J, C Herley, P.C van Oorschot, dan F Stajano. 2012. "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes." *IEEE Symphonium on Security and Privacy*. 553-567.

Cox, Ingemar J., Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. 2008. *Digital Watermarking and Steganography 2nd edition*. Morgan Kaufmann.

Florencio, Dinei, dan Cormac Herley. 2007. "A Large-Scale Study od Web Password Habits." *Proceeding of the 16th International Conference on World Wide Web*. 657-666.

Fridrich, Jessica. 2010. *Steganography in Digital Media - Principal, Algorith, and Applications*. Cambridge University Press.

Idrus, Syed Zulkarnain Syed, Estelle Cherrier, Christophe Rosenberger, dan Jean-Jacques Schwartzmann. 2013. "A Review on Authentication Method." *Australian Journal of Basic and Applied Sciences* 95-107.

Mare, Shrirang, Mary Baker, dan Jeremy Gummeson. 2016. "A Study of Authentication in Daily Life." *Twelfth Symposium on Usable Privacy and Security (SOUPS)*. Colorado. 189-206.

Musianto, Lukas S. 2002. "Perbedaan Pendekatan Kuantitatif dengan Pendekatan Kualitatif dalam Metode Penelitian." *Jurnal Manajemen & Kewirausahaan* 4: 123-136.

Ni, Zhicheng, Yun-Qing Shi, Nirwan Ansari, dan Wei Su. 2006. "Reversible Data Hiding." *IEEE Transactions on Circuits and System for Video Technology* 16: 354-362.

Nurdiansyah, Yanuar, dan Ayu Lusia Fitrasari Riftana. 2017. "Implementasi Steganografi Citra Digital Pemberkasan Arsip Menggunakan Metode Least Significant Bit." *Seminar Nasional Informatika dan Aplikasinya (SNIA).* Cimahi: Universitas Jenderal Achmad Yani. 2-7.

Suo, Xiaoyuan, Ying Zhu, dan G Scott Owen. 2005. "Graphical Password: A Survey." *Computer security applications conference, 21st annual.* IEEE.

Wash, Rick, Emilee Rader, Ruthie Berman, dan Zac Wellmer. 2016. "Understanding Password Choices: How Frequently Entered Password are Re-used Across Website." *Twelfth Symposium on Usable Privacy and Security (SOUPS).* Colorado. 175-188.