KRIPTOGRAFI VIGENERE UNTUK MENGAMANKAN PESAN TEKS BERBASIS OCR (OPTICAL CHARACTER RECOGNITION)

Elkaf Rahmawan Pramudya[1], Lekso Budi Handoko[2], Muslih[3]
[1,2]Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro e-mail:
[1]elkaf.rahmawan@dsn.dinus.ac.id, [2]handoko@dsn.dinus.ac.id, [3]muslih@dsn.dinus.ac.id

ABSTRAK

Keamanan teknologi adalah isu terbesar yang sedang marak di khalayak ramai sampai saat ini, berbagai perkembangan teknologi memunculkan setiap isu keamanan, mulai dari keamanan data pribadi sampai keamanan teks pada citra yang tersimpan pada gawai setiap individu. Hal ini membuat banyak orang semakin risau dengan privasi yang dimilikinya, apalagi privasi pada pesan yang tersimpan dalam bentuk citra, karena data privasi yang dimilikinya rentan akan peretasan yang akan terjadi sewaktu-waktu. Dengan pesatnya perkembangan teknologi terutama pada bidang keamanan memunculkan inovasi terbaru salah satunya adalah mengimplementasikan penggunaan kriptografi dengan metode vigenere dengan Optical Character Recognition atau OCR. Maka dari itu, diperlukannya sebuah penelitian untuk melihat seberapa jauh dan efektifkah penggunaan kriptografi vigenere dengan Optical Character Recognition yang menggunakan algoritma Template Matching dengan pengujian Avalanche Effect dan Black Box Testing. Data yang digunakan merupakan data citra teks yang masing-masing memiliki format atau ekstensi yang berbeda-beda yaitu .jpg/jpeg, .png dan .gif. Data citra teks akan diproses dan diolah terlebih dahulu menggunakan algoritma Template Matching, ketika hasil dari algoritma Template Matching keluar maka hasilnya dapat di enkripsi dan di deskripsi secara langsung dengan menggunakan metode vigenere. Ketika hasil implementasi program dapat berjalan dengan baik, maka perlu adanya sebuah pengujian, pengujian dilakukan dengan menggunakan Avalanche Effect untuk mengukur seberapa kuatkah efek yang timbul dari penggunaan metode vigenere dan dilakukan juga pengujian dengan menggunakan Black Box Testing untuk melihat hasil implementasi program yang dibuat.

Kata Kunci*: Vigenere, Template Matching, Avalanche Effect, Black Box Testing, Optical Character Recognition.*

1. PENDAHULUAN

Keamanan merupakan isu krusial yang sering dihadapi oleh kebanyakan orang termasuk pada penggunaan teknologi, seiring berkembangnya teknologi maka isu-isu keamanan juga akan meningkat terutama pada keamanan data yang sering muncul dan sering didengungkan oleh kebanyakan orang tentang keamanan data privasi mereka. Isu keamanan ini sering terjadi pada instansi perkantoran, pemerintahan, swasta, pertahanan negara maupun pada perbankan dan tidak luput juga pada keamanan pribadi terutama pada data yang ada pada gawai yang kita gunakan. Gawai atau s*martphone* yang kita gunakan juga memiliki kerentanan terhadap keamanan apalagi keamanan terhadap data citra teks yang kita simpan pada galeri. Penggunaan teknologi pada masa informasi seperti ini, keamanan data pribadi sangat dibutuhkan dan menjadi persoalan yang sangat penting pula, data yang disimpan tidak akan dapat menjadi rahasia lagi apabila di tengah berjalannya teknologi informasi banyak penyadapan dan penyalahgunaan data yang tidak bertanggung jawab, apalagi penyalahgunaan data citra teks yang seringkali di sunting tanpa sepengetahuan pemiliknya [1]–[3]. Data citra teks ini lebih sering disalahgunakan dan juga di perjual belikan dengan illegal. Dengan cara penyuntingan pada teks asli dan tanpa pemberitahuan kepada pemilik sebelumnya, maka hal itu akan sangat merugikan bagi pemilik data citra teks tersebut. Untuk mencegah agar terjadinya penyadapan dan penyalahgunaan terhadap data citra teks, maka diperlukan sebuah solusi dalam penanganannya, penanganannya yaitu dengan memanfaatkan teknologi dan algoritma yang sudah ada, termasuk pemanfaatan teknologi OCR *(Optical Character Recognition)* [4]–[6] dengan metode *Template Matching* dan Kriptografi *Vigenere.*

OCR atau kepanjangan dari *(Optical Character Recognition)* merupakan istilah dari pengenalan pola huruf atau abjad dan merupakan bagian dari bidang ilmu *image processing*. *Image Processing* sendiri merupakan bidang ilmu yang mengolah citra atau gambar dalam bentuk digital dengan cara di *filter.* Dengan cara *filter* atau penyaringan dapat meningkatkan kualitas pada citra [7]. Biasanya ada banyak jenis dari *filter* yang ada pada *image processing,* diantaranya adalah Penyaringan Minimum *(Minimum Filtering),* Penyaringan Maksimum *(Maximum Filtering),* Penyaringan Median *(Median Filtering)* dan Penyaringan Rata-rata *(Average Filtering), d*an sebagian penyaringan atau *filtering* tersebut, memiliki algoritmanya masing-masing dalam pengolahan citra.

Dalam pembahasan diatas terdapat sebuah data yang dinamakan citra, sedangkan citra sendiri, memiliki arti yang sangat luas. Citra merupakan sebuah istilah lain untuk mendefinisikan gambar dalam bentuk digital atau sebagai

salah satu komponen yang termasuk kedalam multimedia [8]. Citra yang sudah diolah menjadi bentuk digital secara koordinat areanya maupun secara *brightness* levelnya *f(x,y)*. Citra yang sudah menjadi bentuk digital di analogikan sebagai matriks yang mempunyai ukuran N x M atau dimana N merupakan baris dari matriks dan M merupakan kolom dari matriks. Citra yang dapat dilihat oleh mata merupakan masukan atau input untuk otak dan fungsi sebenarnya dari OCR yang mengolah citra sama seperti pada mata yang melihat citra yang diproses pada otak [9]–[11].

OCR atau *(Optical Character Recognition)* tidak akan dapat berjalan dan memproses citra tanpa adanya teori yang menjelaskannya. OCR sendiri berjalan pada metode *template matching*, yang mana metode ini dapat melakukan pencocokan pada citra inputan. Definisi dari *template matching* sangat beragam, sama halnya dengan citra, *Template Matching* merupakan metode yang mencocokan pada setiap bagian dari citra dengan citra yang menjadi acuan atau *template*-nya, *template matching* juga menjadi salah satu teknik dalam pengolahan citra digital yang sering digunakan dalam riset [12]. Untuk menghitung *template matching*, diperlukan sebuah rumus dan rumus tersebut saya kutip dari

jurnal milik Made Sulastri Dewi yaitu $\Sigma Min\ e = (Ix, y - Tx, y)^2$. Rumus tersebut mencari nilai kesesuaian tingkat citra masukan dan citra acuan *(Template)* dengan menghitung berdasar nilai *error* terkecil yang akan muncul. Dari nilai *error* terkecil yang akan muncul, maka dapat dipastikan bahwa komputer dapat mengenali huruf yang ada pada citra digital dengan baik.

Proses dari pencocokan citra yang dilakukan oleh OCR dengan menggunakan metode *template matching* telah berhasil, output yang dihasilkan adalah bentuk dari digital yang nantinya dapat diproses kembali. Hasil dari OCR nanti akan di enkripsi, menurut Akim Manaor Hara Pardede enkripsi sangat diperlukan karena enkripsi sendiri diartikan sebagai perubahan bentuk asli teks ke dalam kode. Untuk mengenkripsi hasil OCR, digunakanlah metode kriptografi *Vigenere*, kriptografi *Vigenere* sendiri merupakan hasil dari penyederhanaan sandi substitusi polialfabetik dan kriptografi *Vigenere* terdiri dari beberapa bagian sandi *Caesar* dengan proses pergeseran nilai yang berbeda. Dalam proses enkripsi dan dekripsi pada *vigenere*, Arif Amrullah menjelaskan bahwa penyandian *vigenere* bekerja dengan cara membaca kata per karakter, yang mana suatu pesan dikirim melebihi kunci yang sudah ditentukan maka kunci akan diputar kembali sampai pesan yang dikirim menemukan kuncinya masing-masing. Secara matematis untuk menghitung enkripsi dan dekripsi adalah enkripsi $C_i = (P_i + K_i)\ mod\ 26$ sedangkan deskripsinya sendiri $C_i = (P_i - K_i)\ mod\ 26$. Dengan menggunakan rumus enkripsi dan deskripsi tadi, kita dapat menyembunyikan hasil OCR dengan mudah, tentunya dengan mengikuti kaidah yang sudah ada. Dari hasil yang sudah dipaparkan, diharapkan dengan menggunakan OCR atau *(Optical Character Recognition)* dengan menggunakan *template matching* sebagai metodenya dan kriptografi *vigenere* dapat mempermudah untuk menyembunyikan data berupa citra teks yang rentan dalam pembajakan, pemalsuan ataupun kejahatan-kejahatan teknologi lainnya.

2. METODE PENELITIAN

Penelitian kali ini dilakukan dengan menggunakan tiga data citra teks dengan masing-masing ekstensi atau format yang berbeda, tiga ekstensi diantaranya adalah JPG/JPEG, PNG, dan GIF. Proses perhitungan dan ekstraksi citra teks akan diproses dengan menggunakan OCTAVE dari metode perhitungan *template matching* dan hasil dari proses tersebut akan dapat di enkripsi maupun di deskripsi oleh penyandian *Vigenere*.

Template Matching

Template Matching merupakan salah satu metode yang merupakan metode pencocokan pada setiap bagian dari citra dengan citra yang menjadi acuan atau *template*-nya, *template matching* juga menjadi salah satu teknik dalam pengolahan citra digital yang sering digunakan dalam riset. Citra sebagai modal awal dalam *image processing* untuk melakukan pencocokan citra satu ke citra lainnya [13]. Pada citra sendiri, tanpa adanya metode pencocokan yang menggunakan *Template Matching* komputer tidak akan dapat mengenali apa itu citra dan bagaimana citra itu dapat dianalisa oleh sistem komputer. Dalam proses pengenalan citra sendiri, *template matching* dapat di persamaan (1).

$$\Sigma Min\ e = (Ix, y - Tx, y)^2 \quad (1)$$

Dimana *Min e* adalah error Minimum yang paling kecil, *I adalah* pola piksel masukan yang akan dibandingkan sedangkan T adalah pola piksel dari citra template.

Penyandian Vigenere

Keamanan pada data sangat dibutuhkan apalagi pada data citra teks bertulis, dalam penggunaannya dibutuhkan metode agar informasi pesan tersebut dapat disembunyikan dengan aman. Penggunaan Penyandian *Vigenere* adalah salah satu kuncinya, penyandian ini merupakan hasil dari salah satu penyederhanaan sandi substitusi

polialfabetik [14]. Untuk menyembunyikan dan membaca atau dengan kata lain mengenkripsi data dan mendeskripsikan data citra tersebut maka dapat digunakan rumus *vigenere* sesuai persamaan (2) dan persamaan (3).

Enkripsi:

$$C_i = (P_i + K_i) \bmod 26 \quad (2)$$

Deskripsi:

$$C_i = (P_i - K_i) \bmod 26 \quad (3)$$

Keterangan:

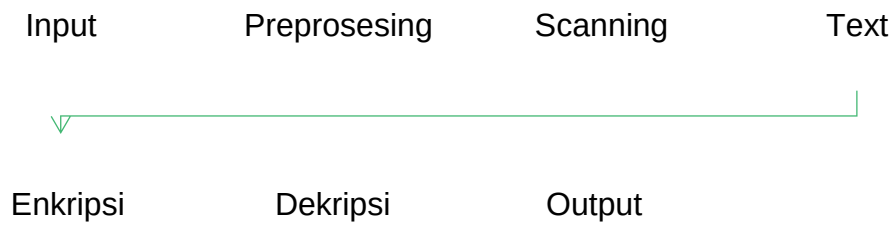
C_i : Nilai Desimal Karakter Chipertext Ke-i
P_i : Nilai Desimal Karakter Plaintext Ke-i
K_i : Nilai Desimal Karakter Key Ke-i
i : Nilai iterasi untuk melakukan perulangan
Mod : Sisa Hasil Bagi

Avalanche Effect

Avalanche Effect merupakan salah satu metode pengujian untuk menguji dan menentukan baik atau tidaknya algoritma kriptografi yang akan digunakan. Dalam pengujian *avalanche effect*, perubahan yang dilakukan pada *plaintext* maupun *key* akan berdampak pada perubahan *ciphertext bit value* yang sangat signifikan. Dengan kata lain, perubahan satu atau dua bit pada *plaintext* dan *key* akan sangat berpengaruh pada perubahan *bit chipertext* yang lainnya.

Usulan Metode

Dalam penelitian ini secara garis besar, metode yang diusulkan adalah data berupa citra bertulis yang akan diproses dalam pembentukan enkripsi dengan menggunakan OCR (Optical Character Recognition) dan Kriptografi Vigenere. Dalam proses enkripsi, maka program akan membagi alur kerja menjadi dua yaitu Alur yang pertama adalah scanning teks terlebih dahulu dan Alur yang kedua adalah proses enkripsi dari hasil scanning yang sudah dilakukan sesuai Gambar 1.



Gambar 1. Skema Usulan Metode

Berdasarkan Gmabar 1, proses awal adalah melakukan input data berupa image. Data image ini nantinya akan di proses pada bagian preprocessing setelah input dilakukan. Pada proses preprocessing, program akan melakukan penerjemahan dengan cara menghilangkan noise atau titik-titik pengganggu pada pola yang tidak berguna. Ketika noise hilang, maka proses yang akan berjalan bisa lebih lancar. Memasuki proses scanning, setiap huruf akan dilakukan pengenalan pola setelah penghilangan noise pada tahap preprocessing. Pengenalan ini dilakukan guna huruf yang akan muncul tidak akan terjadi kesalahan maupun kecacatan bentuk. Text ini adalah hasil dari tahap scanning, apabila tahap scanning berjalan dengan lancar, maka teks yang akan muncul akan sesuai harapan. Memasuki tahap ini, text akan langsung di enkripsi dengan metode kriptografi vigenere. Hasil akan keluar ketika semua proses sudah berjalan dan enkripsi teks yang ada pada image akan muncul secara otomatis.

3. HASIL DAN PEMBAHASAN

Merubah Citra RGB ke Biner

Hasil percobaan bila dimungkinkan ditampilkan dalam bentuk gambar/tabel seperti contoh pada gambar 2 berikut. Dalam proses perhitungan menggunakan metode *template matching*, hal yang pertama dilakukan adalah

dengan merubah citra terlebih dahulu, dari RGB ke Biner. Fungsi perubahan citra RGB ke biner adalah untuk pembentukan pola dan pengenalan huruf dengan menggunakan biner yang sudah terbentuk seperti pada Tabel 1.

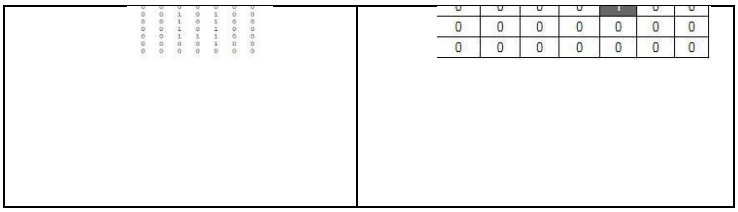
Tabel 1. Merubah Citra RGB Ke Citra Biner

Huruf	RGB ke biner	Huruf	RGB ke biner

Berdasarkan Tabel 1, proses perubahan dari RGB ke Biner dapat dilakukan dengan membuat nama variabel terlebih dahulu. Disini penulis menggunakan nama variabel F1 sebagai object analisa hurufnya, ketika nama variabel sudah ditentukan maka penulis akan memanggil nama variabel F1 dengan fungsi *"imread"* dan diteruskan pada folder mana yang kita simpan. Ketika pemanggilan object huruf sudah dilakukan, maka proses perubahan dari RGB ke Biner dapat dilakukan dengan menggunakan fungsi *"rgb2gray(f1)"*, maka dengan itu hasil akan keluar dengan nilai angka RGB. Nilai hasil penggunaan fungsi *"rgb2gray(f1)"* akan membentuk nilai dari background yang digunakan pada gambar citra dan membentuk juga nilai warna pada teks yang digunakan. Nantinya nilai warna tersebut akan di binerisasi untuk membentuk pola teks seperti ditunjukkan Tabel 2.

Tabel 2. RGB ke biner dan tampilan biner huruf

RGB ke biner	Pola
	0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	0 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0



Perubahan nilai RGB ke Biner sudah dilakukan, maka proses selanjutnya adalah melihat hasil pola dari nilai biner setiap huruf. Pola-pola tersebut akan penulis bentuk secara manual dengan menggunakan *spreadsheet* seperti pada tabel diatas dan penulis sesuaikan pada hasil perubahan biner *thresholding* untuk melihat dari pola yang terbentuk.

Proses Pengenalan Huruf

Untuk menghitung apakah nilai biner yang terbentuk sudah sesuai pola yang ada pada setiap huruf, maka penulis menggunakan persamaan (4).

$$Min\ e = \sum (Ix, y - Tx, y)^2 \quad (4)$$

Pada Tabel 3, penulis memiliki empat huruf yang masing-masing sudah memiliki nilai biner tersendiri. Empat huruf ini nantinya akan dihitung dengan satu huruf sebagai inputan yang diambil untuk membandingkan nilai terkecil dari setiap huruf, karena semakin kecil hasil dari nilai biner tersebut maka akan semakin mudah untuk OCR mengenali, membentuk dan mengekstrak setiap huruf yang sudah di proses.

Penyandian Vigenere

Vigenere Cipher termasuk dalam penyandian cipher substitusi polyalphabetic yang dikarang dan dirilis oleh seorang diplomat sekaligus kriptologis dari Prancis yang bernama Blaise de Vigenere pada abad ke-16. Vigenere cipher adalah metode penyandian teks yang berdasarkan deretan-deretan dan huruf-huruf dari metode penyandian caesar cipher yang terletak pada kata kunci. Letak dari kunci yang ada pada vigenere cipher merupakan pengembangan dari dari penyandian caesar cipher [15], [16], yang dimana setiap huruf asli diubah dengan huruf lain yang memiliki urutan yang berbeda pada setiap abjadnya seperti pada persamaan (5) dan persamaan (6).

$$Ci = (Pi + Ki) \bmod 26 \quad (5)$$

Atau

$$Ci = (Pi + Ki) - 26 \text{ “Jika hasil penjumlahan Pi dan Ki lebih dari 26.”} \quad (6)$$

Dari rumus diatas, dapat diketahui bahwasannya yang dicari adalah chipertextnya atau kata yang akan disandikan. Disini, penulis menggunakan kata *“UDINUS”* sebagai plaintext yang akan di enkripsi menjadi chiper. Selanjutnya yaitu dengan penambahan dari kata *“SIAP”* sebagai kunci. Untuk memulai menggunakan rumusnya, maka penulis akan menjelaskan pada Tabel 3 di bawah ini.

Tabel 3. Perhitungan Enkripsi Vigenere

Abjad	Hasil Angka Dari Abjad	Modulus	Hasil Modulus	Hasil Abjad Dari Modulus
U + S	20 + 18	38 mod 26	12	M
D + I	3 + 8	11 mod 26	11	L
I + A	8 + 0	8 mod 26	8	I
N + P	13 + 15	28 mod 26	2	C
U + S	20 + 18	38 mod 26	12	M
S + I	18 + 8	26 mod 26	0	A

Dimulai dari kolom abjad, terdapat potongan huruf dari plaintext yaitu kata *“UDINUS”* dan ditambah dengan potongan kata kunci yaitu *“SIAP”,* setiap abjadnya mempunyai nilai dari urutan abjad itu sendiri. Untuk huruf U+S mempunyai nilai 20+18 didapat dari urutan abjad A sampai Z setelah itu hasil dari penambahan huruf U+S yang mempunyai nilai 20+18 di modulo dengan 26. Nilai 38 mod 26 mempunyai hasil 12 dan angka 12 ada pada urutan abjad M. Begitu pula berlaku pada abjad yang lainnya dengan menambah abjad plaintext dengan kunci setelah itu hasilnya baru di modulus dan hasil dari modulus akan memperlihatkan urutan dari abjad, maka hasilnya sesuai urutan dari abjad tersebut. Maka Hasil dari perhitungan enkripsi dengan menggunakan rumus *Vigenere* yang ada pada tabel adalah MLICMA.

$$Ci = (Pi - Ki) \bmod 26 \quad (7)$$

Atau

$$Pi = (Ci - Ki) + 26 \text{ “Jika hasil perngurangan Pi dan Ki kurang dari } 26\text{”} \quad (8)$$

Dari persamaan (7) dan persamaan (8) yang penulis cantumkan di atas untuk dekripsi plaintext dari hasil ciphertext enkripsi, maka plaintext enkripsi akan dikurangi dengan kunci yang sama dan hasil pengurangan antara plaintext dan kunci akan di modulus 26 dan akan didapatkan chipertext untuk dekripsinya. Untuk lebih jelasnya penulis akan menjelaskan pada Tabel 4 di bawah ini.

Tabel 4. Perhitungan Dekripsi Vigenere

Abjad	Hasil Angka Dari Abjad	Modulus	Hasil Modulus	Hasil Abjad Dari Modulus
M - S	12 - 18	-6 mod 26	20	U
L - I	11 - 8	3 mod 26	3	D
I - A	8 - 0	8 mod 26	8	I
C - P	2 - 15	-13 mod 26	13	N
M - S	12 - 18	-6 mod 26	20	U
A - I	0 - 8	-8 mod 26	18	S

Sama halnya dengan penjelasan pada tabel enkripsi, pada kolom abjad terdapat plaintext *“MLICMA”* dari hasil enkripsi *“UDINUS”* dan terdapat abjad kata kunci *“SIAP”.* Masing-masing abjad mempunyai nilai urutan dan dari nilai urutan tersebut maka akan dikurangkan dengan nilai dari bajad kata kunci. Hasil pengurangan dari plaintext dengan kata kunci akan di modulus dengan 26 dan akan menghasilkan nilai urutan abjad A-Z. Misal untuk abjad M akan dikurangkan dengan S dan masing-masing abjad mempunyai nilai urutannya sendiri-sendiri, M mempunyai nilai urutan 12 dan S mempunyai nilai urutan 18 maka akan menghasilkan nilai -6 yang nantinya akan di modulus 26 dan akan menghasilkan nilai urutan abjad 20 yang terletak pada huruf U. Untuk hasil dari perhitungan *vigenere* dekripsi di atas akan kembali semula ke teks aslinya yaitu *“UDINUS”.*

Avalanche Effect

Hasil dari pengujian menggunakan *Avalanche Effect* menunjukan pada angka 43,75% yang mana penggunaan algoritma *Vigenere* cukup buat melindungi teks pada citra bertulis. Untuk lebih meyakinkan dalam penggunaan metode *vigenere,* penulis mencoba melakukan testing kedua kali terhadap metode *vigenere* dengan menggunakan *Avalanche Effect* yang diukur dengan jumlah seluruh bit pada *Message-Digest* atau MD2, MD4 dan MD5. Karena masing-masing *Message-Digest* memiliki jumlah keseluruhan bit 128. Oleh karena itu, untuk melihat seberapa berpengaruhkah penggunaan *Message-Digest* ini terhadap metode *Vigenere*.

Dalam perubahan MD2, MD2 memiliki jumlah perubahan karakter yaitu 63 bit, untuk MD4 memiliki perubahan karakter 75 bit dan yang terakhir MD5 memiliki jumlah perubahan karakter 62 bit. Jika masing-masing *Message-Digest* dihitung menggunakan *Avalanche Effect* maka hasilnya adalah MD2 memiliki nilai AV 49.22%, untuk MD4 memiliki nilai AV 58.59% dan yang terakhir yaitu MD5 memiliki nilai AV 48.44%. Dari hasil perhitungan testing *Avalanche Effect* diatas, maka diperoleh bahwa penggunaan *Message-Digest* sangat penting terhadap keamanan terutama menggunakan MD4 yang jika dilihat pada grafik diatas memiliki nilai yang cukup tinggi terhadap keamanan yang nantinya dikombinasikan dengan *vigenere*.

Black Box Testing

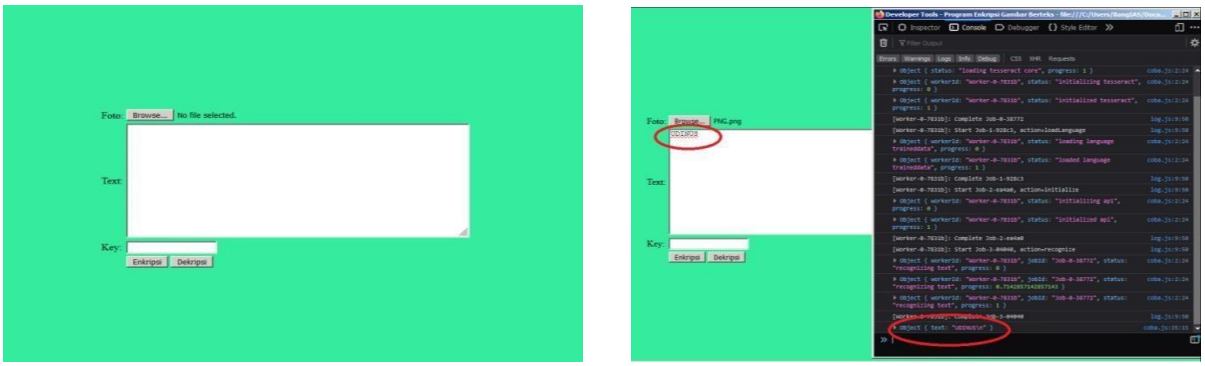
Untuk menguji seberapa baik kualitas dan fungsionalitas dari program yang penulis buat, maka penulis menerapkan *Black Box Testing* sebagai pengujiannya. Seperti pada sub-bab sebelumnya pada pembahasan *Analisa Pengujian* didapatkanlah sebuah hasil pengujian menggunakan Black Box Testing seperti pada Tabel 5.

Tabel 5. Black Box Testing

Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Input kunci vigenere untuk enkripsi hasil dari OCR.	Input Key: "Lulus"/"abc"	Input key atau kunci vigenere dalam sistem dapat dilakukan	Sesuai Harapan	Valid
Input File gambar teks.	Gambar diinput dalam program	OCR menganalisa gambar teks dengan lancar	Sesuai Harapan	Valid
Hasil dari analisa OCR.	Hasil dari gambar yang di input	Hasil akan diolah lagi dalam OCR.	Sesuai Harapan	Valid
Output hasil dari analisa OCR akan keluar dengan bentuk terenkripsi oleh sandi vigenere.	Hasil terenkripsi	Output yang keluar adalah enkripsi dari proses sandi Vigenere yang sudah di analisa oleh OCR	Sesuai Harapan	Valid

Evaluasi

Untuk melihat hasil dari penelitian yang penulis buat, penulis mencoba mengimplementasi hasilnya dengan menggunakan JavaScript yang diimplementasikan pada web. Hasil implementasi yang penulis buat pada Gambar 2, dapat di kembangkan lagi pada penelitian selanjutnya yang dapat digunakan untuk mengamankan data lebih baik lagi.



(a) (b)

Gambar 2. (a) Hasil Homepage, (b) Hasil Ekstraksi dari OCR

Pada homepage program dalam Gambar 2 point a, terdapat menu foto yang nantinya dapat digunakan untuk upload citra teks dan output yang akan muncul ditampung pada form text yang nantinya proses Enkripsi dan Dekripsi juga akan dilakukan pada form text. Sebelum Enkripsi maupun Dekripsi dilakukan, maka kunci harus dimasukan terlebih dahulu agar proses enkripsi maupun dekripsi dapat berjalan. Karena pada dasarnya, metode vigenere menggunakan kunci sebagai alat enkripsi dan dekripsinya. Gambar 2 poin b diatas merupakan hasil dari ekstraksi proses OCR, yang mana prosesnya sudah penulis jelaskan pada pembahasan sebelumnya. Proses ekstraksi tersebut, memisahkan antara teks dengan background gambar.

5. KESIMPULAN

Dari hasil penelitian yang dilakukan dapat diambil kesimpulan bahwa pengamanan teks pada citra bertulis sangat diperlukan apalagi citra bertulis merupakan citra yang mempunyai keamanan yang sensitif. Dengan menggunakan algoritma *Template Matching* sebagai metode pengenalan huruf untuk proses *Optical Character Recognition* atau OCR, dapat diketahui bahwa proses ekstraksi dari citra ke bentuk teks dapat dilakukan untuk enkripsi atau mengamankan teks pada citra bertulis agar tidak mudah untuk dimanipulasi atau diubah tanpa seijin pemiliknya. Dalam hasil *testing* yang penulis lakukan terhadap metode kriptografi *vigenere* dengan menggunakan *Avalanche Effect* sebagai alat hitung *testing*-nya, metode kriptografi *vigenere* sudah cukup aman untuk

mengamankan teks pada citra bertulis dan apabila penggunaan metode kriptografi *vigenere* dengan *Message-Digest* akan lebih aman lagi terutama *Message-Digest* 4. Hasil penelitian yang penulis buat, belum sempurna dan belum begitu baik, oleh karena itu agar program ini sempurna, maka diperlukannya pengembangan lebih lanjut, terutama pada metode keamanannya. Oleh karena itu dibutuhkan sebuah metode lagi untuk membentuk kekuatan pada sistem keamanan.

DAFTAR PUSTAKA

[1] Y. Anggraini and D. V. S. Y. Sakti, "Penerapan Steganografi Metode End of File (Eof) Dan Enkripsi Metode Data Encryption Standard (Des) Pada Aplikasi Pengamanan Data Gambar Berbasis Java," in *Konferensi Nasional Sistem Informasi, STMIK Dipanegara Makassar*, 2014, no. September 2016, pp. 1743–1753.

[2] F. Anwar, E. H. Rachmawanto, C. A. Sari, and de Rosal Ignatius Moses Setiadi, "StegoCrypt Scheme using LSB-AES Base64," in *2019 International Conference on Information and Communications Technology, ICOIACT 2019*, 2019.

[3] C. Sari, E. Rachmawanto, Y. Astuti, and L. Umaroh, "Optimasi penyandian file menggunakan kriptografi shift cipher," in *Seminar Multi Disiplin Ilmu Unisbank (SENDI_U) ke-2 Semarang*, 2016.

[4] F. Patel, J. Solanki, V. Rajguru, and A. Saxena, "Recognition of Vehicle Number Plate Using Image Processing Technique," *Adv. Emerg. Med.*, vol. 7, no. 1, pp. 2–8, 2018.

[5] V. Ong and D. Suhartono, "Using K-Nearest Neighbor in Optical Character Recognition," *ComTech Comput. Math. Eng. Appl.*, vol. 7, no. 1, p. 53, 2016.

[6] A. Chaudhuri, K. Mandaviya, P. Badelia, and S. K Ghosh, "Optical Character Recognition Systems for Different Languages with Soft Computing," in *Optical Character Recognition Systems*, vol. 352, Springer International Publishing AG 2017, 2017, pp. 9–42.

[7] P. Vithlani and C. K. Kumbharana, "A Study of Optical Character Patterns identified by the different OCR Algorithms," *Int. J. Sci. Res. Publ.*, vol. 5, no. 3, pp. 1–5, 2015.

[8] E. Setyaningsih, C. Iswahyudi, and N. Widyastuti, "Image Encryption on Mobile Phone using Super Encryption Algorithm," *Telkomnika*, vol. 10, no. 4, pp. 837–845, 2012.

[9] W. S. Sari, E. H. Rachmawanto, D. R. I. M. Setiadi, and C. A. Sari, "A Good Performance OTP encryption image based on DCT-DWT steganography," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 15, no. 4, 2017.

[10] P. Subhasri and A. Padmapriya, "Enhancing the security of dicom content using modified vigenere cipher," *Int. J. Appl. Eng. Res.*, vol. 10, no. 55, pp. 1951–1956, 2015.

[11] A. Kulsoom, D. Xiao, Aqeel-ur-Rehman, and S. A. Abbas, "An efficient and noise resistive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules," *Multimed. Tools Appl.*, vol. 75, no. 1, pp. 1–23, 2016.

[12] F. Mohammad, J. Anarase, M. Shingote, and P. Ghanwat, "Optical Character Recognition Implementation Using Pattern Matching," *Faisal Mohammad al,/(IJCSIT) Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 2088– 2090, 2014.

[13] R. Gunawan, S. Suwarno, and W. Hapsari, "Penerapan Optical Character Recognition (OCR) untuk Pembacaan Meteran Listrik PLN," *Informatika*, vol. 10, no. 2, pp. 127–134, 2014.

[14] A. Susanto, D. R. I. Moses Setiadi, E. H. Rachmawanto, C. A. Sari, R. R. Ali, and I. U. Wahyu Mulyono, "Dual Security Method for Digital Image using HBV Encryption and Least Significant Bit Steganography," in *Journal of Physics: Conference Series*, 2019, vol. 1201, no. 1.

[15] I. Saputra, Mesran, N. A. Hasibuan, and R. Rahim, "Vigenere Cipher Algorithm with Grayscale Image Key Generator for Secure Text File," *Int. J. Eng. Res. Technol.*, vol. 6, no. 1, pp. 266–269, 2017.

[16] S. D. Nasution, G. L. Ginting, M. Syahrizal, and R. Rahim, "Data Security Using Vigenere Cipher and Goldbach Codes Algorithm," *Int. J. Eng. Res. Technol.*, vol. 6, no. 01, pp. 360–363, 2017.