Terbit *online* pada laman : https://journal.fkpt.org/index.php/BIT
BULLETIN OF INFORMATION TECHNOLOGY (BIT)
ISSN (Media Online) 2722-0524

Implementasi Algoritma Mars Pada Penyandian Citra Digital USG

Alifah Caniago[1], Muhmmad Syahrizal[2], Pristiwanto[3]

[1,2,3] Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: [1]alifahcan@gmail.com

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi : 13 Mei 2020
Revisi Akhir : 23 Mei 2020
Diterima : 30 Mei 2020
Diterbitkan *Online* : 08 Juli 2020

KATA KUNCI

Kriptografi,
Penyandian,
Citra Digital,
USG,
Algoritma Mars

KORESPONDENSI

E-mail: alifahcan@gmail.com

A B S T R A C T

Citra Digital adalah salah satu bentuk data digital saat ini yang banyak dipakai untuk menyimpan photo, gambar ataupun hasil karya dalam format digital, salah satunya adalah citra usg. Dalam bidang medis usg dapat digunakan untuk mendiagnosa beberapa penyakit dalam organ manusia, salah satunya organ hati. Citra digital sangat rentan terhadap penyadapan maupun pencurian data oleh pihak- pihak yang tidak bertanggung jawab. Demi menjaga keamanan citra usg dapat dilakukan dengan pemanfaatan teknik kriptografi.Teknik kriptografi dapat menyandikan citra digital usg dengan mengenkripsikannya ke dalam bentuk sandi-sandi yang tidak dipahami.Dalam penelitian ini menggunakan algoritma Mars.Untuk membangun aplikasi yang terkomputerisasi ini dengan menggunakan Visual Basic 2008 sebagai aplikasi pendukungnya. Aplikasi ini dibuat sebagai upaya untuk meminimalisir tindakan-tindakan penyalahgunaan citra usg.

1. PENDAHULUAN

Kriptografi adalah ilmu yang berdasarkan pada teknik matematika yang erat kaitannya dengan keamanan informasi seperti kerahasiaan, keutuhan data dan autentikasi entitas. Jadi pengertian kriptografi modern adalah bukan hanya penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi[1].

Citra digital adalah salah satu cabang ilmu informatika yang mempelajari mengenai gambar, cara pengolahannya, serta implementasinya dalam kehidupan sehari – hari, usg adalah contohnya. *Ultrasonografi* atau yang lebih dikenal Usg digunakan luas dalam bidang medis, diantaranya mendiagnosa bagian organ tubuh dalam manusia, salah satunya adalah usg hati. Pelaksanaan prosedur diagnosis dapat dilakukan dengan bantuan ultrasonografi, biasanya menggunakan *probe* yang digenggam yang diletakkan diatas pasien dan digerakkan : gel berair agar memastikan penyerasian antara pasien dan *probe*[2].

Perkembangan teknologi komputerisasi saat ini sudah sangat meningkat dengan kebutuhan informasi bagi pengguna komputer. Semakin tinggi teknologi komputer, semakin tinggi tingkat ancaman yang mengancam keamanan pengguna komputer. Pengguna perlu untuk menyimpan data berupa file gambar yang berisi informasi yang berharga, walaupun melalui data digital disebut sebagai sarana yang paling aman untuk saling menyimpan informasi, namun nyatanya tidak demikian, banyak sekali orang yang memiliki kemampuan untuk menyusup ke dalam jaringan informasi tersebut yang menyebabkan informasi dapat dibaca, diambil, atau bahkan dimanipulasi dan dipublikasikan. Informasi yang dimiliki oleh pemilik bisa dirugikan apabila ada kegiatan mengambil tanpa seijin pihak yang bersangkutan. Hal seperti itulah yang menjadi ancaman yang perlu dilindungi karena dapat merugikan pihak yang bersangkutan.

Berdasarkan penelitian sebelumnya yang dilakukan oleh Irfan, Pahrul Prayudi, Yudi, Seminar Nasional Aplikasi Teknologi Informasi, tahun 2015. Menyatakan bahwa salah satu tipe file yang banyak digunakan dan biasanya berisi informasi penting adalah data bertipe gambar atau citra digital. Saat ini citra telah digunakan pada hampir segala bidang seperti rancangan keamanan, ilmu medis, ilmu teknik mesin, arsitekur bangunan, hasil karya seni, iklan, pendidikan dan lain sebagainya. Citra yang disimpan atau ditransmisikan dalam bentuk plainimage rentan terhadap penyadapan atau pencurian, sehingga informasi penting yang terdapat dalam citra dapat diakses oleh pihak - pihak yang tidak bertanggung jawab. Jika citra tersebut dapat diakses oleh orang yang tidak berhak, tentunya pemilik akan mendapat kerugian baik dari segi finansial atau yang lainnya. Oleh karena itu pengamanan terhadap citra menjadi perhatian penting untuk melindungi informasi yang terdapat di dalamnya[3].

Hidayat, Arinten Dewi Afrianto, Irawan, Jurnal Tenik Informatika, tahun 2017, volume IX, Juni. Juga menyatakan dalam penelitiannya bahwa solusi terhadap keamanan citra digital dari penyadapan atau serangan adalah dengan mengekripsinya. Enkripsi citra merupakan teknik untuk melindungi citra dengan cara menyandikan citra (*plain-image*) sehingga tidak dapat dikenali lagi(*chiper-image*)[1].

Salah satu metode kriptografi yang dapat digunakan untuk mengimplementasikan pengamanan citra digital usg adalah dengan menggunakan algoritma mars, dimana algoritma ini dapat menyelesaikan proses enkripsi dan dekripsi

dengan cepat. Operasi xor pada *mars* melibatkan penjumlahan, perkalian, dan pembagian untuk mengabungkan nilai data dan nilai kunci. Penelitian ini menguraikan bagaimana menerapkan algoritma *mars* pada penyandian citra digital *usg* agar data tetap aman dan sulit terpecahkan. Maka penulis mencoba menggunakan kriptografi modern berupa algoritma *mars.* Agar proses yang dilakukan lebih mudah, maka dirancang sebuah aplikasi pengamanan citra digital usg menggunakan bahasa pemrograman *visual basic* 2008.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa yunani, yaitu dari kata *cypto* dan *graphia* yang berarti penulisan rahasia. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. Kriptografi juga merupakan studi terhadap teknik matematis yang terkait dengan aspek keamanan suatu sistem informasi seperti kerahasiaan, integritas data, autentikasi dan ketiadaan penyangkalan[1].

2.2 Citra Digital

Citra (i*mage*) adalah representasi optis dari sebuah obyek yang disinari oleh sebuah sumber radiasi. Pada dasarnya citra yang dilihat terdiri atas berkas-berkas cahaya yang dipantulkan oleh benda-benda disekitarnya. Salah satu bentuk citra adalah citra yang mengandung abstrak dari citra matematis yang berisi fungsi kontinu dan fungis diskrit atau citra digital. Citra yang memiliki fungsi diskrit inilah yang dapat diolah oleh komputer. Setiap citra digital memiliki beberapa karakterstik, antara lain ukuran citra, resolusi dan format nilainya[6].

2.3 Algoritma Mars

Algoritma mars adalah salah satu algoritma kriptografi *chiper* blok, dengan ukuran blok 128 bit dan ukuran kunci yang bervariasi dari 128 bit sampai 400 bit (Burwick et al. 1998)[4]. *Multivariate Adaptive Regression Splines* (MARS) merupakan metode dengan pendekatan regresi nonparametrik yang pertama kali diperkenalkan oleh Friedman pada tahun 1991. Model MARS berguna untuk mengatasi permasalahan data berdimensi tinggi dan menghasilkan prediksi variabel respon yang akurat, dan menghasilkan model kontinu dalam *knot* berdasarkan nilai *Generalized Cross Validation* (GCV) terkecil. Permasalahan berdimensi tinggi adalah suatu permasalahan dengan jumlah variabel yang banyak serta ukuran sampel yang besar sehingga memerlukan perhitungan yang rumit. Data berdimensi tinggi yang dimaksud adalah data dengan ukuran $3 \leq v \leq 20$, dimana v adalah banyak variabel prediktor dan sampel data yang berukuran $50 \leq N \leq 1000$, dimana N untuk ukuran sampel (Friedman, 1991)[5]. Notasi yang digunakan dalam *chiper* adalah :

1. D[ ] adalah suatu array dari 4 32 bit data word. Array ini berisikan *plaintext* dan pada akhir proses enkripsi berisikan *chipertext*.
2. K[ ] adalah array untuk *expended key*, terdiri dari 40 32 bit.
3. S[ ] adalah array yang berisikan *S-box*, terdiri dari 512 bit word.

Perluasan kunci berfungsi untuk membangkitkan sub kunci dari kunci yang diberikan yakni K[ ] terdiri dari n 32 bit dan diperluas menjadi 64 bit sub kunci K[ ]. Tahapan yang dilakukan untuk membangkitkan sub kunci menggunakan modifikasi dari algoritma DES pada perluasan kunci adalah:

1. Konversi karakter kunci menjadi biner
2. Lakukan *Permutation Commpresiton* 1 (PC-1) terhadap biner kunci sesuai dengan ketentuan tabel PC-1.
3. Biner kunci hasil PC-1 dibagi menjadi 2 kelompok yaitu C_0 dan D_0.
4. Lakukan proses *Shift left* terhadap C_0 dan D_0sebanyak 16 kali berdasarkan aturan jumlah perpindahan bit disetiap putaran
5. Gabungkan kembali setiap Ci dan Di hasil *Shift left* .
6. Masing-masing hasil penggabungan Ci dan Di di *Permutation Commpresion* 2 (PC-2) sesuai dengan tabel PC-2 untuk menghasilkan internal key (*subkey*).

3. ANALISA DAN PEMBAHASAN

3.1 Analisa Masalah

Salah satu citra digital yaitu citra usg merupakan suatu informasi rahasia yang dapat digunakan dalam bentuk bidang medis. Maka dari itu citra usg perlu diamankan agar tidak di salahgunakan oleh pihak-pihak yang tidak bertanggung jawab. Apabila Citra usg tersebar luaskan tanpa adanya pengamanan maka dapat menimbulkan kerugian. Keamanan pada citra usg dapat dilakukan dengan salah satu teknik kriptografi. Agar algoritma dapat berjalan dengan baik terhadap enkripsi citra digital usg, maka terlebih dahulu citra di konversi dalam bentuk biner pada setiap *pixel* citra. citra yang akan diamankan pada kasus ini adalah citra *grayscale.*

3.2 Penerpan Algoritma Mars

Berikut ini adalah proses hasil nilai pixel dari matlab yang akan dipakai untuk prosese enkripsi. Nilai elemen warna dari 16 piksel *plainimage* contoh adalah = (20,97,116,80,62,72,170,184,35,62,86,72,19,87,57,51).

Dari metode ini proses yang akan di jalankan terdiri dari ekspansi atau pembangkit kunci, proses enkripsi, dan proses dekripsi .

Plainimage **: 20 97 116 80 62 72 170 184 35 62 86 72 19 87 57 51**
Key **: ALIFAH_C**
***Plainimage* :**

Char	**20**	**97**	**116**	**80**
Bin	**00010100**	**01100001**	**01110100**	**01010000**

Char	**62**	**72**	**170**	**184**
Bin	**00111110**	**01001000**	**10101010**	**10111000**

Char	**35**	**62**	**86**	**72**
Bin	**00100011**	**00111110**	**01010110**	**01001000**

Char	**19**	**87**	**57**	**51**
Bin	**00010011**	**01010111**	**00111001**	**00110011**

Proses pembangkitan kunci karena ada 16 putaran, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu K1, K2, …, K16. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64 bit atau 8 karakter. Lakukan Permutation Commpression (PC-1) terhadap biner kunci sesuai dengan tabel PC-1. Hal ini di lakukan untuk mengkompresikan 64 bit kunci eksternal menjadi 56 bit.

Tabel 1. *Permutation Commpression* (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	45	38	30	22
14	6	61	53	45	37	29
21	13	5	20	20	12	4

Cara melakukan nya cari bit pada posisi ke-57 dan pindahkan pada posisi ke-1, cari bit ke 49 dan pindah kan pada posisi ke-2 cari posisi ke 41 dan pindah kan pada posisi ke-3, dan seterusnya.

****Key*** **:**

Char	A	L	I	F	A
Des	65	76	73	70	65
Bin	01000001	01001100	01001001	01000110	01000001

Char	H	_	C
Des	72	95	67
Bin	01001000	01011111	01000011

Gabungkan semua biner kunci kemudian lakukan Permutasi Compresi-1 (PC-1) untuk mendapatkan 56 bit pra kunci.

*Biner Kunci:
01000001010011000100100101000110010000010100100000101111101000011
*Hasil PC-1
00000000111111110000000001001100100001101010011001010000

Hasil PC-1 di bagi menjadi 2 kelompok yang terdiri dari C_0 dan D_0 yang masing-masing terdiri dari 28 bit:

C_0 : 0000000011111111000000000100
D_0 : 1100100001101010011001010000

Proses Generate key (Pembangkitan Kunci) left shift operation sebanyak 16 iterasi

Tabel 2. Generate key (Pembangkit Kunci)

Putaran	Jumlah Putaran	C_0	D_0
		0000000011111111000000000100	1100100001101010011001010000
1	1	0000000111111110000000001000	1001000011010100110010100001
2	1	0000001111111100000000010000	0010000110101001100101000011
3	2	0000111111110000000001000000	1000011010100110010100001100
4	2	0011111111000000000100000000	0001101010011001010000110010
5	2	1111111100000000010000000000	0110101001100101000011001000

Putaran	Jumlah Putaran	C_0	D_0
6	2	11111100000000001000000000011	10101001100101000011001000 01
7	2	11110000000001000000000001111	10100110010100001100100001 10
8	2	11000000000100000000000111111	10011001010000110010000110 10
9	1	10000000001000000000001111111	00110010100001100100001101 01
10	2	00000000100000000000111111110	11001010000110010000110101 00
11	2	00000010000000000011111111000	00101000011001000011010100 11
12	2	00001000000000001111111100000	10100001100100001101010011 00
13	2	00100000000000111111110000000	10000110010000110101001100 10
14	2	10000000000011111111000000000	00011001000011010100110010 10
15	2	00000000011111111100000000010	01100100001101010011001010 00
16	1	00000000111111111000000000100	11001000011010100110010100 00

Proses generate key (pembangkit kunci) penggabungan kembali C_0& D_0 hasil left shift operation dan lakukan PC-2. Cara melakukan nya cari bit pada posisi ke-14 dan pindahkan pada posisi ke-1, dan seterusnya.

Tabel 3. Permutation Commpression (PC-2)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Setelah di lakukan Generate key maka di hasilkan kunci internal untuk proses Enkripsi:

Tabel 4. Kunci Internal

Round	**Biner Kunci**
K[1]	101000001001001001000010000100011100010100110011
K[2]	101100000001001001010010101101000101101100001100
K[3]	001001000101001001010000000100000001001011110010
K[4]	000001100101000101010100110101011010100000100001
K[5]	000011100100000101010001001000100010111001011000
K[6]	000011110100000100101001001110011011000100010110
K[7]	100010110000000110001001001001010100010010100010
K[8]	000110010000101010001001010011000010100001000111
K[9]	001110010000100010001000011100001000000100110001
K[10]	000100000010100010001100110000110010110000000010
K[11]	000100000010110000010100011011000011001100011000
K[12]	010001000010110000100100001100010101000001101110
K[13]	110000101010010000100100010001001001100010100010
K[14]	110010001000011000100010100001000010110001111101
K[15]	111000001001001000101010001010111001101011010000
K[16]	101000001001001010100010110010100010001000101010

Proses Enkripsi

Langkah 1

P1 = 00010100 00000000 00000000 00000000 00000000 00000000 ⊕

K[1] = 10100000 10010010 01000010 00010001 11000101 00110011

C1 = 10110100 10010010 01000010 00010001 11000101 00110011

Langkah 2

P2 = 01100001 00000000 00000000 00000000 00000000 00000000 ⊕

K[2] = 10110000 00010010 01010010 10110100 01011011 00001100

C2 = 11010001 00010010 01010010 10110100 01011011 00001100

Langkah 3

P3 = 01110100 00000000 00000000 00000000 00000000 00000000 ⊕

K[3] = 00100100 01010010 01010000 00010000 00010010 11110010

C3 = 01010000 01010010 01010000 00010000 00010010 11110010

Langkah 4

P4 = 01010000 00000000 00000000 00000000 00000000 00000000 ⊕

K[4] = 00000110 01010001 01010100 11010101 10101000 00100001

C4 = 01010110 01010001 01010100 11010101 10101000 00100001

Setelah proses enkripsi di lakukan maka mendapatkan hasil cipherimage dari algoritma, dan pembentukan cipherimage di ambil dari setiap hasil 8 bit pertama pada setiap putaran hingga putaran ke 16, maka cipherimage yang di hasilkan adalah :

***Cipherimage**:

Bin	10110100	11010001	01010000	01010110
Des	180	209	80	86
Char	'	Ñ	P	V

Bin	00110000	01000111	00100001	10100001
Des	48	71	33	161
Char	**0**	G	!	i

Bin	00011010	00101110	01000110	00001100
Des	26	46	70	12
Char	→	.	F	♀

Bin	11010001	10011111	11011001	10010011
Des	209	159	217	147
Char	Ñ	Ÿ	Ú	”

Cipherimage: ' Ñ P V 0 G ! I → . F ♀ Ñ Ÿ Ú ”

Proses Dekripsi

Untuk mendapatkan hasil plainimage dari proses dekripsi ini maka di lakukan langkah seperti pada proses enkripsi :

Langkah 1

C1 = 10110100 00000000 00000000 00000000 00000000 00000000 ⊕
K[1] = 10100000 10010010 01000010 00010001 11000101 00110011
P1 = 00010100 10010010 01000010 00010001 11000101 00110011

Langkah 2

C2 = 11010001 00000000 00000000 00000000 00000000 00000000 ⊕
K[2] = 10110000 00010010 01010010 10110100 01011011 00001100
P2 = 01100001 00010010 01010010 10110100 01011011 00001100

Langkah 3

C3 = 01010000 00000000 00000000 00000000 00000000 00000000 ⊕
K[3] = 00100100 01010010 01010000 00010000 00010010 11110010
P3 = 01110100 01010010 01010000 00010000 00010010 11110010

Langkah 4

C4 = 01010110 00000000 00000000 00000000 00000000 00000000 ⊕
K[4] = 00000110 01010001 01010100 11010101 10101000 00100001
P4 = 01010000 01010001 01010100 11010101 10101000 00100001

Setelah proses dekripsi di lakukan maka mendapatkan hasil plainimage dari algoritma kedua, dan pembentukan plainimage di ambil dari setiap hasil 8 bit pertama pada setiap putaran, maka plainimagenya adalah :

Bin	00010100	01100001	01110100	01010000
Char	20	97	116	80
Des	¶	A	t	P

Bin	00111110	01001000	10101010	10111000
Char	62	72	170	184
Des	>	H	a_	¸

Bin	00100011	00111110	01010110	01001000
Char	35	62	86	72
Des	#	>	V	H

Bin	00010011	01010111	01111001	00110011
Char	19	87	57	51
Des	!!	W	9	2

Plainimage: **20, 97, 116, 80, 62, 72, 170, 184, 35, 62, 86, 72, 19, 87, 57, 51**

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 1, No. 2, Juli 2020, pp 68- 74
ISSN 2722-0524 (media online)

4. IMPLEMENTASI

4.1 Implementasi Sistem

Implementasi merupakan langkah yang digunakan untuk mengoperasikan rancangan yang dibangun. Dalam bab ini dijelaskan bagaimana menjalankan sistem tersebut. Sistem pengolahan program merupakan suatu kesatuan pengolahan yang terdiri prosedur dan pelaksanaan data. Komputer sebagai sarana pengolahan program haruslah menyediakan fasilitas-fasilitas pendukung dalam pengolahan nantinya. Secara proposional haruslah memenuhi akses yaitu Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*)

Bentuk tampilan *form Login* yang di desain untuk masuk kedalam akses program dengan mengunakan *username* dan *password.* Dan bentuk tampilan *form* penyandian citra usg saat di jalankan akan menampilkan menu utama yang terdiri dari menu untuk memilih *image*, *textbox* untuk pengimputan kunci, kotak *plainimage* untuk memasukan image yang akan di enkripsi dengan *button* enkripsi, kotak *cipherimage* untuk menampilkan hasil dekripsi dari proses enkripsi dengan *button* dekripsi.

Gambar 1. *Interface* Tampilan *Form Login*

Pada Gambar 1. tampilan *form login* ini didesain minimalis agar pengguna aplikasi tidak sulit untuk memahami cara masuk atau *login* ke dalam program tersebut.

Gambar 2. *Interface* Tampilan *Form* Menu Utama

Pada Gambar 2. tampilan *form* menu utama ini didesain minimalis agar pengguna aplikasi penyandian citra digital usg tidak sulit memahami bagaimana cara menggunakan maupun mengoperasikan aplikasi ini. Bentuk tampilan *input* dan *output* penyandian citra digital usg yang akan tampil pada menu utama yang akan dijalankan proses enkripsi dan dekripsi

Gambar 6. Tampilan *Input Plainimage usg*

Pada gambar 3. untuk pertama pengguna harus memilih file *image* mana yang akan di enkripsikan kemudian pengguna mengisi kotak kunci lalu menekan tombol enkripsi agar image dapat tersandikan.

Gambar 4. Tampilan *Output Cipherimage* Usg

Pada gambar 4. pada proses ini *image* yang sudah terenkripsi yang berbentuk kode-kode dapat di simpan ke dalam kotak *listview*, dan jika ingin mengembalikan *image* ke bentuk semula dapat mengunakan tombol dekripsi.

5. KESIMPULAN

Berdasarkan uraian dari bab-bab sebelumnya, maka penulis dapat memberikan kesimpulan sebagai berikut:

1. Proses penyandian citra digital usg dapat dilakukan dengan algoritma Mars sehingga gambar asli atau informasi tidak dapat dibaca dan dimengerti oleh sembarang pihak.
2. Implementasi Algoritma *Mars* dalam proses penyandian citra digital usg dapat disandian dan menggunakan penyisipan kunci yang ingin disisipkan agar tidak sesuai dengan gambar aslinya.
3. Perancangan aplikasi dengan menggunakan *Visual Basic* 2008 yang telah selesai dirancang dengan desain minimalis diharapkan dapat berguna dalam penyandian c*itra usg.*

REFERENCES

[1] C. Jhony and M. Sianturi, "Analisis Segmentasi Citra USG Hati Menggunakan Metode Fuzzy C-Mean," Citec J., vol. 2, pp. 256–264, 2016.
[2] P. Irfan and Y. Prayudi, "Penggabungan Algoritma Chaos Dan Rivers Shamir Adleman (Rsa) Untuk Peningkatan Keamanan Citra," in Seminar Nasional Apliasi Tenologi Informasi, 2015, pp. 5–10.
[3] E. Setyaningsih, S.Si., M.Kom, Kriptografi & implementasinya menggunakan MATLAB, 1st ed. Yogyakarta: Andipublisher, 2015.
[4] A. P. N. Nurdin, "Analisa Dan Implementasi Kriptografi Pada Pesan Rahasia," Jesik, vol. 3, no. 1, pp. 1–11, 2017.
[5] C. Algoritma, "Ketepatan Klasifikasi Status Pemberian Air Susu Ibu (ASI) Menggunakan Multivariate Adaptive Regression Splines (MARS) dan," vol. 5, pp. 229–238, 2016.
[6] T.sutoyo, Teori pengolahan citra Digital, 1st ed. Yogyakarta: www.andipublisher.com, 2009.
[7] Hondro, R. K., & Nurcahyo, G. W. (2018). Analisis dan Perancangan Sistem yang Menerapkan Algoritma Triangle Chain Cipher (TCC) untuk Enkripsi Record Tabel Database.
[8] Zebua, T., Hondro, R. K., & Ndruru, E. (2018). Message Security on Chat App based on Massey Omura Algorithm. IJISTECH (International Journal Of Information System & Technology), 1(2), 16-23.
[9] Hondro, R. K. (2018). Aplikasi Enkripsi dan Dekripsi SMS dengan Algoritma Zig Zag Cipher pada Mobile Phone Berbasis Android. Pelita Informatika: Informasi dan Informatika, 10(3).
[10] B. J. Hutapea, M. A. Hasmi, and A. Karim, "Sistem Pendukung Keputusan Penentuan Jenis Kulit Terbaik Untuk Pembuatan Sepatu Dengan Menggunakan Metode Vikor," *JURIKOM(Jurnal Ris. Komputer)*, vol. 5, no. 1, pp. 6–12, 2018.