PRISMA

Prosiding Seminar Nasional Matematika

Diterbitkan oleh:
Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Semarang

PRISMA	Volume 3		Semarang Februari 2020	ISSN 2613-9189

ISSN 2613-9189

Editorial

Alamat Korespondensi:
Jurusan Matematika
Universitas Negeri Semarang
Gedung D7 Lantai 1
Fakultas Matematika dan Ilmu
Pengetahuan Alam
Kampus Sekaran, Kel. Sekaran, Kec. Gunung
Pati, Semarang, Jawa Tengah 50229
☎🖷 (024) 8508032.
email: matematika@mail.unnes.ac.id
website: http://matematika.unnes.ac.id

Articles

Implementasi Education 4.0 dan Merdeka Belajar dalam Matematika di Perguruan Tinggi

Basuki Widodo

1-7

PDF

Problematika Pembelajaran Matematika bagi Masyarakat Indonesia Kontemporer

Hardi Suyitno

8-19

PDF

Kemampuan Komunikasi Matematis Siswa Kelas XI dengan Model Brain Based Learning Berbantuan Mobile Learning Ditinjau dari Self-Concept

Achika Nor Kusyaini, Iwan Junaedi

20-25

PDF

Kemampuan Siswa pada Aspek Berpikir Kreatif Ditinjau dari Gaya Belajar Melalui Pembelajaran Problem Posing Berbasis Open-Ended Problem

Adieb Ajie Bayu Mukti, Edy Soedjoko

26-36

Penerapan Algoritma Hybrid Of Ant Colony And Discrete Firefly Algorithms (HADFA) Pada Capacitated Vehicle Routing Problem

Amalia Putri Nur Habibah, Kiswara Agung Santoso, Ahmad Kamsyakawuni

500-507

PDF

Application of Mobile-Based Visual Content for Schools During a Pandemic

Dian Tri Wiyanti, Isnaini Rosyida, Muhammad Kharis, Detalia Noriza Munahefi, Kholifatu Ulil Azmi

508-513

PDF

Penerapan Jaringan Syaraf Tiruan dengan Metode Learning Vector Quantization (LVQ) untuk Klasifikasi Penyakit Infeksi Saluran Pernapasan Akut (ISPA)

Endang Setyowati, Scolastika Mariani

514-523

PDF

Perbandingan Image RGB dan Grayscale pada Pengkodean Image dengan Algoritma 3D Playfair

Farokhi Abdiansyah, Kiswara Agung Santoso, Ahmad Kamsyakawuni

524-533

PDF

Modifikasi Huffman dengan hill cipher pada Pengkodean Data Teks

Giki Krisnawanti, Kiswara Agung Santoso, Ahmad Kamsyakawuni

534-539

PRISMA 4 (2021): 534-539
PRISMA, Prosiding Seminar Nasional Matematika
https://journal.unnes.ac.id/sju/index.php/prisma/
ISSN 2613-9189

Modifikasi *Huffman* dengan *Hill Cipher* pada Pengkodean Data Teks

Giki Krisnawanti[a,*], Kiswara Agung Santoso[a], Ahmad Kamsyakawuni[a]

[a] *Universitas Jember, Jalan Kalimantan No.37, Jember 68121, Indonesia*

[*]*Alamat Surel: gikikirna28@gmail.com*

Abstrak

Perkembangan teknologi yang pesat dapat mengakibatkan keamanan informasi menjadi sangat rentan untuk diketahui. Dalam menangani permasalahan tersebut, diperlukan suatu metode untuk menjaga keamanan informasi yang ingin disampaikan. Salah satu metode yang mampu menjaga keamanan informasi adalah dengan kriptografi. Algoritma yang dapat digunakan dalam kriptografi sangat beragam. Artikel ini membahas tentang modifikasi algoritma *Huffman* dengan *hill cipher*. Algoritma *Huffman* dipilih karena dapat melakukan pemampatan data sekaligus enkripsi, sedangkan *hill cipher* digunakan karena menggunakan kunci matriks $3x3$ yang mempunyai sifat unik. Modulo dalam hill cipher mengalami perubahan dari 26 menjadi 95. Selain itu, rumus *hill cipher* mengalami modifikasi dengan adanya pengurangan dan penambahan 32. Agar karakter yang digunakan berada dalam rentang ASCII *printable characters*. Metode yang digunakan adalah pembentukan kunci *hill cipher*, proses enkripsi, dan proses dekripsi. Pada proses dekripsi juga dapat mengembalikan *ciphertext* ke dalam bentuk *plaintext*.

Kata kunci:

Huffman, *hill cipher*, *ciphertext*, *plaintext*, kriptografi.

1. Pendahuluan

Perkembangan teknologi yang pesat membawa pengaruh sangat besar terhadap keamanan informasi. Sehingga dibutuhkan metode untuk menyelesaikan permasalahan tersebut. Salah satunya dengan ilmu kriptografi. Kriptografi merupakan ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikan ke dalam bentuk yang tidak dapat dipahami lagi agar tetap aman (Setyaningsih, 2015).

Algoritma yang dapat digunakan dalam kriptografi sangat beragam, diantaranya dengan algoritma *Huffman* dan *hill cipher*. Algoritma *Huffman* mengubah ukuran data menjadi lebih kecil dengan mengubah susunan data yang ada menjadi data baru yang tersandikan. Sebaliknya, *Huffman* juga dapat mengembalikan data yang sudah diproses ke bentuk awal. Kriptografi *Huffman* dasarnya merupakan kode prefiks. Kode prefiks biasanya dipresentasikan sebagai pohon biner yang berlabel, dengan setiap sisi diberi label 0 (cabang kiri) dan 1 (cabang kanan). Pohon biner ini disebut pohon *Huffman* yang akan menjadi kunci dalam proses enkripsi dan dekripsi (Pahdi, 2017).

Hill cipher merupakan salah satu kriptografi kunci simetri. *Hill cipher* menggunakan kunci matriks $n \times n$ yang sama dalam proses enkripsi dan dekripsinya, sehingga *plaintext* yang dihasilkan tidak selalu menghasilkan *ciphertext* yang sama (Rahmawati, 2017). Sedangkan kunci matriks yang digunakan merupakan matriks *invertible* yaitu $K.K^{-1} = I$ karena kunci K^{-1} yang akan digunakan dalam proses dekripsi (Hasugian, 2013). Rumus *hill cipher* pada proses enkripsi dan dekripsi dtunjukkan pada Persamaan (1) dan Persamaan (2).

$$C_i = (K \times P_i) \, mod \, 26 \tag{1}$$

$$P_i = (K^{-1} \times C_i) \, mod \, 26 \tag{2}$$

To cite this article:
Krisnawanti, G., Santoso, K. A., & Kamsyakawuni, A. (2021). Modifikasi *Huffman* dengan *Hill Cipher* pada Pengkodean Data Teks. *Prisma, Prosiding Seminar Nasional Matematika 4*, 534-539

Beberapa peneliti pernah menggunakan kedua algoritma ini, yaitu Pardede (2017) melakukan penelitian berupa "Algoritma *Vigenere Cipher* dan *Hill Cipher* dalam Aplikasi Keamanan Data pada *File* Dokumen". Penelitian tersebut mengombinasikan algoritma *vigenere cipher* dan *hill cipher* agar sulit memecahkan sandinya jika dibandingkan sebelum dikombinasikan. Auliyah (2020) melakukan penelitian "Implementasi Kombinasi Algoritma Enkripsi *Rivest Shamir Adleman* (RSA) dan Algoritma Kompresi *Huffman* pada *File Document*". Penelitian tersebut mengombinasikan antara metode enkripsi dan metode kombinasi. Metode kombinasi tersebut meliputi enkripsi-kompresi dan kompresi-enkripsi.

Berdasarkan penjelasan di atas, penulis melakukan modifikasi pada algoritma *Huffman* dengan *hill cipher* diharapkan agar tingkat kesulitannya lebih tinggi dibandingkan dengan sebelum modifikasi. Algoritma modifikasi tersebut juga diharapkan dapat mengamankan suatu pesan ketika dikirimkan ke penerima. Sehingga tidak akan ada pihak ketiga yang akan merubah isi pesan tersebut. Penulis menggunakan metode Algoritma *Huffman* karena dapat melakukan pemampatan data sekaligus enkripsi. Sedangkan *hill cipher* digunakan karena menggunakan kunci matriks yang mempunyai sifat unik.

2. Metode

Penelitian ini menggunakan data teks yang berupa karakter ASCII *printable characters*. Kunci *hill cipher* yang berukuran 3×3. Kunci yang digunakan merupakan matriks *invertible*.

2.1. Proses Enkripsi Modifikasi Huffman dengan hill cipher

Proses enkripsi pada penelitian ini menggunakan algoritma modifikasi *Huffman* dengan *hill cipher*. Berikut ini merupakan langkah-langkah proses enkripsi:

Langkah 1. Kunci matriks yang digunakan berukuran 3×3 dan harus memiliki invers sehingga $K.K^{-1} = I$. Kunci matriks digunakan pada proses enkripsi *hill cipher*.

Langkah 2. Karakter *plaintext* dienkripsi dengan menggunakan algoritma *Huffman*. Proses tersebut menghasilkan pohon *Huffman* dari *plaintext*. Pohon *Huffman* tersebut menginterpretasikan setiap karakter *plaintext* dengan kode biner dari masing-masing karakter tersebut.

Langkah 3. Kemudian dilakukannya proses XOR dengan 00001111. Aturan untuk operasi XOR tertadapat pada Tabel (1) (Azis, 2018). Hasilnya akan dikonversi ke dalam bilangan desimal.

Tabel 1. Aturan operasi XOR

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Langkah 4. Selanjutnya menjalankan Teorema *Euclidean* sesuai Persamaan (3) dengan $0 \leq r < n$ (Wardani & Kurniawan, 2019). Untuk nilai $q = 95$, nilai r adalah hasil operasi modulo q, dan nilai t akan digunakan sebagai kode yang dikirim ke penerima. Sedangkan untuk hasil proses XOR merupakan nilai m.

$$m = n \times q + r \tag{3}$$

Langkah 5. Hasil dari proses tersebut adalah nilai r dengan penambahan angka 32 pada masing-masing karakter. Kemudian dilakukannya proses enkripsi *hill cipher* yang ditunjukkan pada Persamaan (4). Dengan dilakukannya pengurangan dan penambahan nilai 32 agar karakter yang muncul dalam rentang $32 - 126$ yang sesuai dengan ASCII *printable characters*. Modulo yang digunakan juga mengalami perubahan menjadi 95.

$$C_i = (K \,.\, P_i - 32) \; mod \; 95 + 32 \tag{4}$$

Langkah 6. Selanjutnya hasilnya dikonversikan sesuai kode ASCII dan hasilnya menjadi karakter *ciphertext*.

2.2. Proses Dekripsi Modifikasi Huffman dengan hill cipher

Proses dekripsi pada penelitian ini menggunakan algoritma modifikasi *Huffman* dengan *hill cipher*. Berikut ini merupakan langkah-langkah proses dekripsi:

Langkah 1. Kunci yang digunakan merupakan invers dari kunci enkripsi.

Langkah 2. Karakter *ciphertext* dikonversikan ke dalam desimal dan kemudian didekripsi dengan *hill cipher* menggunakan *invers* kunci matriks pada proses enkripsi yang ditunjukkan pada Persamaan (5).

$$P_i = (K^{-1} . C_i - 32) \; mod \; 95 + 32 \tag{5}$$

Langkah 3. Lakukan pengurangan dengan bilangan 32 pada masing-masing karakter sehingga didapatkan nilai r. Sedangkan untuk nilai $q = 95$ dan kode t diperoleh pada proses enkripsi. Selanjutnya dari nilai-nilai tersebut untuk menjalankan Teorema *Euclidean* sesuai Persamaan (3) dan diperoleh nilai m.

Langkah 4. Kemudian lakukan proses XOR dengan 00001111 sesuai aturan Tabel (1).

Langkah5. Hasil dari proses tersebut dikonversikan ke dalam bilangkan biner dan dilakukannya proses enkripsi *Huffman*.

Langkah 6. Selanjutnya setiap karakter dikonversikan sesuai kode ASCII dan hasilnya menjadi karakter *plaintext*.

3. Hasil dan Pembahasan

Data yang digunakan dalam penelitian ini disajikan dalam Tabel (2) sebagai berikut:

Tabel 2. Data penelitian

Plaintext	***Kunci hill cipher***
Kodenya adalah 1#_&0	$K = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 1 & -1 \\ 0 & 1 & 3 \end{pmatrix}$

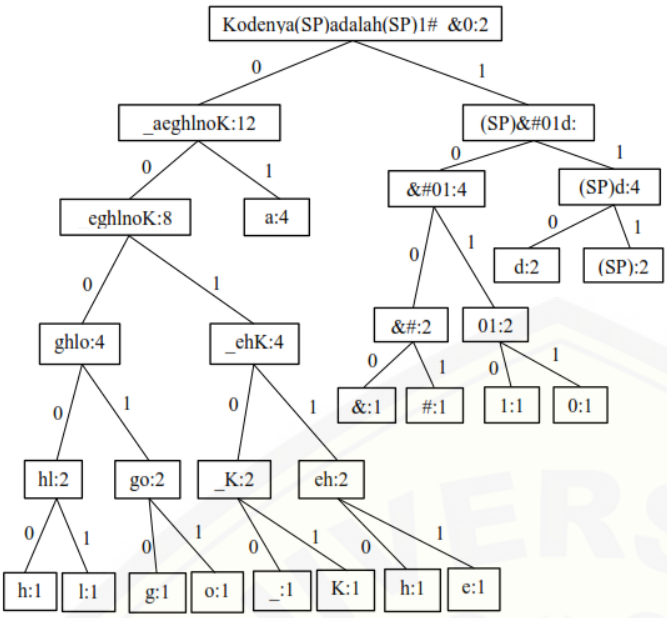
3.1. Pembentukan Kunci Hill Cipher

Pembentukan kunci *hill cipher* dilakukan dengan menyesuaikan elemen kunci sehingga dihasilkan matriks yang *invertible*. Misalkan kunci yang akan digunakan adalah kunci $K = \begin{pmatrix} 0 & 1 & 2 \\ -1 & 1 & -1 \\ 0 & 1 & 3 \end{pmatrix}$ dengan inversnya $K^{-1} = \begin{pmatrix} 4 & -1 & -3 \\ 3 & 0 & -2 \\ -1 & 0 & 1 \end{pmatrix}$.

3.2. Proses Enkripsi

Proses enkripsi modifikasi *Huffman* dengan *hill cipher* disajikan sebagai berikut:

Langkah 1. Karakter *plaintext* berupa **Kodenya adalah 1#_&0**. Kemudian hitung frekuensi setiap karakter *plaintext*. Buat pohon *Huffman* berdasarkan berdasarkan urutan karakter dari yang jumlahnya terbesar ke yang terkecil dan memberi kode untuk tiap karakter. Pohon *Huffman* disajikan pada Gambar 1. Kodekan setiap karakter dengan susunan bit biner hasil dari pohon *Huffman*. Kemudian ganti susunan *plaintext* dengan susunan bit biner yang dihasilkan sehingga "00101000111100011100000000100111101110010000101001101111010100100100100001011" dengan panjang 76 bit. Bagi biner menjadi 8 bit. Tambahkan biner 0 apabila tidak mencukupi. Sehingga menghasilkan "00101000 11110001 11000000 00100111 10111001 00001010 01101111 01010010 01001000 10110000".



Gambar 1. pohon *Huffman* "Kodenya adalah 1#_&0"

Langkah 2. Lakukan proses XOR dengan 00001111 yang hasilnya akan dikonversikan ke dalam desimal. Sehingga menghasilkan 39 254 207 40 182 5 96 93 71 191.

Langkah 3. Kemudian akan dimasukkan ke dalam Teorema *Euclidean* sebagai m dengan $q = 95$ yang disajikan pada Tabel (3).

Tabel 3. hasil teorema *Euclidean* enkripsi

No	n	r	m
1	0	39	39
2	2	64	254
3	2	17	207
4	0	40	40
5	1	87	182
6	0	5	5
7	1	1	96
8	0	93	93
9	0	71	71
10	2	1	191

Sehingga dihasilkan nilai $r + 32$ adalah 71 96 49 72 119 37 33 125 103 33.

Langkah 4. Kemudian lakukan proses *enkripsi hill* cipher dengan kunci matriks K sesuai dengan Persamaan (6). Bagi bilangan desimal menjadi kelompok-kelompok 3 anggota 71 96 49, 72 119 37, 33 125 103, 33 33 33 (ditambahkan bilangan terakhir agar memenuhi perhitungan). Sehingga menghasilkan 99 71 53 98 105 40 46 84 54 99 62 37.

$$C_i = \left(\begin{pmatrix} 0 & 1 & 2 \\ -1 & 1 & -1 \\ 0 & 1 & 3 \end{pmatrix} . P_i - 32 \right) mod\ 95 + 32 \quad (6)$$

Langkah 5. Kemudian hasilnya dikonversikan ke dalam karakter sesuai kode ASCII dan hasilnya menjadi **cG5bi(.T6c**.

3.3. Proses Dekripsi

Proses dekripsi modifikasi *Huffman* dengan *hill cipher* disajikan sebagai berikut:

Langkah 1. Karaker *ciphertext* berupa **cG5bi(.T6c** ditambahkan dengan kode Y hasil enkripsi dan dikonversikan ke dalam desimal, hasilnya 99 71 53 98 105 40 46 84 54 99. Kemudian didekripsi dengan *hill cipher* menggunakan invers kunci matriks pada proses enkripsi yang ditunjukkan pada Persamaan (7).

$$P_i = \left(\begin{pmatrix} 4 & -1 & -3 \\ 3 & 0 & -2 \\ -1 & 0 & 1 \end{pmatrix} . C_i - 32 \right) mod\ 95 + 32 \quad (7)$$

Langkah 2. Lakukan pengurangan dengan bilangan 32 pada masing-masing karakter sehingga didapatkan nilai r. Sedangkan untuk nilai $q = 95$ dan kode t diperoleh pada proses enkripsi. Selanjutnya dari nilai-nilai tersebut untuk menjalankan Teorema *Euclidean* yang disajikan pada Tabel (4). Kemudian diperoleh nilai m dan dikonversikan ke dalam bilangkan biner. sehingga menghasilkan 00100111 11111110 11001111 00101000 10110110 00000101 01100000 01011101 01000111 10111111.

Tabel 4. hasil teorema *Euclidean* dekripsi

No	m	n	r
1	39	0	39
2	254	2	64
3	207	2	17
4	40	0	40
5	182	1	87
6	5	0	5
7	96	1	1
8	93	0	93
9	71	0	71
10	191	2	1

Langkah 3. Selanjutnya lakukan proses XOR dengan 00001111. Hasilnya adalah "00101000 11110001 11000000 00100111 10111001 00001010 01101111 01010010 01001000 10110000"

Langkah 4. Kemudian lakukannya proses dekripsi *Huffman* dengan menggunakan pohon Huffman hasil enkripsi sebagai kunci.

Langkah 5. Selanjutnya setiap karakter dikonversikan sesuai kode ASCII dan hasilnya menjadi adalah **Kodenya adalah 1#_&0**.

3.4. Pembahasan

Berikut ini pembahasan dari penelitian yang dilakukan:

- Modifikasi dilakukan baik pada proses *enkripsi Huffman* maupun pada enkripsi *hill cipher*. Pada algoritma *Huffman* terdapat penambahan operasi XOR dan Teorema *Euclidean* agar hasil yang didapatkan sesuai ASCII *printable characters*. Sedangkan pada proses enkripsi *hill cipher* mengalami perubahan modulo dari 26 menjadi 95. Penambahan dan pengurangan 32 dilakukan agar karakter yang muncul dalam rentang $32 - 126$ yang sesuai dengan ASCII *printable characters*. Pada proses dekripsi juga dapat mengembalikan *ciphertext* ke dalam bentuk *plaintext*.
- Algoritma *Huffman* tidak menggunakan kunci masukan dalam pengkodeannya dan menghasilkan *ciphertext* dengan panjang kurang dari *plaintext*. Kunci *Huffman* diperoleh dari pohon *Huffman* tersebut. Perbedaan panjang tersebut memungkinkan pihak ketiga sulit dalam menebak *plaintext* tersebut. Berbeda dengan *hill cipher* yang menggunakan kunci matriks, tetapi panjang *plaintext* dengan ciphertext-nya sama. Selain *ciphertext*, kunci matriks tersebut juga akan dikirimkan ke penerima. Sehingga apabila kunci tersebut diketahui pihak ketiga, pesan dapat langsung terpecahkan. Sedangkan modifikasi *Huffman* dengan *hill cipher* memerlukan kunci matriks dengan menghasilkan panjang *ciphertext* kurang dari *plaintext*. Meskipun kunci matriks telah diketahui.

4. Simpulan

Pada modifikasi algoritma *Huffman* dengan *hill cipher* menggunakan kunci matriks $n \times n$. Kunci tersebut harus *invertible* yaitu $K.K^{-1} = I$. Kunci K akan digunakan dalam proses enkripsi, sedangkan K^{-1} digunakan dalam proses pesan tetap sulit dipecahkan karena perbedaan panjang tersebut.

Pada proses algoritma *Huffman* terdapat penambahan operasi XOR dan Teorema *Euclidean* agar hasil yang didapatkan sesuai ASCII *printable characters*. Sedangkan pada proses proses *hill cipher* mengalami perubahan modulo dari 26 menjadi 95. Penambahan dan pengurangan 32 dilakukan agar karakter yang muncul dalam rentang $32 - 126$ yang sesuai dengan ASCII *printable characters*. *Ciphertext* yang dihasilkan memiliki panjang kurang dari *plaintext*-nya.

Pengkodean menggunakan *Huffman* dengan *hill cipher* lebih aman dibandingkan algoritma *Huffman* maupun *hill cipher* sebelum dimodifikasi. Karena pada pengkodean *Huffman* tidak menggunakan kunci tambahan. Sedangkan pada *hill cipher* panjang *plaintext* dengan *ciphertext*-nya sama. Apabila kuncinya diketahui, maka pesan akan mudah dipecahkan. Tetapi jika kunci pada modifikasi *Huffman* dengan *hill cipher* diketahui, pesan tetap sulit dipecahkan karena perbedaan panjang *plaintext* dengan *ciphertext*-nya.

Daftar Pustaka

Auliyah, A. I. 2020. Implementasi Kombinasi Algoritma Enkripsi Rivest Shamir Adleman (RSA) dan Algoritma Kompresi Huffman pada File Document. *Indonesian Journal of Data and Science*, *1*(1), 23-28.

Azis, N. 2018. Perancangan Aplikasi Enkripsi Dekripsi Menggunakan Metode Caesar Chiper dan operasi XOR. *IKRAITH Informatika*, *2*(1), 72-80.

Hasugian, A. H. 2013. Implementasi Algoritma Hill Cipher dalam Penyandian Data. *Jurnal Pelita Informatika Budi Darma 4*(2), 115-122.

Pahdi, A. 2017. Algoritma Huffman dalam Pemampatan dan Enkripsi Data. *Indonesian Journal on Networking and Security*, *6*(3), 1-7.

Pardede, A. M. H., Hamunurung, H., & Filina, D 2017. Algoritma Vigenere Cipher Dan Hill Cipher Dalam Aplikasi Keamanan Data Pada File Dokumen. *Jurnal Teknik Informatika Kaputama*, *1*(1), 26-33.

Rahmawati, R. 2017. Penggabungan Vigenere Cipher dengan Hill Cipher pada Pengkodean Plaintext dengan Kunci Bertahap. *Skripsi*. Jember: Universitas Jember.

Setyaningsih, E. 2015. *Kriptografi dan Implementasinya Menggunakan Matlab*. Yogyakarta: Andi Offset.

Wardani, R. D., & Kurniawan, S. M. 2019. Penerapan Teori Bilangan untuk Menentukan Kongruensi pada Lampu Lalu Lintas. *Barekeng: Jurnal Ilmu Matematika dan Terapan*, *13*(1), 47-52.