Implementasi Algoritma Kunci Matriks untuk Keamanan Data Akademik

(*Implementation of Key Matrix Algorithms for Academic Data Security*)

Joko Soebagyo[1], Imay Kurniawan[2]

[1,2]*Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Wastukancana Purwakarta*
Jl. Cikopak 53, Sadang, Purwakarta, Jawa Barat

[1]joko@stt-wastukancana.ac.id
[2]imaykurniawan2013@gmail.com

Abstrak— Tujuan penelitian ini adalah merancang dan membangun perangkat lunak kriptografi menggunakan algoritma kunci matriks berordo 3×3 modifikasi dari algoritma Hill Chiper untuk keamanan data akademik. Enkripsi data akademik menggunakan algoritma kunci matriks ini diperlukan sebagai upaya untuk mengamankan data akademik dari pihak-pihak yang tidak bertanggung jawab. Gagasan dalam memodifikasi algoritma Hill Chiper terkait kunci matriks menjadi tantangan tersendiri bagi para peneliti, seperti modifikasi Hill Cipher berbasis Matriks Sirkulan dan kombinasi Hill Cipher dengan Kurva Elips yang menggunakan kunci matriks secara berturut-turut berukuran 2×2 dan 4×4. Namun terdapat penelitian algoritma Hill Cipher menggunakan algoritma genetik, Hill Cipher Paralel dan Hill Cipher berganda 3 dengan kunci matriks berukuran 3×3. Hasil dari lima penelitian terkait, secara umum menyatakan bahwa implementasi modifikasi Hill Cipher yang digunakan efektif dan efisien. Metode yang digunakan adalah metode pengembangan perangkat lunak menggunakan paradigma pengembangan perangkat lunak Waterfall. Hasil dari penelitian adalah enkripsi dan dekripsi data akademik yang meliputi data mahasiswa, mata kuliah, KRS, dan nilai. Berdasarkan hasil pengujian, maka dapat disimpulkan bahwa implementasi teknik kriptografi menggunakan algoritma kunci matriks menggunakan bahasa pemrograman PHP dan database MySql, untuk pengamanan data akademik berhasil dilakukan.

Kata-kata kunci— **Kriptografi, Algoritma Kunci Matriks, Enkripsi, Dekripsi**

Abstract— The purpose of this study is to design and build cryptographic software using a 3 × 3 modified matrix key algorithm from the Hill Chiper algorithm for academic data security. Encryption of academic data using this matrix key algorithm is needed as an effort to secure academic data from irresponsible parties. The idea of modifying the Hill Chiper algorithm regarding matrix keys is a challenge for researchers, such as the modification of the Circulatory Matrix-based Hill Cipher and the combination of the Hill Cipher with the Ellipse Curve that uses 2 × 2 and 4 × 4 matrix keys, respectively. However, there is research on the Hill Cipher algorithm using genetic algorithms, Parallel Hill Cipher and multiple Hill Cipher with a 3 × 3 matrix key. The results of five related studies, generally states that the implementation of the Hill Cipher modification used is effective and efficient. The method used is a software development method using the Waterfall software development paradigm. The results of the study are the encryption and decryption of academic data which includes student data, courses, KRS, and grades. Based on the test results, it can be concluded that the implementation of cryptographic techniques using a key matrix algorithm using the PHP programming language and MySql database, for safeguarding academic data successfully.

Keywords *— **Cryptography, Key Matrix Algorithms, Encryption, Decryption***

I. PENDAHULUAN

Kriptografi adalah studi tentang "matematika" untuk memecahkan dua jenis masalah keamanan: privasi dan otentikasi [1], di mana salah satu tekniknya adalah menggunakan algoritma Hill Chiper menggunakan matrik invertibel [2]. Berdasarkan definisi, algoritma kriptografi kunci matriks termasuk dalam enkripsi asimetris[2] dimana terdapat dua kunci yang digunakan, yaitu: kunci *private* dan publik [3]. Dalam penelitian ini, algoritma kunci matriks didasarkan pada transformasi *plaintext* dan *ciphertext* ke dalam matriks dengan algoritma Hill Cipher, dimana:

$$C = KP \bmod m \text{ dan } P = K^{-1}C$$

dengan K adalah kunci matriks invertibel, P adalah matriks *plaintext* dan C adalah matriks *ciphertext* [4], [5].

Gagasan dalam memodifikasi Algoritma Hill Cipher terkait matriks kunci menjadi tantangan tersendiri bagi para peneliti, seperti modifikasi Hill Cipher berbasis Matriks Sirkulan [6] dan kombinasi Hill Cipher dengan Kurva Elips [7] yang menggunakan kunci matriks secara berturut-turut berukuran 2×2 dan 4×4. Namun terdapat penelitian algoritma Hill Cipher menggunakan algoritma genetik [8], Hill Cipher Paralel [9] dan Hill Cipher berganda 3 [10] dengan kunci matriks berukuran 3×3. Hasil dari lima penelitian terkait, secara umum menyatakan bahwa implementasi modifikasi Hill Cipher yang digunakan efektif dan efisien.

A. Algoritma Kunci Matriks

Dalam penelitian ini, implementasi rancang bangun algoritma kunci matriks yaitu algoritma Hill Cipher, termodifikasi dalam empat bagian, yang pertama, kunci matriks menggunakan koefesien-koefesien binomial**,** karena nilai determinan matriks sama dengan satu [11], [12]. Kedua, konversi *plaintext* menggunakan kode ASCII atau modulo 256, dan ketiga, ukuran dari matriks *ciphertext* $[c_{ij}]$ dengan j ditentukan oleh panjang karakter *plaintext* m, dimana terdapat dua kondisi:

$$j = \begin{cases} \dfrac{m}{i} & \text{, jika } m \bmod i = 0 \\ \dfrac{m+n}{i} & \text{, jika } m \bmod i \neq 0 \end{cases}$$

Dengan $m, n \in N$. Modifikasi keempat dari algoritma kunci matriks pada penelitian ini adalah: $\forall c_k \in [c_{ij}], \exists p_k 0$. Enkripsi data akademik menggunakan algoritma kunci matriks ini diperlukan sebagai upaya untuk mengamankan data akademik dari pihak-pihak yang tidak bertanggung jawab.

B. Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak *(RPL)* adalah istilah lain dari *Software Engineering*, yang diajukan pada *Software Engineering Conference* yang disponsori oleh NATO tahun 1968. Terdapat perbedaan yang mendasar antara RPL dan *software* atau perangkat lunak (PL), di mana RPL merupakan disiplin teknik yang berkaitan dengan semua aspek produksi perangkat lunak, sedangkan PL berkaitan dengan program komputer dan dokumentasi. [13]

C. Proses Rekayasa Perangkat Lunak

Proses rekayasa perangkat lunak yang digunakan dalam penelitian ini terdiri dari 4 aktifitas, yaitu: 1) *Software specification* berhubungan dengan spesifikasi perangkat lunak yang ingin dihasilkan, 2) *Software development* mencakup desain dan *pemrograman*, 3) *Software validation* berhubungan dengan pengecekan untuk memastikan kesesuaian dengan permintaan, dan 4) *Software evolution* berhubungan dengan modifikasi perangkat lunak [13].

D. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa standar untuk membuat rancangan suatu perangkat lunak berbentuk pemodelan mulai dari sistem informasi *enterprise* hingga mendistribusikan aplikasi berbasis *web* dan bahkan ke sistem tertanam waktu nyata.[14]. Sistem pemodelan atau perancangan dengan UML, dapat terhubung langsung ke berbagai bahasa pemrograman., seperti Java, C ++, atau Visual Basic, atau bahkan ke tabel dalam *database* relasional atau penyimpanan terus-menerus dari *database* berorientasi objek [14]. Dengan kata lain, model UML dapat diterapkan pada berbagai jenis aplikasi perangkat lunak.

1) Use Case Diagram. Istilah "use case" merujuk pada urutan lengkap peristiwa dalam sistem sebagaimana dipahami dari perspektif pengguna [15]. Dengan kata lain, use case merupakan himpunan urutan persitiwa di mana setiap urutan merepresentasikan interaksi dari objek di luar sistem (aktor-aktornya) dengan sistem itu sendiri (dan abstraksi kuncinya) [14].

*2) Class Diagram***.** Class diagram merupakan representasi struktur statis dari suatu sistem yang menunjukkan sistem class, atribut dan operasinya, serta relasi antar class [16] dan sangat relevan dengan bahasa pemrograman PHP [17]. Sementara, class adalah jenis classifier terpenting pada UML, di mana ia mendeskripsikan sekumpulan objek yang memiliki atribut, operasi, hubungan, dan semantik yang sama yang mengimplementasikan satu atau lebih antarmuka (interface) [13]. Class memilliki tiga area pokok, yaitu: 1) Nama (dan stereotype), 2) Atribut, dan 3) Metode [13], [16]. Atribut dan metode dapat memiliki salah satu sifat, yaitu: a) Private, hanya digunakan oleh classifier itu sendiri, disimbolkan dengan "−", b) Protected, hanya digunakan oleh class dan turunannya, disimbolkan dengan "#", dan c) Public, digunakan oleh sebarang classifier luar dengan visibilitas pada classifier yang diberikan, disimbolkan dengan "+"[14].

3) PHP. PHP merupakan bahasa script yang ditempatkan dan diproses pada server lalu hasilnya dikirim ke client, dimana client menerima atau melihat hasil yang dikirimkan oleh server dengan menggunakan web browser [18]. Sebagian besar web yang ada di internet dibangun dengan menggunakan bahasa pemograman PHP. Beberapa alasan penggunaan PHP adalah: a) PHP merupakan bahasa pemograman Open Source dan dikembangkan oleh komunitas tersebut sehingga bisa didapatkan dengan mudah dan digunakan tanpa harus mengeluarkan biaya, b) PHP dapat digunakan pada sistem operasi seperti Linux, Microsoft Windows, Solaris, Mac OS X, Open BSD, dan RISK OS, c) PHP didukung oleh beberapa web server seperti Apache, Personal Web Server, dan Internet Information Server, d) Dalam penggunaannya PHP mendukung beberapa database seperti Interbase, PostgreSQL, Sybase, Mysql, FrontBase, SQLite, Informix, Oracle, dan ODBC, dan e) PHP juga memberikan kemudahan dalam menampilkan berbagai macam teks, gambar dan file PDF.

4) MySQL. MySQL adalah sistem relasional database yang paling banyak digunakan di sektor Open Source dengan kelebihan: cepat, stabil, mudah dipelajari, kompatibel dengan OS populer dan didukung oleh berbagai macam bahasa pemrograman [19]. Selain itu, bahasa pemograman PHP juga sangat mendukung konektivitas database MySQL [20],[18].

5) Black-Box Testing. Black-box testing (BBT) adalah suatu pendekatan pengujian fungsional atau behaviour di mana konten dari black-box tidak diketahui dan fungsi black-box dipahami sepenuhnya dalam hal input dan outputnya [21]. BBT mengidentifikasi jenis-jenis kesalahan sebagai berikut [21]: 1) Fungsi yang tidak benar atau hilang, 2) Interface yang keliru atau hilang, 3) Kesalahan dalam model data, dan 4) Kesalahan dalam akses ke sumber data eksternal.

II. METODE

Metode penelitian ini menggunakan model Waterfall sampai tahap keempat, yaitu:1) *Analysis*, 2) *Design*, 3) *Coding*, dan 4) *Testing*.

A. Tahapan Enkripsi Algoritma Kunci Matriks $\mathbf{3 \times 3}$

Tahapan enkripsi menggunakan algoritma kunci Matriks 3×3 terdiri dari tujuh tahap:

1) Hitung m panjang karakter *plaintext*.

2) Hitung sisa hasil bagi (s) panjang karakter dengan angka 3. Jika $s \neq 0$, tambah *plaintext* dengan sejumlah karakter *null* sehingga $s = 0$.

3) Konversi semua karakter data *plaintext* menjadi angka.

4) Data *plaintext* yang sudah dikonversi menjadi angka ditransformasi ke matriks $3 \times n$, dimana n adalah jumlah kolom dari matriks dengan $n =$ panjang karakter *plaintext* dibagi angka 3.

5) Hitung matriks C perkalian matriks kunci enkripsi (matriks A) dengan matriks *plaintext* yang sudah ditranformasi ke bentuk matriks $3 \times n$ dengan kunci enkripsi matriks 3×3 (matriks B).

6) Tambah setiap isi sel matriks C dengan karakter *null*.

7) Gabungkan isi sel matriks yang sudah ditambah dengan karakter *null*.

B. Tahapan Dekripsi Algoritma Kunci Matriks $\mathbf{3 \times 3}$

Tahapan dekripsi menggunakan algoritma kunci matriks 3×3 terdiri dari enam tahap, yaitu:

1) Ambil semua karakter angka sebelum karakter null dari *ciphertext*, kemudian simpan di $C_1, C_2, C_3, \ldots, C_m$.

2) Hitung $n = \frac{m}{3}$.

3) Transformasi $C_1, C_2, C_3, \ldots, C_m$ ke dalam matriks C $3 \times n$.

4) Hitung matriks P, perkalian matriks kunci dekripsi (matriks B) dengan matriks *ciphertext* yang sudah ditranformasi ke bentuk matriks $3 \times n$ (matriks C).

5) Konversi masing-masing angka isi sel matriks P menjadi karakter.

6) Gabungkan semua isi sel matriks P yang isinya bukan karakter null.

C. Analisis Kebutuhan Non Fungsional

1) Analisis Kebutuhan Pengguna (*User*): Pengguna (*user*) perangkat lunak algoritma kriptografi kunci matriks adalah administrator.

2) Analisis Kebutuhan Perangkat Keras (*Hardware*): Spesifikasi perangkat keras yang dibutuhkan sebagai berikut :*Prosesor Intel atom 1,8GHz; Chipset Intel; Grafis Intel HD Graphics 5500;*

Memori RAM 2GB; Storage hard disk 300GB 5400 rpm.

3) Analisis Kebutuhan Perangkat Lunak (*Software*): Spesifikasi perangkat lunak yang dibutuhkan sebagai berikut :a) Untuk membangun sistem, digunakan: Sistem operasi Microsoft *Windows 7 Enterprise 32bit*; *Microsoft office2010; easywamp1.1; npp.5.9;* Mozilla FireFox 43.0.4; Bahasa pemrograman PHP; *Database* MySQL; Bahasa pemrograman JavaScript, dan b) Untuk mengakses sistem digunakan: Sistem operasi Microsoft Windows; *Software browser* Mozilla FireFox dan *Google Chrome.*

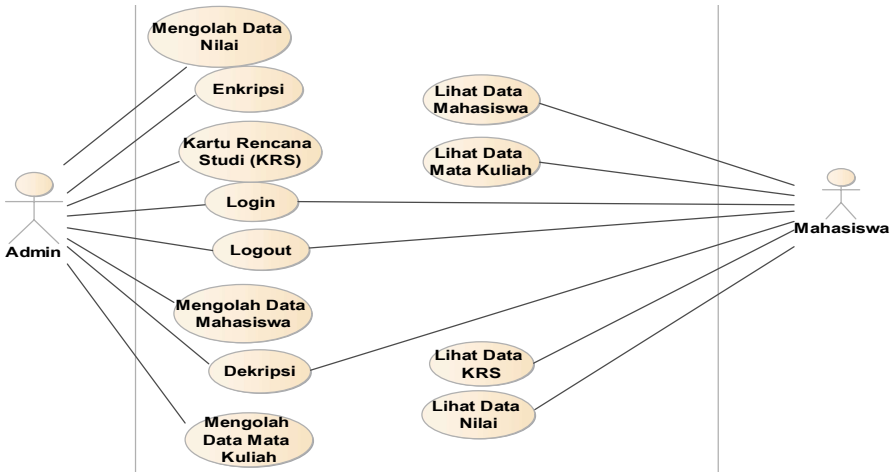
D. Analisis Kebutuhan Fungsional

Terdapat delapan kebutuhan fungsional yang akan dibangun oleh sistem, yaitu proses enkripsi dan dekripsi data: mahasiswa, mata kuliah, KRS, dan nilai.

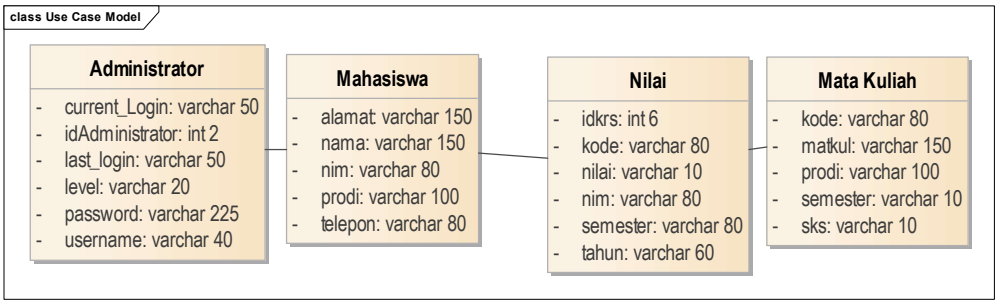
E. Perancangan Sistem

1) Perancangan UML (*Unified Modelling Language*)**.** *Use Case Diagram* yang dirancang untuk menggambarkan perilaku (*behavior*) sistem pada penelitian ini dapat dilihat pada Gambar 1. *Class diagram* yang dirancang untuk menunjukkan nama, operasi dan atribut dalam sistem dapat dilihat pada Gambar 2.

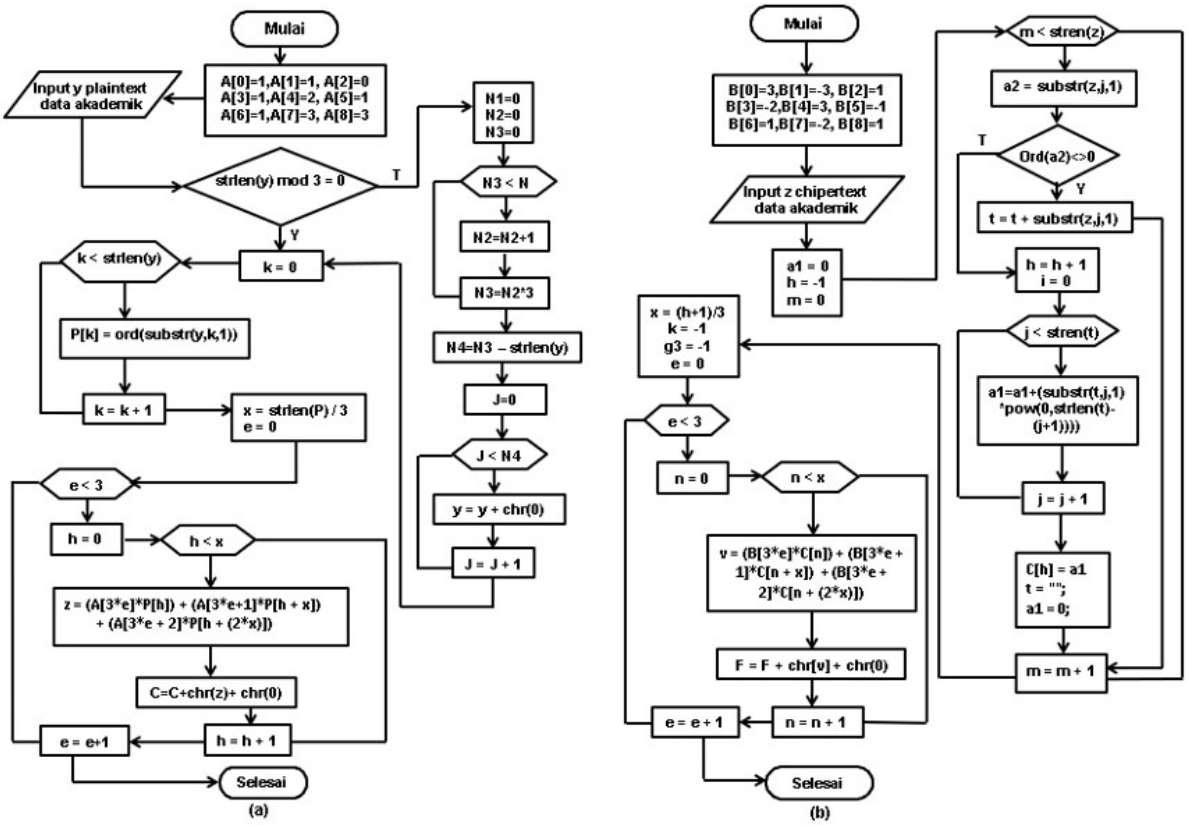
*2) Flowchart Enkripsi dan Dekripsi Algoritma Kunci Matriks***.** Proses enkripsi adalah mengubah data akademik yang asli menjadi data akademik yang sudah disandikan. Untuk bagan alir enkripsi dapat dilihat di Gambar 3(a). Proses dekripsi adalah mengubah data akademik yang sudah disandikan menjadi data akademik yang asli. Untuk bagan alir enkripsi dapat dilihat di Gambar 3(b).



Gambar 1. Use Case Diagram Sistem



Gambar 2. Class Diagram Sistem



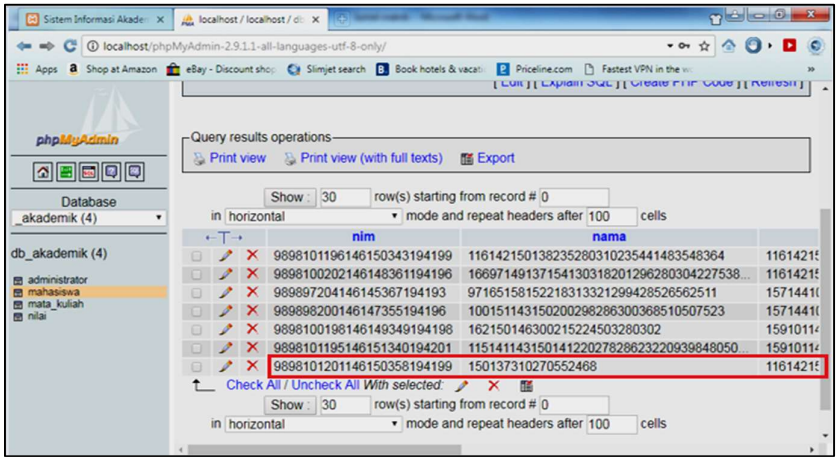
Gambar 3. Flowchart (a) Enkripsi, (b) Dekripsi Algoritma Kunci Matriks

III. HASIL DAN PEMBAHASAN

A. Tampilan Antarmuka

1) Halaman Enkripsi Data Mahasiswa. Proses enkripsi data mahasiswa dilakukan setelah admin memasukkan data, sebagai contoh data mahasiswa, NIM: 1241016, Nama: HENDRA, dan hasil enkrispsinya seperti pada Gambar 4 dalam kotak merah.

2) Halaman Dekripsi Data Mahasiswa. Untuk menguji apakah dekripsi data mahasiswa berhasil dilakukan, hasilnya dapat dilihat pada Gambar 5.



Gambar 4. Hasil Enkripsi Data Mahasiswa

3) Halaman Enkripsi Data Mata Kuliah. Proses enkripsi data mata kuliah dilakukan setelah admin memasukkan data, sebagai contoh data mata kuliah yaitu, Kode Mata Kuliah: IF.127, Nama Mata Kuliah: ALGORITMA PEMROGRAMAN II, SKS: 3, dan Semester: 2. Dengan algoritma kunci matriks akan diperoleh enksripnya, 119119215223361382, 13010815114815915516314827720530828231431926929250636754248154741554050 9, 515151, dan 505050, seperti pada Gambar 6.

4) Halaman Dekripsi Data Mata Kuliah. Untuk menguji apakah dekripsi data mata kuliah berhasil dilakukan, dapat dilihat pada Gambar 7.

5) Halaman Enkripsi Data KRS. Proses enkripsi data KRS dilakukan setelah mahasiswa memasukkan data, sebagai contoh data KRS, NIM: 1241011, Kode Mata Kuliah: IF.146, Nilai: (masih kosong), dan Tahun: 2015. Dengan algoritma kunci matriks akan diperoleh enksripsinya, 9898100202146148361194196, 119119217222367379, (nilai masih kosong), dan 99101148154197207 seperti pada Gambar 8.

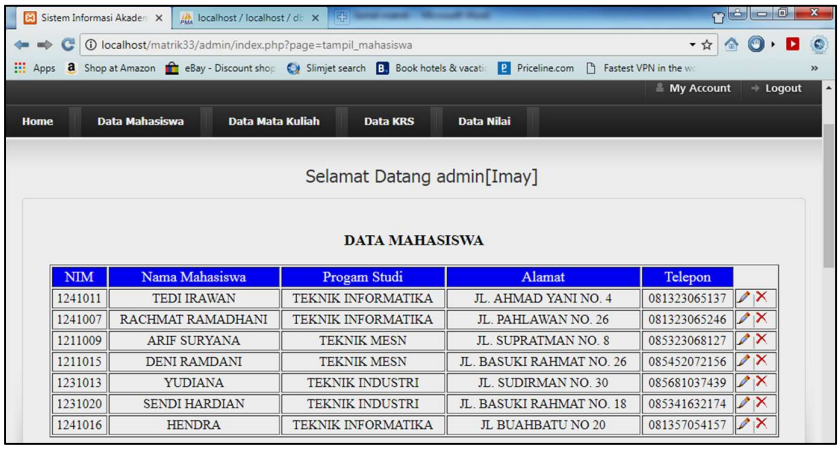
6) Halaman Dekripsi Data KRS. Hasil pengujian dekripsi data KRS, menggunakan kunci matriks diperoleh, NIM: 1241011, Kode Mata Kuliah: IF.146, Nilai: (masih kosong), dan Tahun: 2015.

7) Halaman Enkripsi Data Nilai. Proses enkripsi data nilai dilakukan setelah admin memasukkan data, sebagai contoh data nilai, NIM: 1241011, Kode Mata Kuliah: IF.127, Nilai: C, dan Tahun: 2015. Dengan algoritma kunci matriks akan diperoleh enksripsinya, 98981011961461503431941 99, 119119215223361382, 676767, dan 99101148154197207 seperti pada Gambar 9.

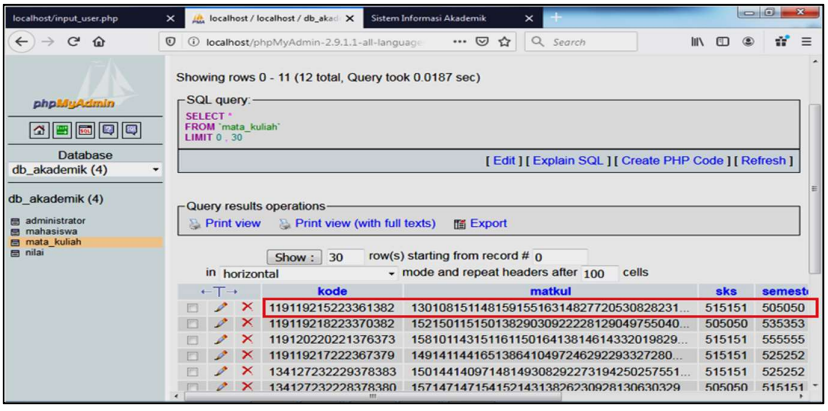
8) Halaman Dekripsi Data Nilai. Hasil pengujian dekripsi data nilai, menggunakan kunci matriks diperoleh, NIM: 1241011, Kode Mata Kuliah: IF.127, Nilai: C, dan Tahun: 2015.

B. Pengujian

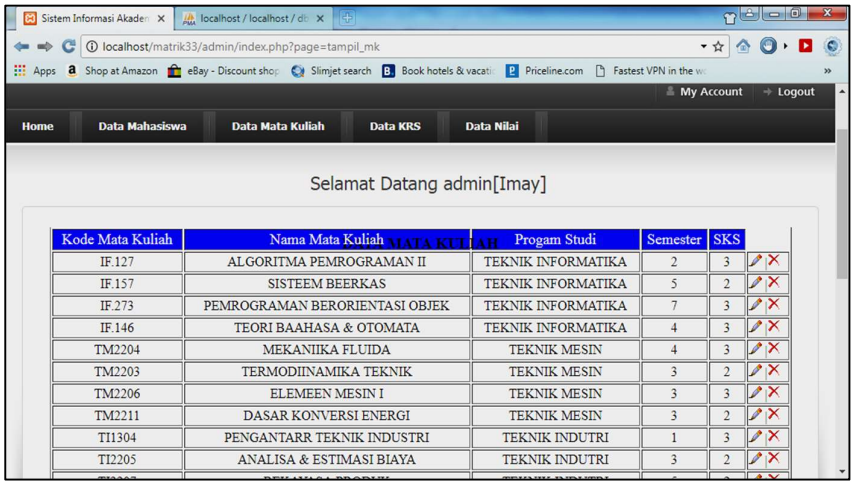
Pengujian yang dilakukan menggunakan teknik pengujian *black box* yang memfokuskan pada domain fungsional dari perangkat lunak. Hasil pengujian dapat dilihat pada Tabel 1.



Gambar 5. Halaman Dekripsi Data Mahasiswa

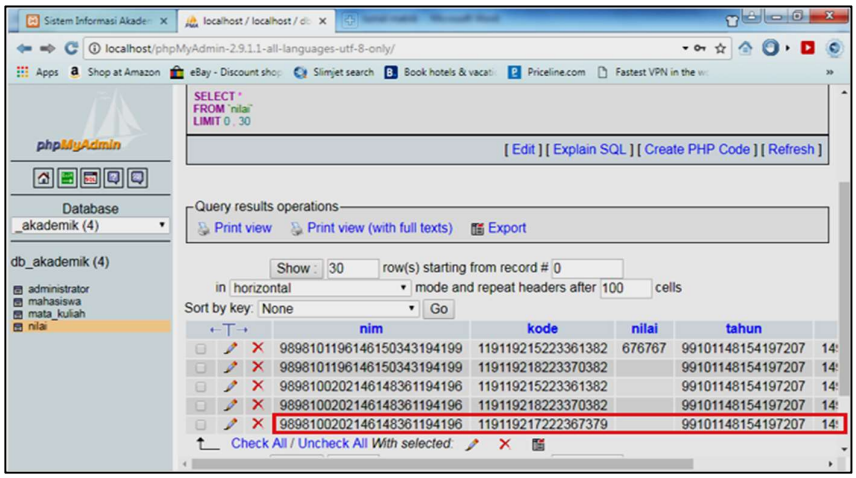


Gambar 6. Hasil Enkripsi Data Mata Kuliah

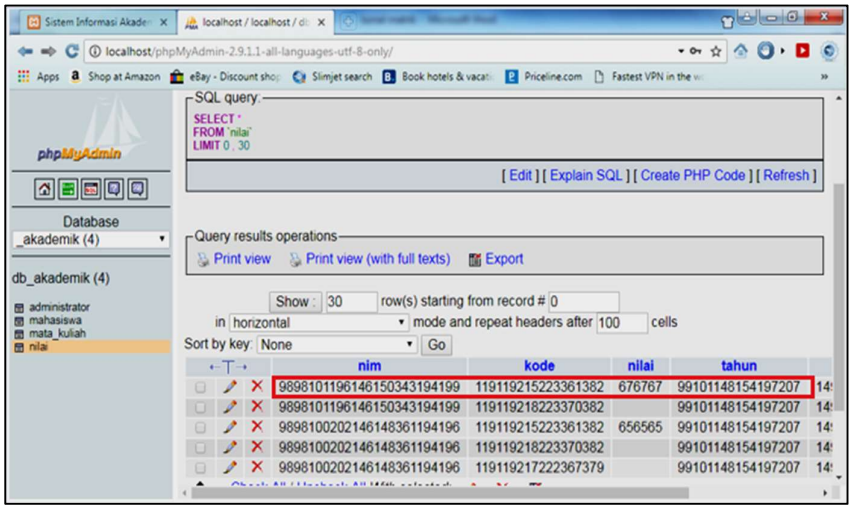


Kode Mata Kuliah	Nama Mata Kuliah	Progam Studi	Semester	SKS
IF.127	ALGORITMA PEMROGRAMAN II	TEKNIK INFORMATIKA	2	3
IF.157	SISTEEM BEERKAS	TEKNIK INFORMATIKA	5	2
IF.273	PEMROGRAMAN BERORIENTASI OBJEK	TEKNIK INFORMATIKA	7	3
IF.146	TEORI BAAHASA & OTOMATA	TEKNIK INFORMATIKA	4	3
TM2204	MEKANIIKA FLUIDA	TEKNIK MESIN	4	3
TM2203	TERMODIINAMIKA TEKNIK	TEKNIK MESIN	3	2
TM2206	ELEMEEN MESIN I	TEKNIK MESIN	3	3
TM2211	DASAR KONVERSI ENERGI	TEKNIK MESIN	3	2
TI1304	PENGANTARR TEKNIK INDUSTRI	TEKNIK INDUTRI	1	3
TI2205	ANALISA & ESTIMASI BIAYA	TEKNIK INDUTRI	3	2

Gambar 7. Halaman Dekripsi Data Mata Kuliah



Gambar 8. Hasil Enkripsi Data KRS



Gambar 9. Halaman Hasil Enkripsi Data Nilai

TABEL I
HASIL PENGUJIAN

No	Pengujian	Hasil Yang Diharapkan	Keterangan
1	Enkripsi data mahasiswa	Sistem menyimpan input data mahasiswa dan hasilnya berbentuk data mahasiswa yang sudah disandikan dan disimpan di simpan *database* db_akademik	Berhasil
2.	Dekripsi data mahasiswa	Sistem mampu mengembalikan data mahasiswa yang sudah disandikan menjadi data mahasiswa yang asli	Berhasil
3	Enkripsi data mata kuliah	Sistem menyimpan input data mata kuliah dan hasilnya berbentuk data mahasiswa yang sudah disandikan dan disimpan di *database* db_akademik	Berhasil
4.	Dekripsi data mata kuliah	Sistem mampu mengembalikan data mata kuliah yang sudah disandikan menjadi data mata kuliah yang asli	Berhasil
5	Enkripsi data KRS	Sistem menyimpan input data KRS dan hasilnya berbentuk data KRS yang sudah disandikan dan disimpan di *database* db_akademik	Berhasil
6.	Dekripsi data KRS	Sistem mampu mengembalikan data KRS yang sudah disandikan menjadi data KRS yang asli	Berhasil
7	Enkripsi data nilai	Sistem menyimpan input data nilai dan hasilnya berbentuk data nilai yang sudah disandikan dan disimpan di simpan *database* db_akademik	Berhasil
8.	Dekripsi data nilai	Sistem mampu mengembalikan data nilai yang sudah disandikan menjadi data nilai yang asli	Berhasil

IV. PENUTUP

A. Kesimpulan

Setelah melakukan tahap analisis, perancangan, dan tahap implementasi terhadap pengamanan data akademik dengan menggunakan algoritma kunci matriks, diperoleh kesimpulan sebagai berikut:

1) Algoritma kunci matriks berordo 3×3 dapat diimplementasikan untuk keamanan data akademik, dan

2) Kunci enkripsi matriks berordo 3×3 harus memiiki invers matriks berordo 3×3.

B. Saran

Berdasarkan kesimpulan, penelitian ini masih jauh dari kata sempurna, oleh karenanya diperlukan masukan dan saran sehingga sistem ini lebih baik lagi di masa yang akan datang, antara lain:

1) Karena sistem dengan algoritma kunci matriks berordo 3×3 beserta kuncinya telah ditetapkan oleh sistem, maka untuk pengembangan selanjutnya, proses mengenkripsi dan dekripsi dapat dilakukan oleh *user* administrator yang menginputkan matriks serta dapat menggunakan matriks berordo $4 \times 4, 5 \times 5$ dan seterusnya.

2) Sistem ini menyimpan hasil enkripsi (*ciphertext*) di *database* akademik berbentuk karakter angka sehingga membutuhkan penyimpanan *record database* yang besar, diharapkan ke depannya dapat disimpan dalam bentuk karakter huruf sehingga penyimpanan *record database* menjadi lebih kecil.

3) Untuk penelitian selanjutnya diharapkan algoritma kunci matriks dapat dikombinasi dengan algoritma yang lain sehingga tingkat keamanan data akademik menjadi lebih sulit dipecahkan. Selain itu, perlu dilakukan komparasi efektifitas dan efisiensi antara penggunaan algoritma kunci matriks dan modifikasi Algoritma Hill Cipher lainnya.

UCAPAN TERIMA KASIH

Peneliti bersyukur kepada Allah SWT dan berterima kasih kepada semua pihak yang telah mendukung penelitian ini. Proyek ini didanai oleh Kemenristekdikti dengan nomor kontrak penelitian 107/PP/STT-WKN/PWK/III/2019.

DAFTAR PUSTAKA

[1] W. Diffie and M. E. Hellman, "New Directions in Cryptography Invited Paper," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[2] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press Inc., 1996.

[3] D. Salama, A. Minaam, H. M. Abdual-kader, and M. M. Hadhoud, "Evaluating the Effects of Symmetric Cryptography Algorithms on Power Consumption for Different Data Types," *Int. J.*, vol. 11, no. 2, pp. 91–100, 2010.

[4] L. S. Hill, "Cryptography in an algebraic alphabet," *Am.*

Math. Mon., vol. 36, no. 6, pp. 306–312, 1929.

[5] L. S. . Hill, "Concerning Certain Linear Transformation Apparatus of Cryptography," *Am. Math. Mon.*, vol. 38, no. 3, pp. 135–154, 1931.

[6] K. A. Reddy, B. Vishnuvardhan, and A. V. N. Krishna, "A Modified Hill Cipher Based on Circulant Matrices," *Procedia Technol.*, vol. 4, pp. 114–118, 2012.

[7] Z. E. Dawahdeh, S. N. Yaakob, and R. Razif, "A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 3, pp. 349–355, 2018.

[8] A. Putera, U. Siahaan, and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *Int. J. Secur. Its Appl.*, vol. 10, no. 8, pp. 173–180, 2016.

[9] M. H. Qasem, "Parallel Hill Cipher Encryption Algorithm," *Int. J. Comput. Appl.*, vol. 179, no. 19, pp. 16–24, 2018.

[10] A. A. M. Khalaf, M. S. A. El-karim, and H. F. A. Hamed, "A Triple Hill Cipher Algorithm Proposed to Increase the Security of Encrypted Binary Dataand its Implementation Using FPGA," *Trans. Adv. Commun. Technol.*, vol. 5, no. 1, pp. 752–759, 2016.

[11] A. Edelman and G. Strang, "Pascal matrices," *Am. Math. Mon.*, vol. 111, no. 3, pp. 189–197, 2004.

[12] H. Anton and C. Rorres, *Elementary Linear Algebra*, Ninth edit. John Wiley & Sons, Inc., 2010.

[13] I. Sommerville, *Software Engineering*, Ninth Edit. Boston: Pearson Education, Inc., 2011.

[14] G. Booch, J. Rumbaugh, and I. Jacobson, *The unified modeling language*, vol. 14, no. 13. Massachusetts: Addison Wesley, 1999.

[15] A. Gemino and D. Parker, "Use case diagrams in support of use case modeling: Deriving understanding from the picture," *J. Database Manag.*, vol. 20, no. 1, pp. 1–24, 2009.

[16] A. Cal, G. Gottlob, G. Orsi, and A. Pieris, "Querying UML Class Diagrams," in *International Conference on Foundations of Software Science and Computational Structures*, 2012, no. March, pp. 1–25.

[17] J. E. Sweat, *PHP Architect's Guide to PHP Design Patterns - Page 2*. 2005.

[18] L. Welling and L. Thomson, *PHP and MySQL Web Development*, Fifth Edit. Pearson Education, Inc., 2017.

[19] M. Kofler, *The Definitive Guide to MySQL 5*, 3rd ed. Berkeley: Apress, 2005.

[20] J. Castagnetto, H. Rawat, S. Schumann, C. Scollo, and D. Veliath, *Professional PHP Programming*. 1999.

[21] B. Agarwal, B., P. Tayal, S., and M. Gupta, *Software Engineering & Testing An Introduction*. Boston: Jones and Bartlett Publishers, LLC., 2010.