pISSN : 2337-3601
eISSN : 2549-015X

JURNAL TIMES

(Technology Informatics & Computer System)

Vol. XI No. 2 Tahun 2022

061 4561932 KAMPUS MEDAN
4575843 KAMPUS MULTATULI
www.stmik-time.ac.id
Official Web STMIKTIME
sosmed@stmik-time.ac.id
Official Facebook STMIKTIME
@stmiktime
Official Instagram STMIKTIME

Dewan Editor

Editor

Edi Wijaya, Indonesia (ediwijaya@stmik-time.ac.id)

Editor Bagian

Hendri Hendri, Indonesia (hendri@stmik-time.ac.id)

Eninta Ibrena, Indonesia (eninta@stmik-time.ac.id)

Jurnal TIMES Terindeks :

Template Jurnal TIMES

Template Jurnal TIMES (https://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/libraryFiles/downloadPublic/1)

Informasi

Untuk Pembaca (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/information/readers)

Untuk Penulis (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/information/authors)

Untuk Pustakawan (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/information/librarians)

Bahasa

Bahasa Indonesia (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/user/setLocale/id_ID?source=%2Findex.php%2FjurnalTIMES%2Fabout%2FeditorialTeam)

English (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/user/setLocale/en_US?source=%2Findex.php%2FjurnalTIMES%2Fabout%2FeditorialTeam)

JTM : Jurnal TIMES [ISSN Print 2337-3601 (http://issn.pdii.lipi.go.id/issn.cgi?daftar&1395053872&1&&)] [ISSN Online 2549-015X (http://issn.pdii.lipi.go.id/issn.cgi?daftar&1484036122&1&&)]

Dipublikasikan oleh

Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) STMIK TIME

Jalan Merbabu No. 32 AA-BB, Medan Kota, Kota Medan, Sumatera Utara.

Kode Pos: 20212 | **Telp :** 061-4561932 | **Contact :** ejournal@stmik-time.ac.id

Diterbitkan: 2023-02-02

Articles

DISTANCE (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/article/view/681)

Leony Hoki

40-48

PDF (HTTP://EJOURNAL.STMIK-TIME.AC.ID/INDEX.PHP/JURNALTIMES/ARTICLE/VIEW/681/252)

Jurnal TIMES Terindeks :

Template Jurnal TIMES

Template Jurnal TIMES (https://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/libraryFiles/downloadPublic/1)

Informasi

Untuk Pembaca (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/information/readers)

Untuk Penulis (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/information/authors)

Untuk Pustakawan (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/information/librarians)

Bahasa

Bahasa Indonesia (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/user/setLocale/id_ID?source=%2Findex.php%2FjurnalTIMES%2Fissue%2Fview%2F23)

English (http://ejournal.stmik-time.ac.id/index.php/jurnalTIMES/user/setLocale/en_US?source=%2Findex.php%2FjurnalTIMES%2Fissue%2Fview%2F23)

JTM : Jurnal TIMES [ISSN Print 2337-3601 (http://issn.pdii.lipi.go.id/issn.cgi?daftar&1395053872&1&&)] [ISSN Online 2549-015X (http://issn.pdii.lipi.go.id/issn.cgi?daftar&1484036122&1&&)]

Dipublikasikan oleh

Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) STMIK TIME

Jalan Merbabu No. 32 AA-BB, Medan Kota, Kota Medan, Sumatera Utara.

Kode Pos: 20212 | **Telp :** 061-4561932 | **Contact :** ejournal@stmik-time.ac.id

JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XI No 2, Desember 2022
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

PENGEMBANGAN APLIKASI PENYANDIAN DATA MENGGUNAKAN ALGORITMA RIJNDAEL

Trihastuti Yuniati
Program Studi Teknik Informatika
Institut Teknologi Telkom Purwokerto
Jl. D.I. Panjaitan No.128 Purwokerto, Banyumas, Jawa Tengah
e-mail: trihastuti@ittelkom-pwt.ac.id

Abstrak

Algoritma Rijndael adalah salah satu algoritma kriptografi yang beroperasi dalam mode operasi *cipher block*. Algoritma ini meraih kemenangan dalam kontes yang diselenggarakan oleh *National Institute of Standards and Technology* (NIST). Rijndael dijadikan sebagai standar algoritma kriptografi terbaru untuk menggantikan *Data Encryption Standard* (DES) dan kemudian dikenal sebagai *Advanced Encryption Standard* (AES) setelah distandarkan oleh NIST. Rijndael mendukung panjang kunci dan ukuran blok 128-bit hingga 256-bit dengan step 32 bit. Tujuan dari penelitian ini adalah merancang dan mengembangkan aplikasi penyandian data dengan Algoritma Rijndael yang dapat digunakan untuk mengenkripsi berbagai jenis data baik teks, gambar, audio, maupun video dengan pilihan panjang kunci dan ukuran blok 128-bit, 192-bit dan 256-bit dalam mode operasi *Electronic Code Block* (ECB), *Cipher Block Chaining* (CBC), dan *Cipher Feedback* (CFB). Pengembangan aplikasi menggunakan bahasa pemrograman C#. Hasil penelitian menunjukkan aplikasi berbasis *desktop* telah berhasil dikembangkan. Hasil pengujian dengan metode *black box testing* menunjukkan bahwa 100% fitur dapat berjalan sebagaimana mestinya.

Kata Kunci: aes, enkripsi, kriptografi, rijndael

1. Pendahuluan

Data merupakan salah satu aset yang sangat penting bagi perusahaan, pemerintah, dan institusi-institusi pendidikan. Data ini bisa bersifat terbuka, yang dapat diketahui oleh semua orang, atau bersifat rahasia, yang hanya dapat diketahui oleh orang-orang tertentu. Data rahasia harus dikelola dengan metode khusus untuk menjaga kerahasiaannya [1]–[4]. Kriptografi adalah salah satu metode pengamanan data yang digunakan untuk menjaga kerahasiaan data. Metode ini melakukan penyandian data menjadi kode-kode yang tidak dimengerti sehingga, meskipun data tersebut jatuh ke tangan pihak yang tidak berhak, pihak tersebut tetap tidak dapat memahami informasi yang tersimpan dalam data tersebut [4]–[7]. Dengan menggunakan kriptografi, data rahasia dapat disandikan dan hanya dapat dibuka dengan kunci yang tepat. Ini membuat data rahasia tetap terlindungi dari pihak-pihak yang tidak berhak mengaksesnya [4], [6]–[8]. Untuk menjaga keamanan data, diperlukan algoritma kriptografi yang kuat dan sulit ditembus. Semakin kuat sebuah algoritma kriptografi, semakin lama waktu yang dibutuhkan untuk memecahkan data yang telah disandikan [6], [7].

Seiring dengan perkembangan teknologi komputer, algoritma kriptografi yang lebih kuat dan aman dibutuhkan untuk menangkal ancaman-ancaman keamanan data yang semakin canggih. Algoritma kriptografi yang baik harus memenuhi beberapa kriteria, seperti kecepatan, kekuatan, dan ketahanan terhadap teknik-teknik pemecahan yang ada [6]. Rijndael adalah salah satu algoritma kriptografi yang telah dijadikan sebagai standar dalam pengamanan data. Algoritma ini menggantikan algoritma *Data Encryption Standard* (DES) yang sebelumnya telah banyak digunakan. Rijndael memiliki keunggulan dalam hal performansi dan kesederhanaan kode. Algoritma ini juga menawarkan tingkat keamanan data yang tinggi untuk ukuran teknologi komputer saat ini. Rijndael dapat digunakan untuk menyandikan data dan memastikan bahwa data tersebut tidak dapat dibaca oleh pihak yang tidak berhak [9].

Beberapa penelitian sebelumnya telah ada yang mengembangkan aplikasi penyandian data dengan algoritma Rijndael, diantaranya penelitian [10] untuk mengamankan dokumen teks, penelitian [11] untuk mengamankan dokumen teks berupa berkas excel (.xls), text (.txt), dan word (.doc), penelitian [12] untuk mengenkripsi direktori *file* pada *website*, yaitu URL, penelitian [13] untuk mengamankan format suara, penelitian [14] untuk mengamankan dokumen gambar, dan penelitian [15] dapat mengamankan berbagai jenis berkas baik teks, suara, gambar, animasi, video, maupun aplikasi, seperti berkas berekstensi doc (teks), exe (aplikasi), mp3 (suara), swf (animasi), mp4, avi (video), dan lainnya. Pada penelitian ini juga dikembangkan aplikasi yang dapat menyandikan berbagai jenis data seperti penelitian [15] dengan perbedaan pada penelitian ini pengguna dapat memilih panjang kunci, ukuran blok, serta mode operasi yang ingin digunakan, yaitu pilihan panjang kunci dan ukuran blok 128-bit, 192-bit, dan 256-bit dan pilihan mode operasi ECB, CBC, dan CFB. Sedangkan pada penelitian [15] pengguna

tidak dapat memilih ketiga parameter tersebut, tidak disebutkan di dalam artikel parameter berapa panjang kunci, ukuran blok serta mode operasi yang digunakan pada aplikasi yang dikembangkan di penelitian [15].

2. Landasan Teori

Keamanan Data

Keamanan data merupakan hal yang sangat penting bagi sebagian besar organisasi, terutama yang menyimpan informasi sensitif atau kritikal. Keamanan data adalah proses atau tindakan yang dilakukan untuk menjaga integritas, keamanan, dan aksesibilitas data agar tidak terganggu atau dicuri oleh pihak yang tidak bertanggung jawab. Hal tersebut termasuk tindakan seperti enkripsi data, pembatasan akses ke data, dan menjalankan tindakan pencegahan terhadap serangan *cyber*. Ada empat aspek keamanan data yang juga merupakan tujuan utama dari kriptografi, yaitu: 1) integritas (*integrity*), yaitu menjamin bahwa data tidak ada yang diubah atau dirusak oleh pihak yang tidak bertanggungjawab; 2) kerahasiaan (*confidentiality*), yaitu menjaga agar informasi tidak dapat diketahui oleh pihak yang tidak memiliki otoritas untuk mengaksesnya; 3) autentikasi (*authentication*), yaitu proses verifikasi atau validasi kebenaran identitas pihak-pihak yang berkomunikasi maupun identifikasi kebenaran sumber data; dan 4) nir-penyangkalan (*non-repudiation*), yaitu mencegah pihak yang berkomunikasi melakukan penyangkalan bahwa ia telah mengirim atau menerima pesan [4], [6], [7].

Multimedia

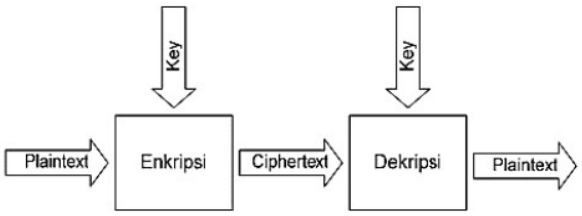
Kriptografi adalah salah satu teknik untuk menjaga kerahasiaan data. Di dalam kriptografi terdapat dua proses utama yang dilakukan, yaitu enkripsi dan dekripsi. Enkripsi atau *enciphering* adalah proses yang dilakukan untuk mengubah pesan asli, disebut sebagai *plaintext*, menjadi pesan kode-kode atau simbol-simbol yang tidak dapat dimengerti maknanya, disebut sebagai *ciphertext,* menggunakan suatu algoritma tertentu. Di dalam proses enkripsi tersebut biasanya diperlukan suatu kunci atau *key*. Berdasarkan kunci yang digunakan, algoritma kriptografi dibedakan menjadi dua, yaitu kriptografi kunci simetris dan kunci asimetris. Kriptografi kunci simetris adalah jika proses enkripsi dan dekripsi menggunakan kata kunci yang sama, sedangkan kriptografi kunci asimetris adalah jika kata kunci yang digunakan untuk enkripsi berbeda dengan kata kunci yang digunakan untuk dekripsi [4]–[7]. Secara matematis, misalkan P menyatakan *plaintext,* C menyatakan *ciphertext,* dan K menyatakan kunci atau *key*, maka fungsi enkripsi E memenuhi persamaan 2.1 [6], [7]:

$$E_K(P) = C \tag{2.1}$$

dan fungsi dekripsi D memenuhi persamaan 2.2 [6], [7]:

$$D_K(C) = P \tag{2.2}$$

Secara umum proses enkripsi dan dekripsi menggunakan kunci dapat ditunjukkan dengan skema pada Gambar 1.



Gambar 1. Proses Enkripsi dan Dekripsi Menggunakan Kunci [6], [7]

Algoritma kriptografi simetri yang beroperasi pada mode bit dapat dikelompokkan dalam dua kategori, yaitu: 1) *Stream cipher*, yang beroperasi dalam bentuk bit tunggal, dimana rangkaian bit dienkripsikan secara bit-per-bit; dan 2) *Block cipher*, yang beroperasi dalam bentuk blok bit, dimana rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya [6]. Algoritma kriptografi yang diterapkan dalam penelitian ini adalah algoritma Rijndael yang termasuk dalam kelompok algoritma *block cipher* dengan kunci simetris

Block Cipher

Block cipher pada dasarnya adalah proses penyandian terhadap blok-blok data yang ukuran bloknya sudah ditentukan. Terdapat berbagai macam mode operasi dalam *block cipher,* diantaranya *Electronic Code Block* (ECB), *Cipher Block Chaining* (CBC), dan *Cipher Feedback* (CFB).

1) ECB: mode ECB membagi *plaintext* menjadi blok-blok yang ukurannya telah ditentukan, disebut sebagai *Pi*. Masing-masing blok *plaintext Pi* dienkripsi secara individual dan independent menjai blok *ciphertext Ci*

menggunakan kunci yang telah ditentukan. Pada mode ECB, blok *plaintext* yang sama selalu dienkripsi menjadi blok *ciphertext* yang sama.

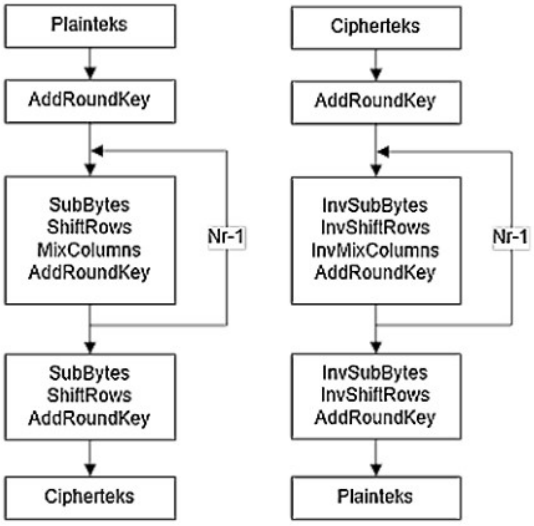
2) CBC: mode CBC menggunakan operasi umpan balik dari luaran blok sebelumnya, sehingga terlihat seperti berantai (*chaining*). Adanya umpan balik ini membuat adanya ketergantungan antar blok. *Ciphertext* luaran dari hasil enkripsi blok sebelumnya di-XOR-kan dengan *plaintext* blok *current*, baru kemudian dilakukan proses enkripsi untuk blok *current*. Umpan balik untuk blok pertama menggunakan suatu *initialization vector* (IV) yang diberikan oleh pengguna atau dibangkitkan secara acak oleh program.

3) CFB: mode CFB membagi satu blok *plaintext* menjadi unit-unit yang lebih untuk kemudian dienkripsi. Misal CFB 8-bit akan mengenkripsi tiap 8-bit data, sehingga dapat dikatakan CFB mengenkripsi tiap unit bit-per-bit, seperti *stream cipher*. Mode CFB memerlukan sebuah antrian (*queue*) yang berukuran sama dengan blok masukan.

Rijndael

Rijndael merupakan suatu algoritma kriptografi kunci simetris yang termasuk dalam kelompok *block cipher.* Algoritma ini dirancang oleh Vincent Rijmen dan John Daemen dalam rangka mengikuti kompetisi algoritma kriptografi pengganti DES yang diadakan oleh *National Institute of Technology* (NIST). Dalam kompetisi tersebut Rijndael terpilih sebagai pemenang, dan selanjutnya dikenal sebagai *Advanced Encryption Standard* (AES).

Algoritma Rijndael menggunakan operasi substitusi, permutasi, dan sejumlah putaran yang diterapkan pada setiap blok yang akan dienkripsi. Setiap putaran menggunakan kunci yang berbeda, yang disebut sebagai *round key*. Algoritma Rijndael mendukung panjang kunci dan ukuran blok dari 128-bit sampai dengan 256-bit dengan step 32-bit. Sebagaimana *block cipher* pada umumnya, Rijndael dapat dioperasikan dalam berbagai mode operasi, seperti ECB, CBC, dan CFB. Gambar 2 menunjukkan proses enkripsi (a) dan dekripsi (b) pada algoritma Rijndael.



Gambar 2. Diagram alur proses (a) enkripsi dan (b) dekripsi pada algoritma Rijndael

3. Metode Penelitian

Penelitian ini bertujuan untuk merancang dan membangun perangkat lunak berupa aplikasi berbasis desktop untuk enkripsi-dekripsi data dengan menerapkan algoritma kriptografi Rijndael. Adapun metode perancangan perangkat lunak yang diterapkan dalam penelitian ini mencakup 4 tahapan, yaitu analisis, perancangan, implementasi, dan pengujian.

Tahap Analisis

Pada tahap analisis, peneliti mengumpulkan informasi dan data yang diperlukan untuk menentukan kebutuhan dan spesifikasi aplikasi enkripsi-dekripsi yang akan dibangun. Hal ini meliputi studi literatur mengenai algoritma kriptografi Rijndael, identifikasi kebutuhan pengguna, dan analisis kebutuhan sistem.

Tahap Perancangan

Tahap selanjutnya adalah perancangan, di mana peneliti menentukan arsitektur dan fitur-fitur yang akan ada pada aplikasi enkripsi-dekripsi tersebut. Hal ini meliputi perancangan sistem, perancangan *user interface* dan perancangan diagram alur proses enkripsi-dekripsi yang akan diterapkan.

Perancangan Sistem

Pada perancangan sistem, aplikasi dikembangkan menggunakan perangkat lunak dan perangkat keras sebagaimana terlihat pada Tabel 1 dan Tabel 2 berikut:

Tabel 1. Spesifikasi Perangkat Lunak

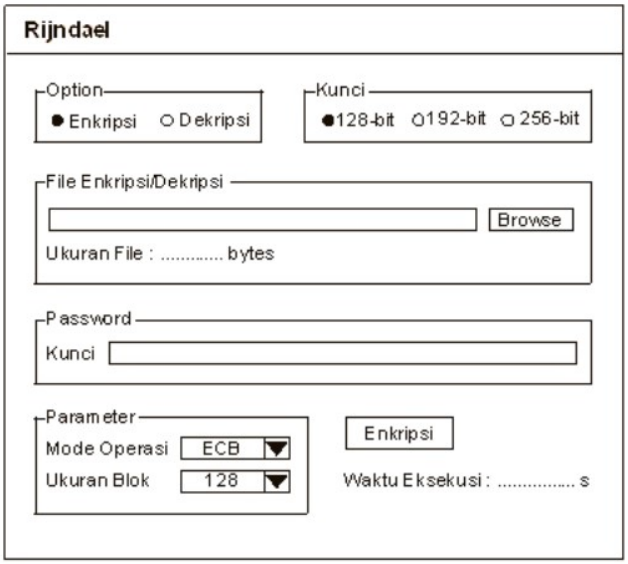
Jenis	Spesifikasi
Sistem Operasi	Microsoft Windows 10 Education
Bahasa Pemrograman	C#
Text Editor / IDE	Microsoft Visual Studio Enterprise 2017 Version 15.9.36

Tabel 2. Spesifikasi Perangkat Keras

Jenis	Spesifikasi
Laptop	Lenovo Ideapad S410p
Processor	Intel(R) Core™ i5-4200U CPU @ 1.60GHz 2.30 GHz
Memory	RAM DDR3 8 GB
System Type	64-bit Operating System, x64-based processor

Perancangan *User Interface*

Antarmuka sistem terdiri dari satu tampilan *form* yang dapat digunakan untuk proses enkripsi maupun dekripsi. Rancangan antarmuka sistem secara *low-end* dapat dilihat pada Gambar 3.

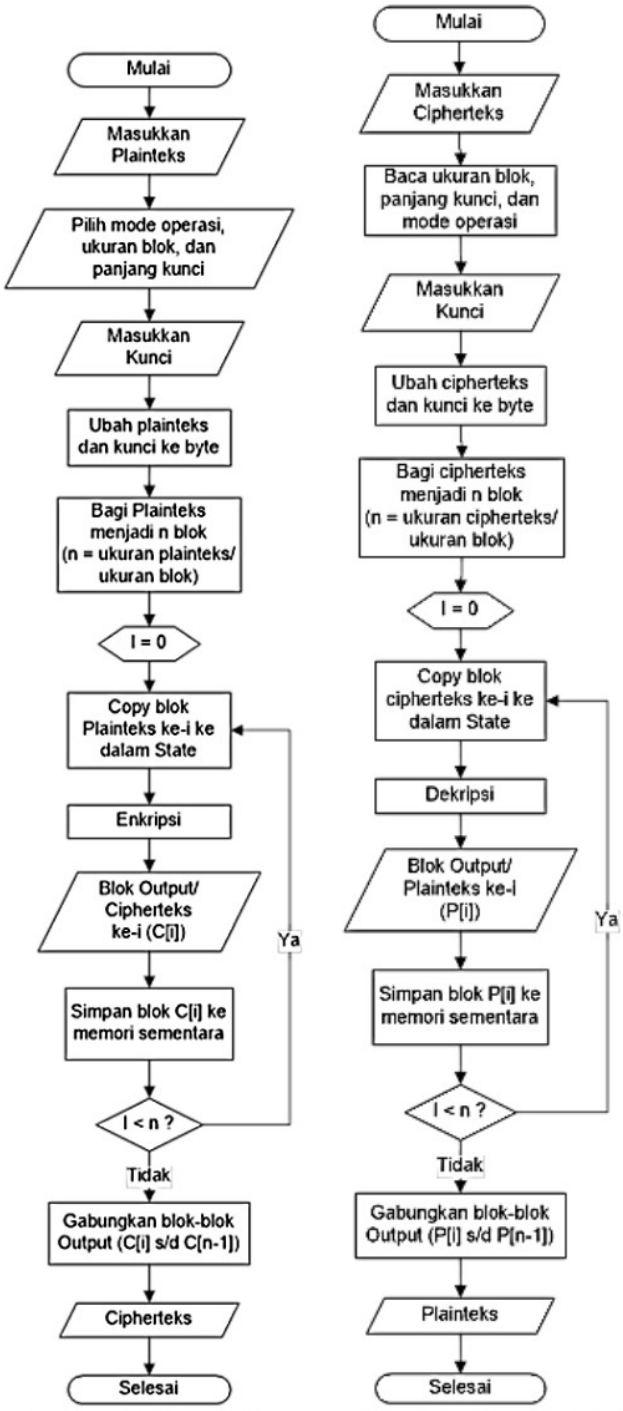


Gambar 3. Rancangan *Low-End User Interface*

Pada rancangan antarmuka aplikasi tersebut terdapat: a) *option button* untuk memilih proses yang akan dijalankan, yaitu enkripsi atau dekripsi; b) *option button* untuk memilih panjang kunci yang akan digunakan, yaitu 128-bit, 192-bit, atau 256-bit; c) *form* untuk memilih berkas yang akan dilakukan proses enkripsi atau dekripsi. Berkas untuk enkripsi dapat berupa dokumen apapun, baik dalam bentuk teks, audio, video, maupun aplikasi dalam berbagai format. Hasil enkripsi disimpan dalam ekstensi .rjn. Sedangkan berkas untuk dekripsi adalah dokumen dalam ekstensi .rjn. Pada *form* ini juga ditampilkan ukuran dari berkas yang dipilih; d) *input form* untuk menuliskan *password* enkripsi atau dekripsi; e) *dropdown list* untuk memilih mode operasi *block cipher* yang akan diterapkan pada proses enkripsi atau dekripsi, meliputi mode operasi *Electronic Code Block (ECB), Cipher Block Chaining (CBC)* dan *Cipher Feedback (CFB);* f) *dropdown list* untuk memilih ukuran blok yang akan diterapkan pada proses enkripsi dan dekripsi; serta g) *button* untuk mengeksekusi proses. Pada antarmuka ini juga ditampilkan lama waktu eksekusi program dalam satuan detik atau *sekon (s).*

Perancangan Diagram Alur Proses Enkripsi dan Dekripsi

Algoritma yang diterapkan adalah Rinjdael dengan ukuran blok dan panjang kunci 128, 192, dan 256, serta dalam mode operasi ECB, CBC, dan CFB. Perancangan algoritma disusun dalam bentuk diagram alir (*flowchart*). Terdapat dua proses utama yang dijalankan, yaitu proses enkripsi dan dekripsi. Gambar 4 menunjukkan alur proses enkripsi-dekripsi secara umum.



Gambar 4. Diagram alur proses (a) enkripsi dan (b) dekripsi

Tahap Implementasi

Pada tahap implementasi, peneliti membangun aplikasi enkripsi-dekripsi sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini meliputi pembuatan kode program, pengujian komponen-komponen aplikasi, dan integrasi komponen-komponen tersebut menjadi sebuah aplikasi yang utuh. Kode program dibangun menggunakan bahasa pemrograman C# dengan *text editor* Visual Studio.

Tahap Pengujian

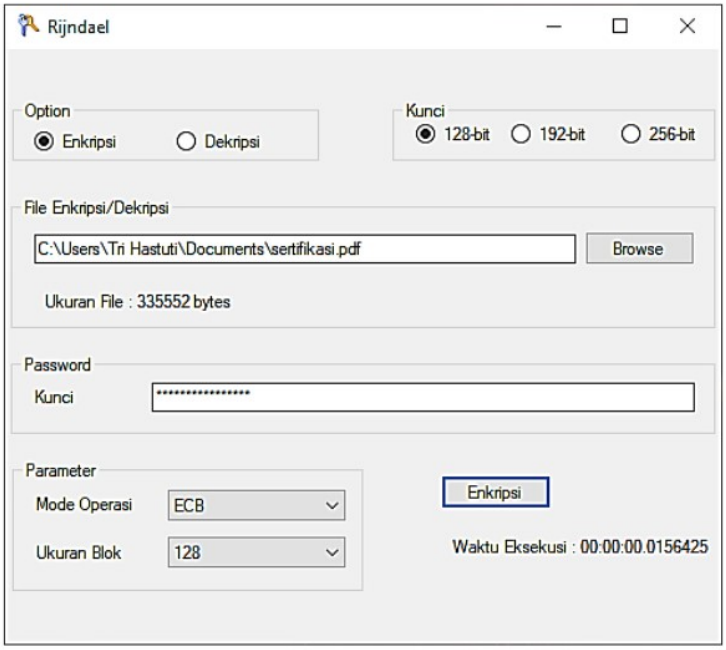
Terakhir, pada tahap pengujian, aplikasi yang telah dibangun diuji untuk mengetahui apakah aplikasi tersebut dapat bekerja sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Pengujian yang dilakukan adalah pengujian *black box* untuk menguji fungsionalitas sistem. Pengujian dilakukan dengan menggunakan data uji yang telah ditentukan sebelumnya, dan hasil pengujian kemudian dianalisis untuk menentukan apakah aplikasi tersebut sudah siap untuk dipakai atau masih perlu dilakukan perbaikan.

4. Hasil Penelitian

Tahap berikutnya setelah dilakukan analisis kebutuhan sistem dan perancangan adalah tahap implementasi dan pengujian system.

Implementasi

Sistem yang dibangun adalah sebuah aplikasi perangkat lunak berbasis *desktop* untuk menyandikan data. Perangkat lunak dibangun menggunakan bahasa pemrograman C# dengan *text editor* Visual Studio. Aplikasi ini dapat dijalankan untuk menangani dua proses, yaitu enkripsi dan dekripsi. Pada proses enkripsi terdapat masukan asli, disebut sebagai *plainteks*. Masukan dapat berupa berkas teks dalam berbagai format, seperti txt, doc, docx, xls, xlsx, serta pdf, berkas gambar dalam berbagai format, seperti jpg, png, dan bmp, berkas video dalam berbagai format, seperti mkv dan mp4, maupun berkas aplikasi dalam format .exe. Gambar 5 menunjukkan tampilan aplikasi yang telah dibangun.



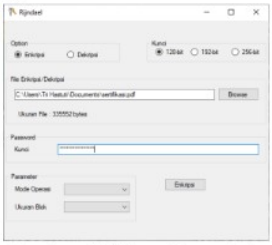
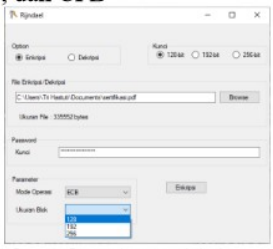
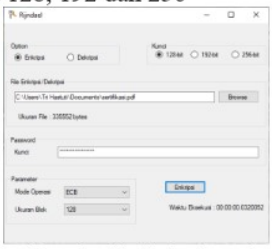
Gambar 5. Tampilan Aplikasi yang Dikembangkan

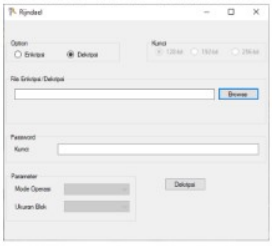
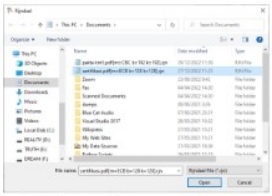
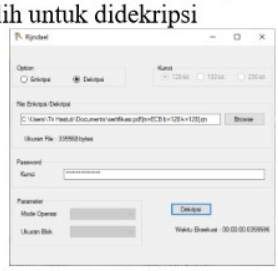
Tampilan *default* aplikasi berada pada proses enkripsi dengan panjang kunci 128-bit. Enkripsi dapat dilakukan pada semua jenis berkas, baik teks, audio, video, maupun aplikasi dalam berbagai format atau ekstensi. Berkas hasil dari proses dekripsi, disebut sebagai *ciphertext*, disimpan dalam ekstensi .rjn dengan tambahan nama berkas, yaitu "[m=mode b=blok k=kunci]" yang menunjukkan mode operasi, ukuran blok, serta panjang kunci yang digunakan. Misalkan berkas dengan nama "contoh.txt" dienkripsi menggunakan mode operasi CBC, panjang kunci 128-bit, dan ukuran blok 128-bit, maka berkas hasil enkripsinya akan menjadi "contoh.txt[m=CBC b=128 k=128].rjn". Hal tersebut dimaksudkan untuk mempermudah pada saat proses dekripsi, sehingga pengguna tidak perlu memasukkan parameter mode operasi, panjang kunci, serta ukuran blok, sebab sistem akan membaca dari namanya. Pengguna cukup memasukkan kata kunci. Selain mempermudah dekripsi, nama tambahan ini juga mengurangi resiko kesalahan pengguna dalam memasukkan parameter dan akan lebih menghemat waktu. Hasil dari proses dekripsi adalah seperti format berkas awal dengan tambahan nama _rjn di akhir. Misalkan berkas asli bernama "contoh.txt", maka berkas hasil dekripsi akan bernama "contoh_rjn.txt" untuk membedakan dengan berkas asli.

Pengujian

Pengujian sistem dilakukan menggunakan metode *black box testing* untuk menguji fungsionalitas perangkat lunak. Hasil dari *black box testing* ditunjukkan oleh Tabel 3.

Tabel 3. Hasil Pengujian

No	Aktivitas Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Memilih berkas yang akan dienkripsi menggunakan *button Browse*	Berkas berhasil dipilih dan *path* berkas serta ukuran berkas muncul di *form* yang disediakan	 Berkas berhasil dipilih dan path berkas serta ukuran berkas muncul di *form*
2	Memasukkan kunci untuk enkripsi	Kunci yang dimasukkan oleh user berhasil diinputkan pada form yang disediakan dalam tampilan simbol-simbol, serta panjang kunci dibatasi maksimal sesuai dengan ukuran panjang kunci yang dipilih pada option kunci.	 Kunci berhasil diinputkan dalam tampilan simbol dan panjang kunci sudah dibatasi sesuai dengan option yang dipilih
3	Memilih mode operasi *block cipher*	Pilihan mode operasi ECB, CBC, dan CFB muncul dalam *dropdown list* pada parameter mode operasi	 *Dropdown list* mode operasi memunculkan mode yang ECB, CBC, dan CFB
4	Memilih ukuran blok	Pilihan ukuran blok 128, 192 dan 256 muncul dalam *dropdown list* pada parameter ukuran blok	 *Dropdown list* ukuran blok memunculkan pilihan ukuran blok untuk proses enkripsi/dekripsi, yaitu 128, 192 dan 256
5	Melakukan proses enkripsi sesuai parameter panjang kunci, mode operasi, dan ukuran blok yang dipilih dengan menekan *button* Enkripsi	Proses enkripsi berhasil dilakukan dan waktu eksekusi ditampilkan di *form*	 Enkripsi berhasil dilakukan dan muncul informasi lamanya waktu eksekusi di *form*

No	Aktivitas Pengujian	Hasil yang Diharapkan	Kesimpulan
		Berkas hasil enkripsi menjadi berformat .rjn dan berubah menjadi kode-kode tidak terbaca	 Berkas hasil enkripsi berubah menjadi kode-kode tidak terbaca
6	Memilih mode Dekripsi	*Option* Dekripsi terpilih dan parameter kunci, mode operasi, serta ukuran blok dalam kondisi *inactive*	 *Option* Dekripsi berhasil dipilih, *option* kunci dan parameter lainnya dalam keadaan *inactive*
7	Memilih berkas untuk didekripsi dengan menekan *button Browse*	Hanya berkas dengan tipe .rjn yang dapat dipilih	 Hanya berkas .rjn yang dapat dipilih untuk didekripsi
8	Melakukan proses dekripsi dengan membaca otomatis parameter panjang kunci, mode operasi, dan ukuran blok berdasarkan nama berkas dengan menekan *button* Dekripsi	Proses dekripsi berhasil dilakukan dan waktu eksekusi ditampilkan di *form*	 Dekripsi berhasil dilakukan dan muncul informasi lamanya waktu eksekusi di *form*
		Berkas hasil dekripsi berhasil dipulihkan dengan tambahan nama berkas _rjn di akhir	 Berkas hasil dekripsi berhasil dipulihkan

5. Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan sebagaimana diuraikan di dalam bab sebelumnya, maka dapat disimpulkan bahwa perancangan aplikasi penyandian data menggunakan algoritma Rijndael berhasil dibangun menggunakan bahasa pemrograman C#. Hasil pengujian dengan metode *black box testing* menunjukkan bahwa aplikasi dapat berjalan dengan baik, dibuktikan dengan 100% fitur yang ada dapat berjalan sebagaimana mestinya tanpa ada *error*.

6. Daftar Pustaka

[1] C. Chazar, "Standar Manajemen Keamanan Informasi Berbasis ISO/IEC 27001: 2005," *J. Inf.*, vol. VII, no. 2, pp. 48–57, 2017.
[2] M. Syafrizal, "ISO 17799 : Standar Sistem Manajemen Keamanan Informasi," *Semin. Nas. Teknol. 2007 (SNT 2007)*, vol. 2007, no. November, pp. 1–12, 2007.
[3] IT Governance, "Information Security & ISO 27001," *IT Gov. Green Pap.*, 2018.
[4] Jamaludin *et al.*, *Kriptografi: Teknik Keamanan Data*. Yayasan Kita Menulis, 2022.
[5] H. Mukhtar, *Kriptografi untuk Keamanan Data*. Yogyakarta: Deepublish, 2018.

[6] R. Munir, *Kriptografi*, 2nd ed. Bandung: Penerbit Informatika, 2019.
[7] E. Setyaningsih, *Kriptografi & Implementasinya Menggunakan Matlab*. Yogyakarta: Penerbit Andi, 2015.
[8] M. Aggusti, M. D. Bhastary, and S. Khairani, *Sistem Informasi Manajemen*. Medan: Perdana Publishing, 2016.
[9] T. Yuniati, E. Suryani, and A. Aziz, "Pengaruh Variasi Panjang Kunci, Ukuran Blok, dan Mode Operasi Terhadap Waktu Eksekusi pada Algoritma Rijndael," *J. Teknol. Inf. ITSmart*, vol. 1, no. 1, p. 20, 2016, doi: 10.20961/its.v1i1.580.
[10] T. Tiepsi and K. Siregar, "Penerapan Algoritma Rijndael untuk Mengamankan Teks," *KAKIFIKOM (Kumpulan Artik. Karya Ilm. Fak. Ilmu Komputer)*, vol. 02, no. 338, pp. 47–53, 2020, doi: 10.54367/kakifikom.v1i2.637.
[11] A. Sifaunajah, "Implementasi Algoritma Rijndael Pada Enkripsi Dokumen Elektronik untuk Keamanan Informasi," *Exact Pap. Compil.*, vol. 2, no. 2, pp. 255–258, 2020.
[12] I. M. A. Bhaskara, D. M. Wiharta, and O. Saputra, "Perancangan Sistem Penyedia File Sharing dengan Enkripsi URL menggunakan Algoritma Rijndael," *Maj. Ilm. Teknol. Elektro*, vol. 19, no. 2, p. 171, 2020, doi: 10.24843/mite.2020.v19i02.p08.
[13] K. D. R. Sianipar, S. W. Siahaan, M. Siregar, and I. Gunawan, "Pengamanan File Suara Menggunakan Kriptografi Algoritma Rijndael Dengan Proses Enkripsi Dan Dekripsi," *TECHSI - J. Tek. Inform.*, vol. 11, no. 3, p. 431, 2019, doi: 10.29103/techsi.v11i3.1967.
[14] A. C. Saputra and A. S. Saragih, "Implementasi Algoritma Rijndael Dalam Enkripsi Dan Dekripsi Gambar Digital Berbasis Web," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 14, no. 1, pp. 52–63, 2020, doi: 10.47111/jti.v14i1.609.
[1] [15] A. A. Pradypta, "Perancangan Aplikasi Data Security Dalam Melindungi Informasi Digital Menggunakan Teknik Algoritma Rijndael Berbasis Desktop," *J. Maklumatika*, vol. 9, no. 1, pp. 68–76, 2022, [Online]. Available: https://maklumatika.i-tech.ac.id/index.php/maklumatika/article/view/141/138

PAPER NAME

jurnal times - thy.docx

WORD COUNT

2669 Words

CHARACTER COUNT

17235 Characters

PAGE COUNT

8 Pages

FILE SIZE

1003.0KB

SUBMISSION DATE

Feb 3, 2023 4:07 PM GMT+7

REPORT DATE

Feb 3, 2023 4:08 PM GMT+7

24% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 21% Internet database
- Crossref database
- 15% Submitted Works database
- 7% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Manually excluded text blocks

PENGEMBANGAN APLIKASI PENYANDIAN DATA MENGGUNAKAN ALGORITMA RIJNDAEL

Trihastuti Yuniati[1]
Program Studi S1 Teknik Informatika
Institut Teknologi Telkom Purwokerto
Jl. D.I. Panjaitan No.128 Purwokerto, Banyumas, Jawa Tengah
e-mail: trihastuti@ittelkom-pwt.ac.id[1]

Abstrak

Algoritma Rijndael adalah salah satu algoritma kriptografi yang beroperasi dalam mode operasi *cipher block*. Algoritma ini meraih kemenangan dalam kontes yang diselenggarakan oleh *National Institute of Standards and Technology* (NIST). Rijndael dijadikan sebagai standar algoritma kriptografi terbaru untuk menggantikan *Data Encryption Standard* (DES) dan kemudian dikenal sebagai *Advanced Encryption Standard* (AES) setelah distandarkan oleh NIST. Rijndael mendukung panjang kunci dan ukuran blok 128-bit hingga 256-bit dengan step 32 bit. Tujuan dari penelitian ini adalah merancang dan mengembangkan aplikasi penyandian data dengan Algoritma Rijndael yang dapat digunakan untuk mengenkripsi berbagai jenis data baik teks, gambar, audio, maupun video dengan pilihan panjang kunci dan ukuran blok 128-bit, 192-bit dan 256-bit dalam mode operasi *Electronic Code Block* (ECB), *Cipher Block Chaining* (CBC), dan *Cipher Feedback* (CFB). Pengembangan aplikasi menggunakan bahasa pemrograman C#. Hasil penelitian menunjukkan aplikasi berbasis *desktop* telah berhasil dikembangkan. Hasil pengujian dengan metode *black box testing* menunjukkan bahwa 100% fitur dapat berjalan sebagaimana mestinya.

Kata Kunci: aes, enkripsi, kriptografi, rijndael

1. Pendahuluan

Data merupakan salah satu aset yang sangat penting bagi perusahaan, pemerintah, dan institusi-institusi pendidikan. Data ini bisa bersifat terbuka, yang dapat diketahui oleh semua orang, atau bersifat rahasia, yang hanya dapat diketahui oleh orang-orang tertentu. Data rahasia harus dikelola dengan metode khusus untuk menjaga kerahasiaannya [1]–[4]. Kriptografi adalah salah satu metode pengamanan data yang digunakan untuk menjaga kerahasiaan data. Metode ini melakukan penyandian data menjadi kode-kode yang tidak dimengerti sehingga, meskipun data tersebut jatuh ke tangan pihak yang tidak berhak, pihak tersebut tetap tidak dapat memahami informasi yang tersimpan dalam data tersebut [4]–[7]. Dengan menggunakan kriptografi, data rahasia dapat disandikan dan hanya dapat dibuka dengan kunci yang tepat. Ini membuat data rahasia tetap terlindungi dari pihak-pihak yang tidak berhak mengaksesnya [4], [6]–[8]. Untuk menjaga keamanan data, diperlukan algoritma kriptografi yang kuat dan sulit ditembus. Semakin kuat sebuah algoritma kriptografi, semakin lama waktu yang dibutuhkan untuk memecahkan data yang telah disandikan [6], [7].

Seiring dengan perkembangan teknologi komputer, algoritma kriptografi yang lebih kuat dan aman dibutuhkan untuk menangkal ancaman-ancaman keamanan data yang semakin canggih. Algoritma kriptografi yang baik harus memenuhi beberapa kriteria, seperti kecepatan, kekuatan, dan ketahanan terhadap teknik-teknik pemecahan yang ada [6]. Rijndael adalah salah satu algoritma kriptografi yang telah dijadikan sebagai standar dalam pengamanan data. Algoritma ini menggantikan algoritma *Data Encryption Standard* (DES) yang sebelumnya telah banyak digunakan. Rijndael memiliki keunggulan dalam hal performansi dan kesederhanaan kode. Algoritma ini juga menawarkan tingkat keamanan data yang tinggi untuk ukuran teknologi komputer saat ini. Rijndael dapat digunakan untuk menyandikan data dan memastikan bahwa data tersebut tidak dapat dibaca oleh pihak yang tidak berhak [9].

Beberapa penelitian sebelumnya telah ada yang mengembangkan aplikasi penyandian data dengan algoritma Rijndael, diantaranya penelitian [10] untuk mengamankan dokumen teks, penelitian [11] untuk mengamankan dokumen teks berupa berkas excel (.xls), text (.txt), dan word (.doc), penelitian [12] untuk mengenkripsi direktori *file* pada *website*, yaitu URL, penelitian [13] untuk mengamankan format suara, penelitian [14] untuk mengamankan dokumen gambar, dan penelitian [15] dapat mengamankan berbagai jenis berkas baik teks, suara, gambar, animasi, video, maupun aplikasi, seperti berkas berekstensi doc (teks), exe (aplikasi), mp3 (suara), swf (animasi), mp4, avi (video), dan lainnya. Pada penelitian ini juga dikembangkan aplikasi yang dapat menyandikan berbagai jenis data seperti penelitian [15] dengan perbedaan pada penelitian ini pengguna dapat memilih panjang kunci, ukuran blok, serta mode operasi yang ingin digunakan, yaitu pilihan panjang kunci dan ukuran blok 128-bit, 192-bit, dan 256-bit dan pilihan mode operasi ECB, CBC, dan CFB. Sedangkan pada penelitian [15] pengguna tidak dapat memilih ketiga parameter tersebut, tidak disebutkan di dalam artikel parameter berapa panjang kunci, ukuran blok serta mode operasi yang digunakan pada aplikasi yang dikembangkan di penelitian [15].

2. Tinjauan Pustaka

Keamanan Data

Keamanan data merupakan hal yang sangat penting bagi sebagian besar organisasi terutama yang menyimpan informasi sensitif atau kritikal. Keamanan data adalah proses atau tindakan yang dilakukan untuk menjaga integritas, keamanan, dan aksesibilitas data agar tidak terganggu atau dicuri oleh pihak yang tidak bertanggung jawab. Hal tersebut termasuk tindakan seperti enkripsi data, pembatasan akses ke data, dan menjalankan tindakan pencegahan terhadap serangan *cyber*. Ada empat aspek keamanan data yang juga merupakan tujuan utama dari kriptografi, yaitu: 1) integritas (*integrity*), yaitu menjamin bahwa data tidak ada yang diubah atau dirusak oleh pihak yang tidak bertanggungjawab; 2) kerahasiaan (*confidentiality*), yaitu menjaga agar informasi tidak dapat diketahui oleh pihak yang tidak memiliki otoritas untuk mengaksesnya; 3) autentikasi (*authentication*), yaitu proses verifikasi atau validasi kebenaran identitas pihak-pihak yang berkomunikasi maupun identifikasi kebenaran sumber data; dan 4) nir-penyangkalan (*non-repudiation*), yaitu mencegah pihak yang berkomunikasi melakukan penyangkalan bahwa ia telah mengirim atau menerima pesan [4], [6], [7].

Kriptografi

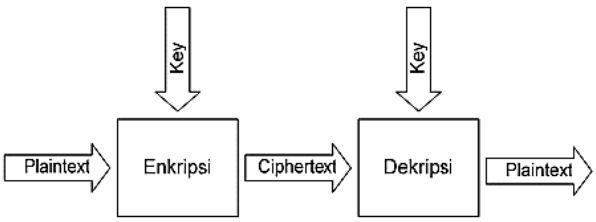
Kriptografi adalah salah satu teknik untuk menjaga kerahasiaan data. Di dalam kriptografi terdapat dua proses utama yang dilakukan, yaitu enkripsi dan dekripsi. Enkripsi atau *enciphering* adalah proses yang dilakukan untuk mengubah pesan asli, disebut sebagai *plaintext*, menjadi pesan kode-kode atau simbol-simbol yang tidak dapat dimengerti maknanya, disebut sebagai *ciphertext*, menggunakan suatu algoritma tertentu. Di dalam proses enkripsi tersebut biasanya diperlukan suatu kunci atau *key*. Berdasarkan kunci yang digunakan, algoritma kriptografi dibedakan menjadi dua, yaitu kriptografi kunci simetris dan kunci asimetris. Kriptografi kunci simetris adalah jika proses enkripsi dan dekripsi menggunakan kata kunci yang sama, sedangkan kriptografi kunci asimetris adalah jika kata kunci yang digunakan untuk enkripsi berbeda dengan kata kunci yang digunakan untuk dekripsi [4]–[7]. Secara matematis, misalkan P menyatakan *plaintext*, C menyatakan *ciphertext*, dan K menyatakan kunci atau *key*, maka fungsi enkripsi E memenuhi persamaan 2.1 [6], [7]:

$$E_K(P) = C \tag{2.1}$$

dan fungsi dekripsi D memenuhi persamaan 2.2 [6], [7]:

$$D_K(C) = P \tag{2.2}$$

Secara umum proses enkripsi dan dekripsi menggunakan kunci dapat ditunjukkan dengan skema pada Gambar 1.



Gambar 1. Proses Enkripsi dan Dekripsi Menggunakan Kunci [6], [7]

Algoritma kriptografi simetri yang beroperasi pada mode bit dapat dikelompokkan dalam dua kategori, yaitu: 1) *Stream cipher*, yang beroperasi dalam bentuk bit tunggal, dimana rangkaian bit dienkripsikan secara bit-per-bit; dan 2) *Block cipher*, yang beroperasi dalam bentuk blok bit, dimana rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya [6]. Algoritma kriptografi yang diterapkan dalam penelitian ini adalah algoritma Rijndael yang termasuk dalam kelompok algoritma *block cipher* dengan kunci simetris.

Block Cipher

Block cipher pada dasarnya adalah proses penyandian terhadap blok-blok data yang ukuran bloknya sudah ditentukan. Terdapat berbagai macam mode operasi dalam *block cipher*, diantaranya *Electronic Code Block* (ECB), *Cipher Block Chaining* (CBC), dan *Cipher Feedback* (CFB).

1) ECB: mode ECB membagi *plaintext* menjadi blok-blok yang ukurannya telah ditentukan, disebut sebagai *Pi*. Masing-masing blok *plaintext Pi* dienkripsi secara individual dan independent menjai blok *ciphertext Ci* menggunakan kunci yang telah ditentukan. Pada mode ECB, blok *plaintext* yang sama selalu dienkripsi menjadi blok *ciphertext* yang sama.

2) CBC: mode CBC menggunakan operasi umpan balik dari luaran blok sebelumnya, sehingga terlihat seperti berantai (*chaining*). Adanya umpan balik ini membuat adanya ketergantungan antar blok. *Ciphertext* luaran dari hasil enkripsi blok sebelumnya di-XOR-kan dengan *plaintext* blok *current*, baru kemudian dilakukan proses

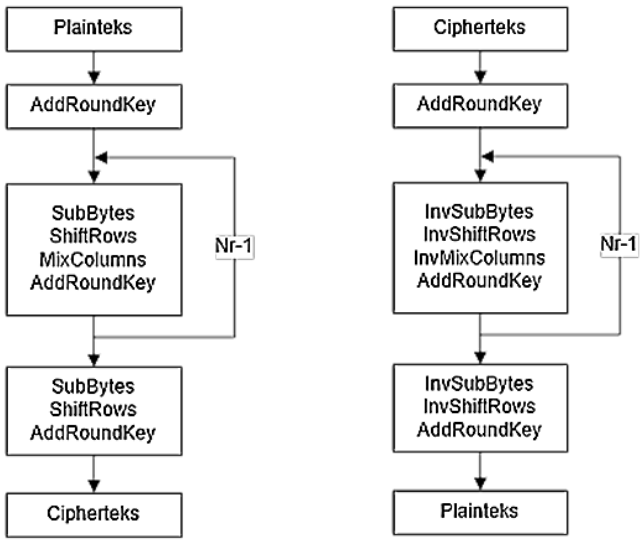
enkripsi untuk blok *current*. Umpan balik untuk blok pertama menggunakan suatu *initialization vector* (IV) yang diberikan oleh pengguna atau dibangkitkan secara acak oleh program.

3) CFB: mode CFB membagi satu blok *plaintext* menjadi unit-unit yang lebih untuk kemudian dienkripsi. Misal CFB 8-bit akan mengenkripsi tiap 8-bit data, sehingga dapat dikatakan CFB mengenkripsi tiap unit bit-per-bit, seperti *stream cipher*. Mode CFB memerlukan sebuah antrian (*queue*) yang berukuran sama dengan blok masukan.

Rijndael

Rijndael merupakan suatu algoritma kriptografi kunci simetris yang termasuk dalam kelompok *block cipher*. Algoritma ini dirancang oleh Vincent Rijmen dan John Daemen dalam rangka mengikuti kompetisi algoritma kriptografi pengganti DES yang diadakan oleh *National Institute of Technology* (NIST). Dalam kompetisi tersebut Rijndael terpilih sebagai pemenang, dan selanjutnya dikenal sebagai *Advanced Encryption Standard* (AES).

Algoritma Rijndael menggunakan operasi substitusi, permutasi, dan sejumlah putaran yang diterapkan pada setiap blok yang akan dienkripsi. Setiap putaran menggunakan kunci yang berbeda, yang disebut sebagai *round key*. Algoritma Rijndael mendukung panjang kunci dan ukuran blok dari 128-bit sampai dengan 256-bit dengan step 32-bit. Sebagaimana *block cipher* pada umumnya, Rijndael dapat dioperasikan dalam berbagai mode operasi, seperti ECB, CBC, dan CFB. Gambar 2 menunjukkan proses enkripsi (a) dan dekripsi (b) pada algoritma Rijndael.



(a) (b)

Gambar 2. Diagram alur proses (a) enkripsi dan (b) dekripsi pada algoritma Rijndael

3. Metode Penelitian

Penelitian ini bertujuan untuk merancang dan membangun perangkat lunak berupa aplikasi berbasis desktop untuk enkripsi-dekripsi data dengan menerapkan algoritma kriptografi Rijndael. Adapun metode perancangan perangkat lunak yang diterapkan dalam penelitian ini mencakup 4 tahapan, yaitu analisis, perancangan, implementasi, dan pengujian.

Tahap Analisis

Pada tahap analisis, peneliti mengumpulkan informasi dan data yang diperlukan untuk menentukan kebutuhan dan spesifikasi aplikasi enkripsi-dekripsi yang akan dibangun. Hal ini meliputi studi literatur mengenai algoritma kriptografi Rijndael, identifikasi kebutuhan pengguna, dan analisis kebutuhan sistem.

Tahap Perancangan

Tahap selanjutnya adalah perancangan, di mana peneliti menentukan arsitektur dan fitur-fitur yang akan ada pada aplikasi enkripsi-dekripsi tersebut. Hal ini meliputi perancangan sistem, perancangan *user interface* dan perancangan diagram alur proses enkripsi-dekripsi yang akan diterapkan.

Perancangan Sistem

Pada perancangan sistem, aplikasi dikembangkan menggunakan perangkat lunak dan perangkat keras sebagaimana terlihat pada Tabel 1 dan Tabel 2 berikut:

Tabel 1. Spesifikasi Perangkat Lunak

Jenis	Spesifikasi

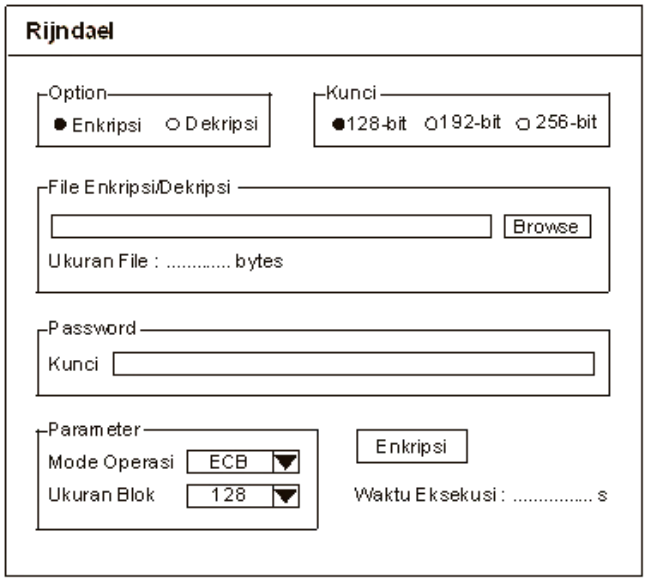
Sistem Operasi	Microsoft Windows 10 Education
Bahasa Pemrograman	C#
Text Editor / IDE	Microsoft Visual Studio Enterprise 2017 Version 15.9.36

Tabel 2. Spesifikasi Perangkat Keras

Jenis	Spesifikasi
Laptop	Lenovo Ideapad S410p
Processor	Intel(R) Core™ i5-4200U CPU @ 1.60GHz 2.30 GHz
Memory	RAM DDR3 8 GB
System Type	64-bit Operating System, x64-based processor

Perancangan *User Interface*

Antarmuka sistem terdiri dari satu tampilan *form* yang dapat digunakan untuk proses enkripsi maupun dekripsi. Rancangan antarmuka sistem secara *low-end* dapat dilihat pada Gambar 3.



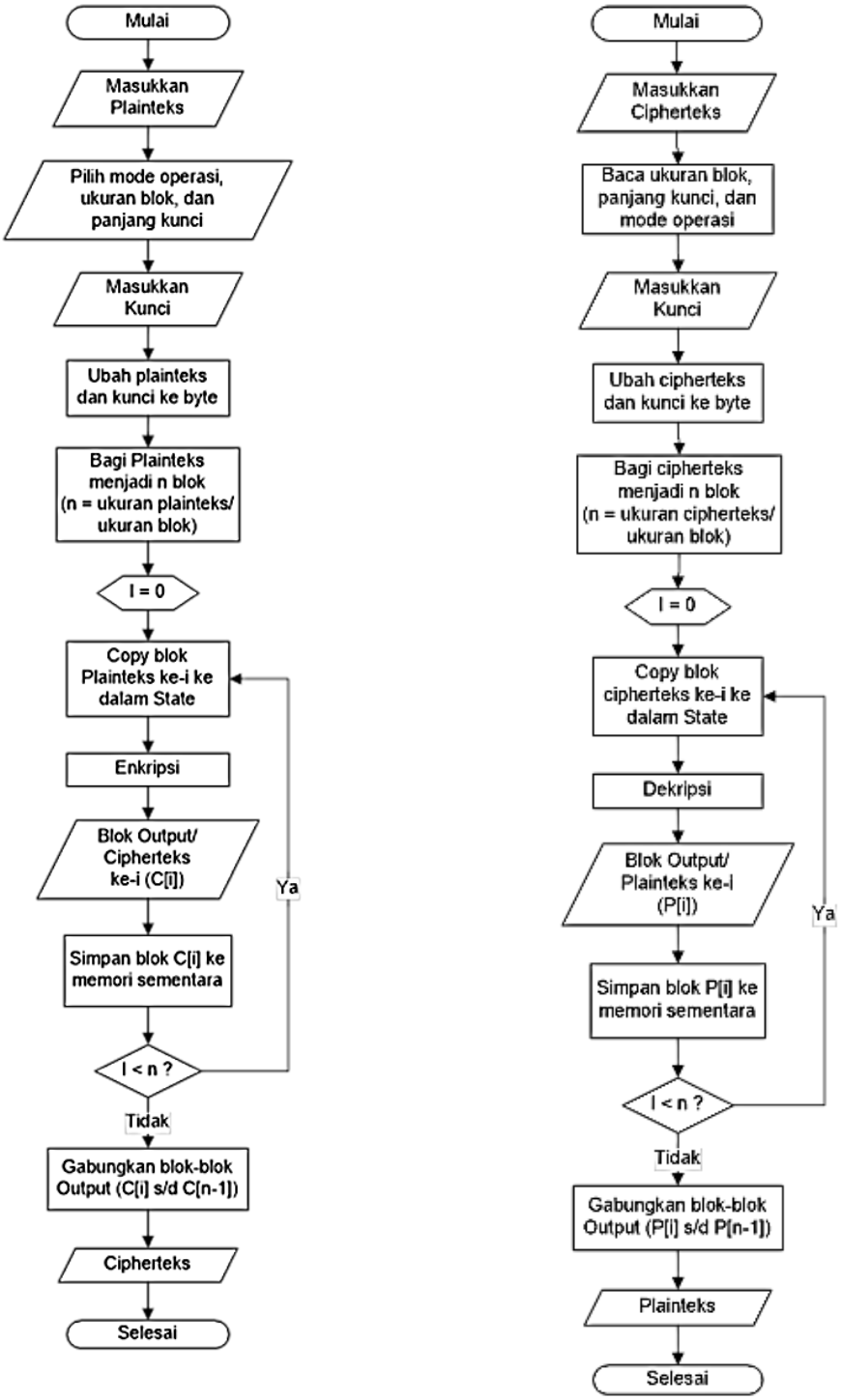
Gambar 3. Rancangan *Low-End User Interface*

Pada rancangan antarmuka aplikasi tersebut terdapat: a) *option button* untuk memilih proses yang akan dijalankan, yaitu enkripsi atau dekripsi; b) *option button* untuk memilih panjang kunci yang akan digunakan, yaitu 128-bit, 192-bit, atau 256-bit; c) *form* untuk memilih berkas yang akan dilakukan proses enkripsi atau dekripsi. Berkas untuk enkripsi dapat berupa dokumen apapun, baik dalam bentuk teks, audio, video, maupun aplikasi dalam berbagai format. Hasil enkripsi disimpan dalam ekstensi .rjn. Sedangkan berkas untuk dekripsi adalah dokumen dalam ekstensi .rjn. Pada *form* ini juga ditampilkan ukuran dari berkas yang dipilih; d) *input form* untuk menuliskan *password* enkripsi atau dekripsi; e) *dropdown list* untuk memilih mode operasi *block cipher* yang akan diterapkan pada proses enkripsi atau dekripsi, meliputi mode operasi *Electronic Code Block (ECB), Cipher Block Chaining (CBC)* dan *Cipher Feedback (CFB);* f) *dropdown list* untuk memilih ukuran blok yang akan diterapkan pada proses enkripsi dan dekripsi; serta g) *button* untuk mengeksekusi proses. Pada antarmuka ini juga ditampilkan lama waktu eksekusi program dalam satuan detik atau *sekon (s).*

Perancangan Diagram Alur Proses Enkripsi dan Dekripsi

Algoritma yang diterapkan adalah Rinjdael dengan ukuran blok dan panjang kunci 128, 192, dan 256, serta dalam mode operasi ECB, CBC, dan CFB. Perancangan algoritma disusun dalam bentuk diagram alir (*flowchart*).

Terdapat dua proses utama yang dijalankan, yaitu proses enkripsi dan dekripsi. Gambar 4 menunjukkan alur proses enkripsi-dekripsi secara umum.



Gambar 4. Diagram alur proses (a) enkripsi dan (b) dekripsi

Tahap Implementasi

Pada tahap implementasi, peneliti membangun aplikasi enkripsi-dekripsi sesuai dengan rancangan yang telah dibuat sebelumnya. Hal ini meliputi pembuatan kode program, pengujian komponen-komponen aplikasi, dan integrasi komponen-komponen tersebut menjadi sebuah aplikasi yang utuh. Kode program dibangun menggunakan bahasa pemrograman C# dengan *text editor* Visual Studio.

Tahap Pengujian

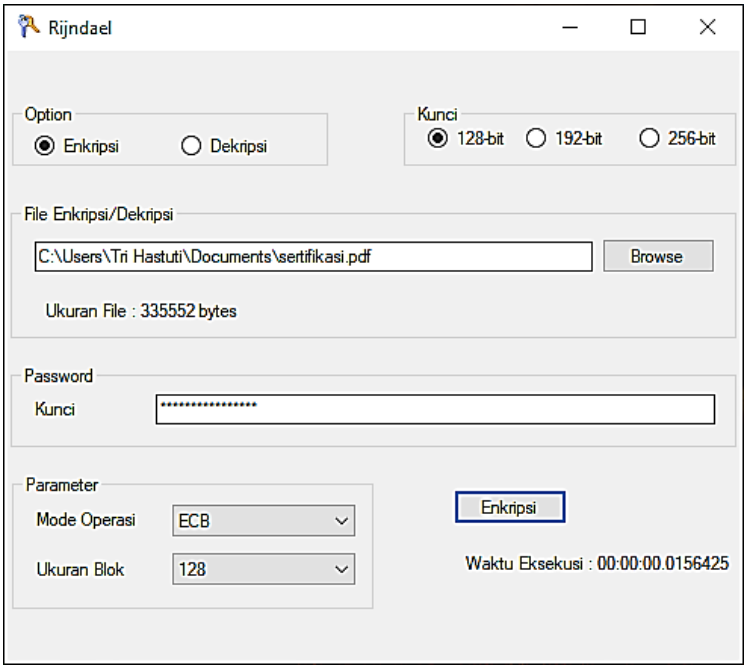
Terakhir, pada tahap pengujian, aplikasi yang telah dibangun diuji untuk mengetahui apakah aplikasi tersebut dapat bekerja sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Pengujian yang dilakukan adalah pengujian *black box* untuk menguji fungsionalitas sistem. Pengujian dilakukan dengan menggunakan data uji yang telah ditentukan sebelumnya, dan hasil pengujian kemudian dianalisis untuk menentukan apakah aplikasi tersebut sudah siap untuk dipakai atau masih perlu dilakukan perbaikan.

4. Hasil dan Pembahasan

Tahap berikutnya setelah dilakukan analisis kebutuhan sistem dan perancangan adalah tahap implementasi dan pengujian sistem.

Implementasi

Sistem yang dibangun adalah sebuah aplikasi perangkat lunak berbasis *desktop* untuk menyandikan data. Perangkat lunak dibangun menggunakan bahasa pemrograman C# dengan *text editor* Visual Studio. Aplikasi ini dapat dijalankan untuk menangani dua proses, yaitu enkripsi dan dekripsi. Pada proses enkripsi terdapat masukan asli, disebut sebagai *plainteks*. Masukan dapat berupa berkas teks dalam berbagai format, seperti txt, doc, docx, xls, xlsx, serta pdf, berkas gambar dalam berbagai format, seperti jpg, png, dan bmp, berkas video dalam berbagai format, seperti mkv dan mp4, maupun berkas aplikasi dalam format .exe. Gambar 5 menunjukkan tampilan aplikasi yang telah dibangun.



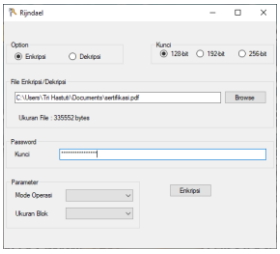
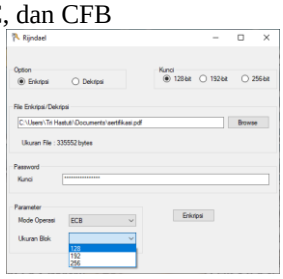
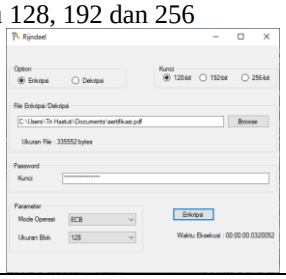
Gambar 5. Tampilan Aplikasi yang Dikembangkan

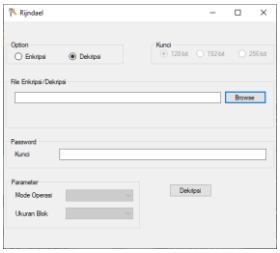
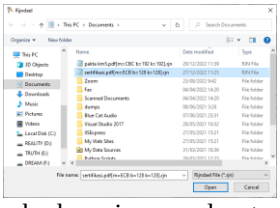
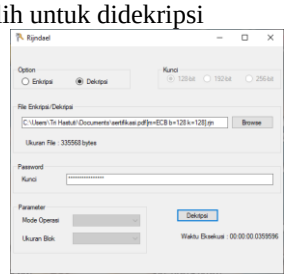
Tampilan *default* aplikasi berada pada proses enkripsi dengan panjang kunci 128-bit. Enkripsi dapat dilakukan pada semua jenis berkas, baik teks, audio, video, maupun aplikasi dalam berbagai format atau ekstensi. Berkas hasil dari proses dekripsi, disebut sebagai *ciphertext*, disimpan dalam ekstensi .rjn dengan tambahan nama berkas, yaitu "[m=mode b=blok k=kunci]" yang menunjukkan mode operasi, ukuran blok, serta panjang kunci yang digunakan. Misalkan berkas dengan nama "contoh.txt" dienkripsi menggunakan mode operasi CBC, panjang kunci 128-bit, dan ukuran blok 128-bit, maka berkas hasil enkripsinya akan menjadi "contoh.txt[m=CBC b=128 k=128].rjn". Hal tersebut dimaksudkan untuk mempermudah pada saat proses dekripsi, sehingga pengguna tidak perlu memasukkan parameter mode operasi, panjang kunci, serta ukuran blok, sebab sistem akan membaca dari namanya. Pengguna cukup memasukkan kata kunci. Selain mempermudah dekripsi, nama tambahan ini juga mengurangi resiko kesalahan pengguna dalam memasukkan parameter dan akan lebih menghemat waktu. Hasil dari proses dekripsi adalah seperti format berkas awal dengan tambahan nama _rjn di akhir. Misalkan berkas asli bernama "contoh.txt", maka berkas hasil dekripsi akan bernama "contoh_rjn.txt" untuk membedakan dengan berkas asli.

Pengujian

Pengujian sistem dilakukan menggunakan metode *black box testing* untuk menguji fungsionalitas perangkat lunak. Hasil dari *black box testing* ditunjukkan oleh Tabel 3.

Tabel 3. Hasil Pengujian

No	Aktivitas Pengujian	Hasil yang Diharapkan	Kesimpulan
1	Memilih berkas yang akan dienkripsi menggunakan *button Browse*	Berkas berhasil dipilih dan *path* berkas serta ukuran berkas muncul di *form* yang disediakan	 Berkas berhasil dipilih dan path berkas serta ukuran berkas muncul di *form*
2	Memasukkan kunci untuk enkripsi	Kunci yang dimasukkan oleh user berhasil diinputkan pada form yang disediakan dalam tampilan simbol-simbol, serta panjang kunci dibatasi maksimal sesuai dengan ukuran panjang kunci yang dipilih pada option kunci.	 Kunci berhasil diinputkan dalam tampilan simbol dan panjang kunci sudah dibatasi sesuai dengan option yang dipilih
3	Memilih mode operasi *block cipher*	Pilihan mode operasi ECB, CBC, dan CFB muncul dalam *dropdown list* pada parameter mode operasi	 *Dropdown list* mode operasi memunculkan mode yang ECB, CBC, dan CFB
4	Memilih ukuran blok	Pilihan ukuran blok 128, 192 dan 256 muncul dalam *dropdown list* pada parameter ukuran blok	 *Dropdown list* ukuran blok memunculkan pilihan ukuran blok untuk proses enkripsi/dekripsi, yaitu 128, 192 dan 256
5	Melakukan proses enkripsi sesuai parameter panjang kunci, mode operasi, dan ukuran blok yang dipilih dengan menekan *button* Enkripsi	Proses enkripsi berhasil dilakukan dan waktu eksekusi ditampilkan di *form*	

No	Aktivitas Pengujian	Hasil yang Diharapkan	Kesimpulan
			Enkripsi berhasil dilakukan dan muncul informasi lamanya waktu eksekusi di *form*
		Berkas hasil enkripsi menjadi berformat .rjn dan berubah menjadi kode-kode tidak terbaca	 Berkas hasil enkripsi berubah menjadi kode-kode tidak terbaca
6	Memilih mode Dekripsi	*Option* Dekripsi terpilih dan parameter kunci, mode operasi, serta ukuran blok dalam kondisi *inactive*	 *Option* Dekripsi berhasil dipilih, *option* kunci dan parameter lainnya dalam keadaan *inactive*
7	Memilih berkas untuk didekripsi dengan menekan *button Browse*	Hanya berkas dengan tipe .rjn yang dapat dipilih	 Hanya berkas .rjn yang dapat dipilih untuk didekripsi
8	Melakukan proses dekripsi dengan membaca otomatis parameter panjang kunci, mode operasi, dan ukuran blok berdasarkan nama berkas dengan menekan *button* Dekripsi	Proses dekripsi berhasil dilakukan dan waktu eksekusi ditampilkan di *form*	 Dekripsi berhasil dilakukan dan muncul informasi lamanya waktu eksekusi di *form*
		Berkas hasil dekripsi berhasil dipulihkan dengan tambahan nama berkas _rjn di akhir	 Berkas hasil dekripsi berhasil dipulihkan

5. Kesimpulan

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan sebagaimana diuraikan di dalam bab sebelumnya, maka dapat disimpulkan bahwa perancangan aplikasi penyandian data menggunakan algoritma Rijndael berhasil dibangun menggunakan bahasa pemrograman C#. Hasil pengujian dengan metode *black box testing* menunjukkan bahwa aplikasi dapat berjalan dengan baik, dibuktikan dengan 100% fitur yang ada dapat berjalan sebagaimana mestinya tanpa ada *error*.

24% Overall Similarity

Top sources found in the following databases:

- 21% Internet database
- Crossref database
- 15% Submitted Works database
- 7% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	jurnal.uns.ac.id Internet	9%
2	text-id.123dok.com Internet	2%
3	adoc.pub Internet	<1%
4	scribd.com Internet	<1%
5	Sriwijaya University on 2021-03-15 Submitted works	<1%
6	Rohana Yola Parastika Hutasoit, Rahmaddeni Rahmaddeni, Erlin Erlin, ... Crossref	<1%
7	UIN Maulana Malik Ibrahim Malang on 2022-03-23 Submitted works	<1%
8	cybermovic.wordpress.com Internet	<1%

9	informatika.stei.itb.ac.id Internet	<1%
10	Sriwijaya University on 2019-05-13 Submitted works	<1%
11	docobook.com Internet	<1%
12	kriptografijaringan.blogspot.com Internet	<1%
13	opac.atmaluhur.ac.id Internet	<1%
14	zephyrnet.com Internet	<1%
15	Anriza Kurnia Aziiz, Magdalena A. Ineke Pakereng. "Perancangan Tekn... Crossref	<1%
16	ejournal.unkhair.ac.id Internet	<1%
17	repository.uin-suska.ac.id Internet	<1%
18	smartlib.umri.ac.id Internet	<1%
19	ajengjiwapangestu.wordpress.com Internet	<1%
20	pt.scribd.com Internet	<1%

21	repository.unmuhjember.ac.id Internet	<1%
22	sir.stikom.edu Internet	<1%
23	123dok.com Internet	<1%
24	I Gusti Agung Gede Arya Kadyanan. "Proteksi Short Message Service (... Crossref	<1%
25	Universitas Brawijaya on 2021-01-20 Submitted works	<1%
26	Universitas Negeri Jakarta on 2019-01-07 Submitted works	<1%
27	Universitas Sumatera Utara on 2021-12-06 Submitted works	<1%
28	begawe.unram.ac.id Internet	<1%
29	kelas9ablog.wordpress.com Internet	<1%
30	vaskoedo.wordpress.com Internet	<1%
31	dskon.com Internet	<1%
32	Sriwijaya University on 2021-12-27 Submitted works	<1%

33	Universitas Brawijaya on 2018-05-22 Submitted works	<1%
34	Universitas Brawijaya on 2018-07-20 Submitted works	<1%
35	Sriwijaya University on 2019-12-10 Submitted works	<1%
36	Universitas Brawijaya on 2017-05-31 Submitted works	<1%
37	core.ac.uk Internet	<1%
38	e-journal.uajy.ac.id Internet	<1%
39	Pahrizal Pahrizal, David Pratama. "IMPLEMENTASI ALGORITMA RSA U... Crossref	<1%
40	Sriwijaya University on 2020-11-11 Submitted works	<1%
41	Taufik Ramadan Firdaus. "Keamanan Aplikasi Web Melalui Penerapan ... Crossref	<1%

Excluded from Similarity Report

- Manually excluded text blocks

EXCLUDED TEXT BLOCKS

InformatikaInstitut Teknologi Telkom PurwokertoJl. D.I. Panjaitan No.128 Purwok...

jurnal.stkippgritulungagung.ac.id

1]Program Studi S1 Teknik

Novian Adi Prasetyo, Yudha Saintika. "Integration between Moodle and Academic Information System using...

LaptopLenovo Ideapad S410pProcessorIntel(R) Core™ i5

Universitas Brawijaya on 2018-07-12

8

forum.kaspersky.com

Tabel 2. Spesifikasi Perangkat

Universitas Brawijaya on 2017-01-10

Institute of Standards andTechnology (NIST

Sriwijaya University on 2020-11-11

proses enkripsi dan dekripsi

Sriwijaya University on 2022-04-05

P) = C(2.1)dan fungsi dekripsi D

text-id.123dok.com

Tabel 2 berikut:Tabel

Universitas Brawijaya on 2017-01-10

Microsoft Visual Studio Enterprise 2017 Version 15.9

github.com

4200U

repositori.uin-alauddin.ac.id

3. Metode PenelitianPenelitian ini bertujuan untuk merancang dan membangun

123dok.com

Perancangan SistemPada perancangan sistem

Universitas Brawijaya on 2018-07-20

KesimpulanBerdasarkan hasil

123dok.com

pihak-pihak yang berkomunikasi maupun

repository.uin-suska.ac.id

penyangkalan (non-repudiation

jurusan.tik.pnj.ac.id

untuk

jurnal.uns.ac.id

dalam bentuk bit tunggal, dimana rangkaian bit

Universitas Brawijaya on 2018-12-17

tidak dapat diketahui oleh pihak yang tidak memiliki otoritas untuk

www.kaskus.co.id

misalkan P menyatakan plaintext, C menyatakan ciphertext

Sriwijaya University on 2021-11-26

Pada penelitian ini

Sriwijaya University on 2021-03-15

Secara umum proses enkripsi dan dekripsi

repository.uksw.edu

C) = P(2.2

Universitas Brawijaya on 2017-02-14

AbstrakAlgoritma Rijndael

jurnal.uns.ac.id

mode operasi dalam block cipher

informatika.stei.itb.ac.id