Jurnal Review Pendidikan dan Pengajaran
http://journal.universitaspahlawan.ac.id/index.php/jrpp
Volume 7 Nomor1, 2024
P-2655-710X e-ISSN 2655-6022

Submitted : 28/01/2024
Reviewed : 02/02/2024
Accepted : 06/02/2024
Published : 08/02/2024

John Swatrahadi Permana [1]
Hendrana [2]
Rinaldi Munir[3]

MENGAMANKAN FILE RAHASIA MENGGUNAKAN HYBRID KRIPTOGRAFI

Abstrak
Penyelenggara sistem elektronik mengelola sitem aplikasi mengelola file dari pengguna yang sifatnya rahasia. File yang bersifat rahasia yang dikelola menggunakan suatu sistem informasi perlu dilindungi dari ancaman kebocoran data. Salah satu metode yang dapat digunakan untuk melindungi file tersebut adalah menggunakan metode hybrid cryptography yaitu algoritma Adcanced Enryption Standart atau AES dikombinasikan dengan Rivest Shamir Adleman atau RSA. File rahasia yang diupload ke sistem aplikasi sebelum disimpan file tersebut dienkripsi menggunakan kriptografi kunci simetri AES, dengan kunci yang degenerate oleh sistem. Kunci yang degenerate oleh sistem tersebut sifatnya rahasia yang perlu diamankan. Untuk mengamankan kunci dari file rahasia yang disimpan pada database diamankan Kembali menggunakan kriptografi kunci asimetris Rivest Shamir Adlemen. Hasil penelitian menunjukkan waktu yang diperlukan untuk mengamankan file rahasia cukup singkat sehingga tidak mempengaruhi kinerja sistem secara keseluruhan. Proses enkripsi maupun dekripsi hybrid kriptografi AES dan RSA memerlukan waktu rata-rata sebesar 295,8774 bytes / micro second. Proses enkripsi lebih cepat 0,633537 kali dibandingkan dengan proses dekripsi. Metode hybrid kriptografi AES dan RSA cukup kuat untuk melindungi file berklasifikasi rahasia.
Kata Kunci: Hybrid Kriptografi, AES, RSA

Abstract
Electronic system administrators manage application systems and manage files from users that are confidential in nature. Confidential files that are managed using an information system need to be protected from the threat of data leaks. One method that can be used to protect these files is to use a hybrid cryptography method, namely the Advanced Encryption Standard or AES algorithm combined with Rivest Shamir Adleman or RSA. Secret files that are uploaded to the application system before being saved are encrypted using AES symmetric key cryptography, with the key being degenerated by the system. The key degenerated by the system is a secret that needs to be secured. To secure the key of the secret file stored in the database, it is secured again using Rivest Shamir Adlemen's asymmetric key cryptography. The research results show that the time required to secure confidential files is short enough that it does not affect overall system performance. The encryption and decryption process of hybrid AES and RSA cryptography requires an average time of 295.8774 bytes / microsecond. The encryption process is 0.633537 times faster than the decryption process. The hybrid method of AES and RSA cryptography is strong enough to protect confidential classified files.
Keywords: Hybrid Cryptography, AES, RSA

PENDAHULUAN

Penyelenggara Sistem Elektronik (PSE) adalah setiap orang, penyelenggara negara, badan usaha, dan masyarakat yang menyediakan, mengelola, dan/atau mengoperasikan sistem elektronik secara sendiri-sendiri maupun bersama-sama kepada pengguna sistem elektronik untuk keperluan dirinya dan/atau keperluan pihak lain(Peraturan Pemerintah No 71 Tahun 2019 Tentang Penyelenggaraan Sistem Dan Transaksi Elektronik, 2019). Sistem elektronik mengandalkan database untuk pengelolaan data yang terstruktur. Sedangkan untuk data-data yang tidak terstruktur biasanya disimpan dalam bentuk dokumen file seperti Microsoft word atau pdf. Kedua file tersebut umum digunakan dalam menuangkan informasi yang tidak

[1,2,3)] Universitas Pertahahan Republik Indonesia
e-mail: john.permana@idu.ac.id, hendranatj@gmail.com, rinaldi@staff.stei.itb.ac.id

terstruktur dalam bentuk dokumen softfile. Terkadang file tersebut mengandung informasi yang sensitif ataupun rahasia, hal tersebut menjadi incaran peretas yang ingin memanfaatkannya baik untuk motivasi ekonomi, sabotase dan lainnya.

Kriptografi Enkripsi adalah suatu proses mengubah sebuah teks murni (plaintext) menjadi sebuah runtutan karakter atau data yang terlihat tidak berarti dan mempunyai urutan bit yang tidak beraturan, disebut ciphertext. Proses pengubahan kembali ciphertext menjadi plaintext disebut dekripsi.

Keunggulan algoritma AES atau Rijndael adalah rumitnya chipper text dipecahkan menjadi plain text. Selain itu AES Rijndael memiliki daya memori dan kecepatan komputasi sehingga banyak diminati pasar (Nurnaningsih & Permana, 2018). Algoritma AES dapat diimplementasikan pada proses enkripsi dan dekripsi dokumen dengan waktu enkripsi 0,212 second untuk dokumen ukuran 19,212 Kb dan 20,533 second untuk dokumen ukuran 1.966 Kb sedangkan pada proses dekripsi membutuhkan waktu 0,213 second untuk dokumen ukuran 19,212 Kb dan 20,882 second untuk dokumen dengan ukuran 1.966 Kb (Widodo & Purnomo, 2020). Kecepatan mengenkripsi dan mendekripsi data menggunakan algoritma AES lebih cepat dibanding tiga algoritma lainnya seperti Blowfish, DES, dan IDEA. Proses enkripsi paling cepat 48% dan proses dekripsinya adalah 45% (Meko, 2018). Proses penyimpanan dan pertukaran informasi dalam bentuk file docx, xlsx, pdf menjadi aman jika dienkripsi menggunakan algoritma AES dikombinasikan dengan One-Time-Password atau OTP (Indra Nugraha et al., 2018). File yang dienkripsi menggunakan algoritma AES akan berubah bentuk menjadi file yang tidak bisa dibaca, file tersebut dapat kembali kebentuk aslinya jika didekripsi dengan algortima AES dan menggunakan kunci yang benar sama pada saat dienkripsi (Muharram, 2018). Kelemahan AES kalau persamaan matematis yang mendasarinya berhasil dipecahkan maka algoritma AES dapat ditembus yang mengakibatkan pertahanan menjadi hancur (Asriyanik, 2017).

RSA berasal dari kata penemu algoritma itu sendiri yaitu Ron Rivest, Adi Shamir, dan Leonard Adleman dari Massachusetts Institute of Technology (MIT). RSA merupakan satu algoritma asimetrik yang mempunyai dua kunci yang berbeda, disebut pasangan kunci (key pair) yang punya hubungan matematis untuk proses enkripsi dan dekripsi. Algoritma RSA mempunyai cara enkripsi yang mudah, tetapi data yang sudah terenkripsi sulit untuk dibobol jika hanya punya kunci publiknya saja tanpa kunci privat. RSA selain dipakai pada software diantaranya untuk proses autentikasi data dan tanda tangan digital, juga dipakai pada perangkat keras, telepon anti penyadapan, kartu jaringan ethernet, dan kartu cerdas (Andy Wicaksono -, 2008). Proses enkripsi pesan RSA (Munir, 2018)sebagai berikut :

1. Nyatakan pesan menjadi blok-blok plainteks: m1 , m2 , m3 , … (syarat: $0 < mi < n - 1$).
2. Hitung blok cipherteks ci untuk blok plainteks pi dengan persamaan $ci = mi\ e \bmod n$

yang dalam hal ini, e adalah kunci publik. Sedangkan untuk proses dekripsinya adalah sebagai berikut: Proses dekripsi dilakukan dengan menggunakan persamaan $mi = ci\ d \bmod n$,
dimana, d adalah kunci privat.

Untuk memudahkan memberi gambaran kerja algoritma RSA misalnya plainteks berikut :
M = ‘HARI INI’
ASCII: 7265827332737873
Pecah M menjadi blok yang 3 digit:
m1 = 726
m2 = 582
m3 = 733
m4 = 273
m5 = 787
m6 = 003
(mi terletak antara 0 sampai n – 1 = 3337)
Enkripsi setiap blok:
c1 = 72679 mod 3337 = 215
c2 = 58279 mod 3337 = 776
dst Hasilnya :
C = 215 776 1743 933 1731 158.
Dekripsi (menggunakan kunci privat d = 1019) maka
m1 = 2151019 mod 3337 = 726

m2 =7761019 mod 3337 = 582
dst untuk sisi blok lainnya Plainteks
M = 7265827332737873
yang dalam ASCII adalah 'HARI INI'. Dalam praktek, RSA tidak digunakan untuk mengenkripsi pesan, tetapi mengenkripsi kunci simetri (kunci sesi) dengan kunci publik penerima pesan. Sedangkan pesannya dienkripsi menggunakan algoritma AES.

Dengan adanya kelemahan dan kelebihan dari kriptografi simetris dan asimetris, kedua metode bisa dioptimalkan dengan mengkombinasikan kedua metode kriptografi tersebut yaitu Hybrid Cryptography. Algoritma RSA kunci publik digunakan untuk mengkenkripsi kunci simetri supaya kunci simetri tersebut bisa dikirimkan melalui saluran komunikasi. Algoritma kriptografi hybrid dapat digunakan untuk mengamankan properti rahasia aplikasi (William, 2023).

PHP HyperText PreProcessor adalah bahasa pemrograman open source yang bekerja pada sisi server-side yang dapat diakses oleh client nya menggunakan browser. PHP merupakan Bahasa pemrogaman web yang banyak manfaatnya, untuk menjalankannya memerlukan web server seperti IIS, apache, nginx atau xitami pada sisi server dan browser seperti chrome, Microsoft edge, dll pada sisi client. Beberapa contoh apliasi yang menggunakan Bahasa pemrograman PHP yaitu Sistem mempercepat penyampaian informasi akademik (Tumini & Fitria, 2021), website SMA Al-Mukhtariyah untuk mendapatkan file materi sekolah dengan fitur download (Suhartini et al., 2020), pengambilan keputusan penerimaan mahasiswa baru LP3I (Sahi, 2020).

MYSQL adalah database management system atau DBMS yang multithread, multi-user gratis. MySQL dimiliki dan disponsori oleh sebuah perusahaan asal Swedia MySQL AB. beberapa kelebihan database mysql (Chillia Furda Chudhrotus, 2017) diantaranya adalah sebagai berikut :

1. MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows atua Linux,
2. Bersifat Open Source atau gratis
3. Bisa multiuser
4. Kecepatannya cukup baik untuk memproses SQL per detik
5. keamanan datanya terdiri beberapa lapis seperti level subnet mask, nama host, dan izin akses user yang detail dan password yang terenkripsi.

Tujuan penelitian ini adalah untuk melakukan pengujian penerapan kriptografi hybrid AES 256 dan RSA pada file berklasifikasi rahasia. Penelitian ini dilakukan untuk memperoleh keyakinan keberhasilan penerapan metode kriptografi hybrid untuk menjaga kerahasiaan file yang dikelola atau disimpan pada server aplikasi. Penelitian ini dilakukan juga untuk menghitung kecepatan proses enkripsi kriptografi hybrid begitu juga sebeliknya proses dekripsi kriptografi hybrid.

Ruang lingkup penelitian ini adalah penerapan kriptogafi hybrid pada object file rahasia dengan format pdf dan Microsoft office word. Kriptografi hybrid yang digunakan adalah algoritma AES 256 dan RSA. Algoritma AES 256 digunakan untuk mengenkripsi file rahasia dalam format PDF dan Microsoft word, kemudian kunci untuk mendekripsi file tersebut diamankan Kembali menggunakan algoritma RSA. pada saat proses enkripsi dan dekripsi dicatat atau disimpan waktunya pada database untuk analisis kecepatan kinerja algoritma.

METODE

Permasalahan dalam penelitian ini adalah bagaiamana caranya untuk mengamankan dokumen rahasia agar dokumen rahasia yang disimpan di server tidak dapat dibaca oleh pihak lain tanpa menggunakan aplikasi yang sudah dibuat. suatu sistem aplikasi terkadang mengharuskan menyimpan file dokumen rahasia di dalam server. Biasanya file tersebut diupload oleh pengguna melalui suatu sistem aplikasi. namun kebanyakan file tersebut dibiarkan begitu saja tanpa dilindungi kerahasiaannya. Selain usaha yang diperlukan untuk mengembangan teknik untuk mengamankan file rahasia tersebut terkadang juga pengguna merasa aplikasi akan semakin lambat apabila diterapkan kriptografi padanya, padahal tingkat kelambatan tersebut belum dirasakan apakah betul-betul lambat atau malahan tidak terasa jika sedang dilakukan proses kriptografi.

Metode penelitian ini menggunakan metode kuantitatif dengan data kontinum interval. Metode kuantitatif digunakan karena menggunakna data berupa angka-angka dan analisis statistitik. Angka-angka diperoleh dari hasil pengukuran kecepatan proses enkripsi file dokumen dan dekripsi file dokumen. Pengukuran kecepatan diperoleh dengan mengembangkan sistem aplikasi sederhana secara rapid prototyping yang merupakan pengguna metode urutan ke-2 atau 32 % penelitian bidang computer science (Rinanto et al., 2017) dengan menggunakan bahasa pemrograman php.

Beberapa kode program yang digunakan untuk mengenkripsi dan mendekripsi dituliskan seperti dibawah ini :

```
function encrypt($data, $private_key_file_name = "id_rsa.pub")
{
 // $data -> (string, file)
 // $private_key_file_name -> (string)
 // Get the public key
 $private_key = openssl_get_publickey(file_get_contents($private_key_file_name));
 // Generate a random secret key
 $secretKey = openssl_random_pseudo_bytes(16);
 // Encrypt the text
 $iv = openssl_random_pseudo_bytes(16);
 $encrypted_data = openssl_encrypt($data, "aes-256-cbc", $secretKey, 0, $iv);
 $dataandiv = base64_encode($encrypted_data) . "::" . base64_encode($iv);
 // encrypt the secret key with RSA
 openssl_public_encrypt($secretKey, $encrypted_key, $private_key);
 return [
  "dataandiv" => $dataandiv,
  "aes_key" => $encrypted_key,
  "rsa_key" => $private_key_file_name
 ];
}
```

Untuk fungsi mendekripsi file ditulis dalam kode program php seperti di bawah ini :

```
function decrypt ($dataandiv, $public_key_file_name, $encrypted_secret_aes_key, $mysqli)
{
 $time_start = microtime(true);
 // Remove .pub since private key has the same file name with the public key
 $private_key_file = file_get_contents(str_replace(".pub", "", $public_key_file_name));
 $private_key = openssl_get_privatekey($private_key_file);
 // Decrypt the secret key from rsa
openssl_private_decrypt($encrypted_secret_aes_key, $decrypted_secret_aes_key, $private_key);
 // Seperate IV and encrypted text
 $iv = base64_decode(explode("::", $dataandiv)[1]);
 $encrypted_data = base64_decode(explode("::", $dataandiv)[0]);
 // Decrypt the text from aes
 $decrypt = openssl_decrypt($encrypted_data, "aes-256-cbc", $decrypted_secret_aes_key, 0, $iv);
 $time_end = microtime(true);
 $time_taken = ($time_end - $time_start);
 $time_taken = number_format($time_taken, 6,'.', '');
 $mysqli->query('UPDATE app SET time_taken_decrypt = ' . $time_taken . ' WHERE id = ' . $_GET['id']);
 return $decrypt;
}
```

Kriptografi hybrid ini menggunakan algoritma AES untuk mengenkripsi file rahasia. kemudian untuk menjaga kerahasiaan kuncinya dienkripsi Kembali menggunakan algoritma RSA. Penggabungan kedua model algoritma AES yang termasuk kriptografi kunci simetri

dengan algoritma RSA yang termasuk kriptografi kunci asimetri diharapkan dapat digunakan untuk mengamankan file dokumen rahasia dengan tetap menjaga kenyamanan pengguna dalam mengupload file.

Algoritma RSA untuk proses enkripsinya menggunakan persamaan matematis (1) sedangkan proses dekripsi menggununakan persamaan matematis (2) (Munir, 2018) seperti berikut ini :

Enkripsi AES dengan rumus : $C = m^e \text{ (Mod n)}$ (1)

Dekripsi AES dengan rumus : $M = C^d \text{ (Mod n)}$ (2)

Keterangan :

C : Chiphertext,

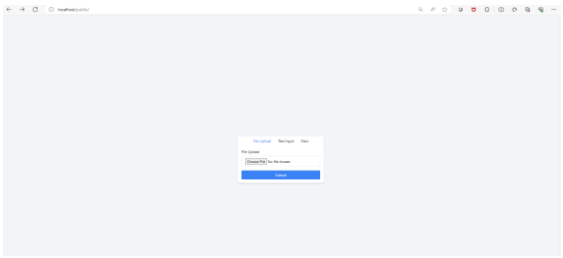
M : Message

HASIL DAN PEMBAHASAN

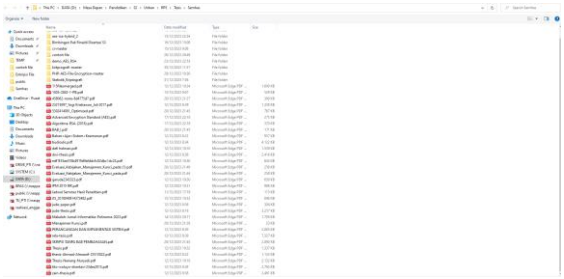
Untuk mencapai tujuan penelitian, dilakukan pengembangan perangkat lunak sederhana. Perangkat lunak yang dikembangkan dengan fungsional utamanya adalah mengupload file, mengenkripsi file, menampilkan daftar file yang telah di enkripsi dan mendekripsi file. Selain itu perangkat lunak yang dikembangkan merupakan aplikasi web yang dapat menghitung kecepatan proses enkripsi dan dekripsi.

Enkripsi

Fungsionalitas pertama adalah mengupload file rahasia yang bertipe pdf dan word saja yang bisa diupload ke server. Kemudian dienkripsi secara otomatis oleh sistem sebelum disimpan pada server seperti yang ditunjukkan pada gambar 1 dan gambar 2 dibawah ini. Pada penelitian ini file yang dapat diupload hanya satu file saja karena tujuan sistem aplikasi dikembangkan untuk mensimulasikan hybrid kriptografi saja.

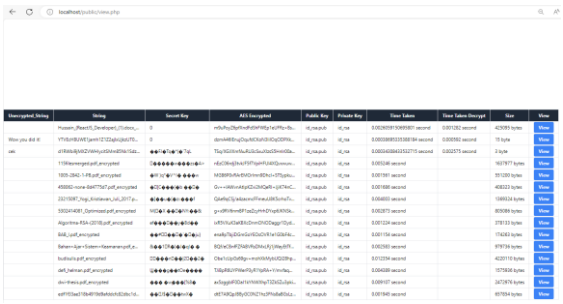


Gambar 1 Antar Muka Upload File



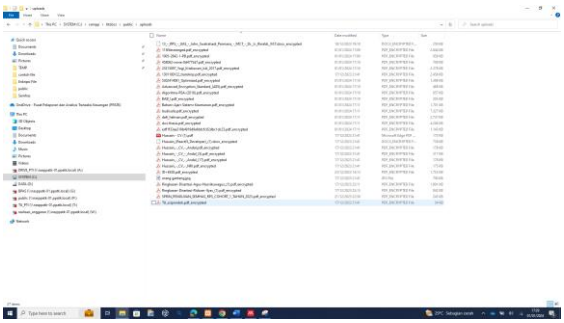
Gambar 2 Memilih file Rahasia Untuk di Upload

Daftar file yang sudah dienkripsi disimpan pada database, untuk simulasi ditampilkan seluruh property hybrid kriptografi. Pertama adalah nama file, kemudian kunci rahasia AES 256 kemudian selanjutnya adalah kunci rahasia AES 256 yang sudah dienkripsi dengan algoritma RSA. Publik Key dan private key RSA ditampilkan nama kuncinya saja. Selanjutnya adalah waktu yang diperlukan untuk enkripsi dan waktu yang diperlukan untuk dekripsi. Satu kolom sebelum terakhir button view untuk mendekripsi adalah kolom ukuran file. Untuk mendapatkan gambaran secara garis besar aplikasinya seperti pada gambar 3 di bawah ini :



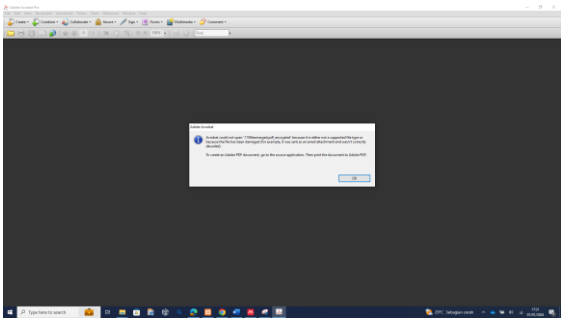
Gambar 3 Daftar file yang sudah di enkripsi

Selanjutnya setelah mengupload beberapa file untuk dienkripsi mencoba mengakses file yang sudah diupload ke folder uploads pada server localhost seperti pada gambar 4, dalam penelitian ini file yang diupload dibatasi hanya dalam format ms word dan pdf saja.



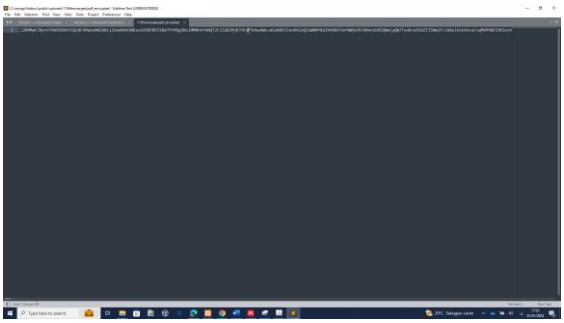
Gambar 4 File pada Folder Upload Server

Apabila server telah dikuasa oleh peretas dan peretas dapat mengakses file rahasia maka file tersebut maka peretas akan mencari file yang berharga. Namun dengan setelah dilakukan enkripsi menggunakan AES dan RSA file tersebut tidak akan dapat dibaca seperti pada tampilan gambar 5 di bawah ini.



Gambar 5 File Encripted Dibaca Menggunakan Adode PDF reader

Tidak bisa dibuka menggunakan adobe PDF reader maka coba dibaca dengan editor text yang lainnya. Dan hasil yang didapat juga akan sama yaitu tidak akan dapat dibaca file asli yang sebenarnya seperti pada gambar 6 di bawah ini :



Gambar 6 File Encrypted Dibuka Menggunakan Sublime Text

Untuk mengetahui kinerja algoritma hybrid kriptografi dapat dilakukan dengan melakukan percobaan beberapa kali enkripsi file dengan 2 jenis file yang berbeda yaitu Microsoft word dan pdf. Pada saat melakukan percobaan sekaligus juga dapat dicatat kecepatan rata proses enkripsinya dan dekripsi filenya. Dengan demikian dapat diketahui kinerja penggunaaan algoritma AES dan RSA pada beberapa kali percobaan. Hasil percobaan yang berhasil dicatat adalah sebagai berikut :

Table 1 Data Kecepatan Proses Enkripsi File format Pdf

No.	Ukuran File (bytes)	Waktu Enkripsi (ms)	Kecepatan (bytes / ms)
1.	1637977	5246	312,2335
2.	551200	1561	353,107
3.	408323	1686	242,1845
4.	1369324	4003	342,0744
5.	805086	2873	280,2249
6.	378133	1224	308,9322
7.	174263	1154	151,0078
8.	979736	2583	379,3016
9.	4220110	12354	341,5987
10.	1575936	4389	359,0649
11.	2472976	9107	271,5467
12.	657854	1945	338,2283

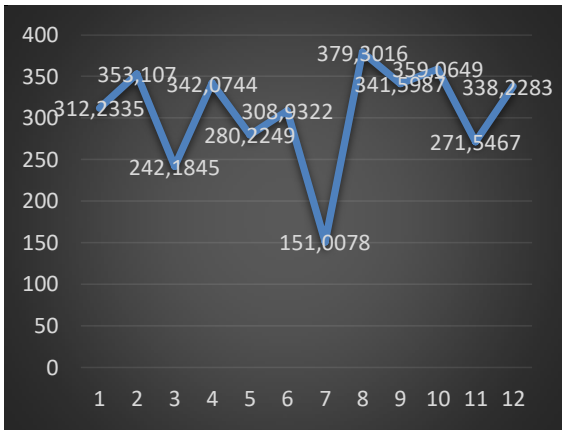
Dari tabel 1 diatas dapat dihitung rata-rata proses mengenkripsi file yang diupload melalui aplikasi web adalah sebesar : 306,6254 bytes / millisecond. Sedangkan untuk dokumen rahasia format Microsoft word waktu enkripsinya seperti di bawah tabel 2 berikut ini :

Tabel 2 Data Kecepatan Proses Enkripsi File format Ms Word

No.	Ukuran File (bytes)	Waktu Enkripsi (ms)	Kecepatan (bytes / ms)
1	139598	2687	51,95311
2	133707	1260	106,1167
3	138496	1030	134,4621
4	139028	840	165,5095
5	185827	916	202,8679
6	54724	916	59,74236
7	57566	1033	55,72701
8	61016	1052	58
9	275739	1048	263,1097
10	226941	1018	222,9283
11	138527	825	167,9115
12	398583	3610	110,4108
13	6170698	16147	382,1576

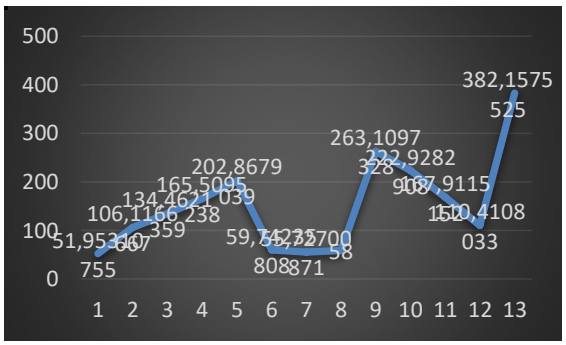
Dari tabel 2 diatas dapat dihitung rata-rata proses mengenkripsi file yang diupload melalui aplikasi web adalah sebesar : 152,3767 bytes / millisecond.

Kemudian dari tabel 1 untuk memudahkan visualisasi ditampilkan dalam bentuk grafik yang menunjukkan proses enkripsi file dengan format Pdf berfluktuasi naik turun. Pada percobaan ke-1 sampai dengan ke-6 kecepatan proses enkripsi naek turun dan terjadi puncak penurunan kecepatan yang drastis pada percobaan ke-7. Pada percobaan ke-8 sampai dengan ke-12 kecepatan enkripsi berfluktuasi naik dan turun.



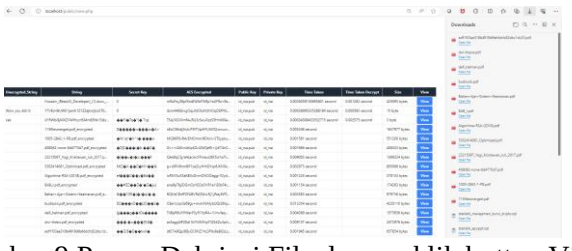
Gambar 7 Grafik Kecepatan Proses Enkripsi file Pdf

Kecepatan untuk file dengan format Microsoft word dengan 13 kali percobaan kecepatan enkripsi cenderung semakin meningkat seperti ditampilkan pada Grafik 8 di bawah. Pada percobaan ke-1 s.d percobaan ke- 5 kecepatan cederung naik, kemudian pada percobaan ke-6 sampai dengan ke-8 cukup stagnan. Penurunan kecepatan terjadi pada percobaan ke-9 sampai dengan 12 akan tetapi meningkat drastic pada percobaan ke-13.



Gambar 8 Grafik Kecepatan Proses Enkripsi file Word

Dekripsi



Gambar 9 Proses Dekripsi File dengan klik button View

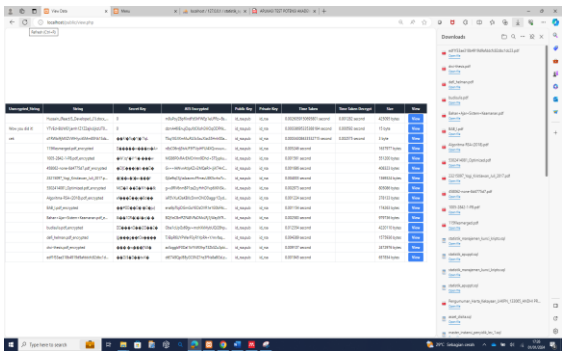
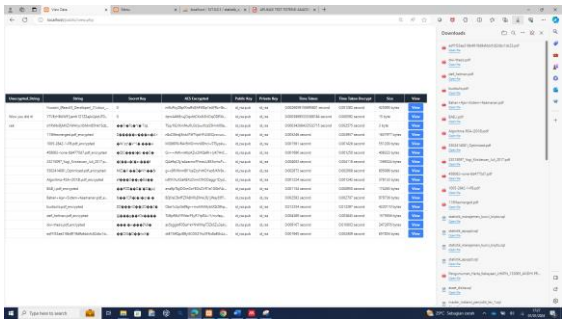
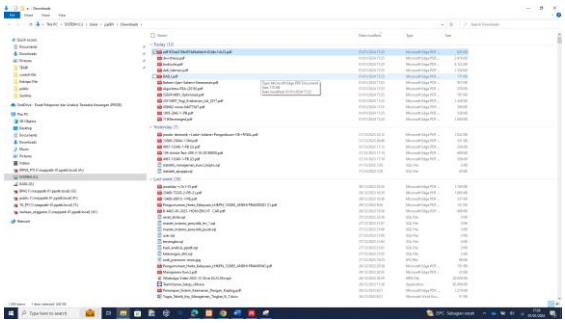


Figure 10 Klik Refresh Browser Untuk Munculkan Perhitungan Waktu Dekripsi



Gambar 11 Hasil Perhitungan Waktu Dekripsi Muncul



Gambar 12 File Hasil Dekripsi

Untuk mengukur kecepatan rata-rata proses mendekripsi file dengan menggunakan algoritma AES dan RSA ini telah dilakukan beberapa kali percobaan. Pertama mendekripsi file dengan format Pdf. Selanjutnya mendekripsi file dengan format word. hasil dari kedua percobaan mendekripsi file dicatat pada tabel 3 untuk dekripsi file Pdf dan tabel 4 untuk dekripsi file word. tabel tersebut adalah sebagai berikut :

Table 3 Table 3 Data Kecepatan Proses Dekripsi File Pdf

No	Ukuran File (bytes)	Waktu Dekripsi (ms)	Kecepatan (bytes / ms)
1.	1637977	3957	413,9441
2.	551200	1428	385,9944
3.	408323	1250	326,6584
4.	1369324	4118	332,5216
5.	805086	2008	400,9392
6.	378133	1242	304,4549
7.	174263	950	183,4347
8.	979736	2707	361,9269
9.	4220110	12397	340,4138
10.	1575936	3843	410,0796
11.	2472976	10003	247,2234
12.	657854	2409	273,0818

dari tabel 3 dengan melakukan 12 kali percobaan dapat diperoleh nilai rata-rata kecepatan proses mendekripsi file dengan format Pdf adalah sebesar : 331,7227 bytes tiap millisecond.

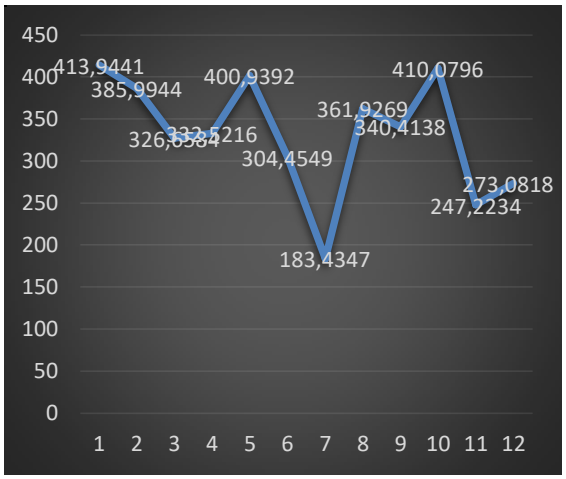
Table 4 Data Kecepatan Proses Dekripsi File Word

No	Ukuran File (bytes)	Waktu Dekripsi (ms)	Kecepatan (bytes / ms)
1	139598	1195	116,8184
2	133707	2417	55,3194
3	138496	1798	77,02781

4	139028	2322	59,87425
5	185827	3725	49,88644
6	54724	1176	46,53401
7	57566	1584	36,34217
8	61016	1040	58,66923
9	275739	1300	212,1069
10.	226941	1394	162,7984
11.	138527	1791	77,34618
12.	398583	1400	284,7021
13.	6170698	1595	3868,776

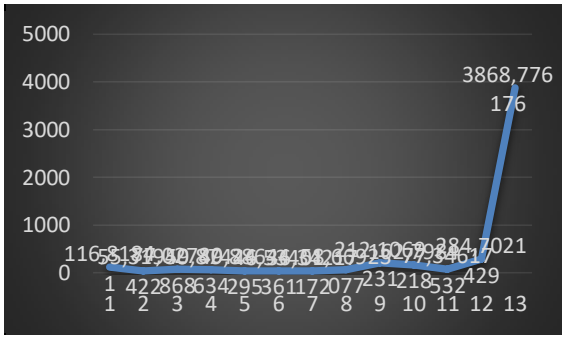
Dari tabel 4 diatas menunjukkan bahwa dengan melakukan 13 kali percobaan dapat diperoleh nilai rata-rata kecepatan proses mendekripsi file dengan format Microsoft office word adalah sebesar 392,7847 bytes tiap mikro second. Kecepatan paling rendah pada percobaan ke-2 yaitu sebesar 55,3194 bytes tiap Micro second dengan ukuran file 133707. Akan tetapi kecepatan paling tinggi justru pada file yang paling besar yaitu file word dengan ukuran 6170698 bytes di proses dengan kecepatan sebesar 3868,776 bytes tiap Micro second.

Dari tabel 3 dapat digambarkan dalam bentuk grafik yang hasilnya ternyata fluktuatif seperti pada gambar 13. Penurunan kecepatan sangat tajam terjadi pada percobaan ke- 7. Sedangkan pada percobaan yang lain proses mendekripsi file Pdf sangat dinamis baik kenaikan atau penurunan kecepatannya.



Gambar 13 Grafik Kecepatan Proses Dekripsi File Format Pdf

Sedangkan pada tabel 4 proses mendekripsi file word dapat digambarkan dalam bentuk grafik pada gambar 14 yang hasilnya ternyata hampir merata hanya pada satu percobaan terakhir saja yang meningkat signifikan. Kenaikan kecepatan sangat tajam terjadi pada percobaan yang terakhir. Sedangkan pada percobaan yang lain sangat stabil merata kecepatannya.



Gambar 14 Grafik Kecepatan Proses Dekripsi File Format Word

Data kecepatan hasil pengujian proses enkripsi dan dekripsi file rahasia dengan jenis file Microsoft office word dan pdf dapat dilihat pada tabel 5 dalam satuan bytes / micro second :

Table 5 Rata-rata Kecepatan Proses Enkripsi dan Dekripsi

Format File	Enkripsi	Dekripsi
Pdf	306,6254	331,7227
Word	152,3767	392,7847

SIMPULAN

Metode hybrid kriptografi dapat diterapkan untuk mengenkripsi dokumen rahasia yang disimpan pada server melalui suatu sistem aplikasi. hybrid kriptografi yang dapat digunakan untuk mengenkripsi file dokumen rahasia adalah Advances Encryption Standard atau AES dan Rivest Samir Adleman atau RSA. Algoritma AES cukup kuat untuk digunakan untuk mengenkripsi file dokumen yang sifatnya rahasia misalanya dalam format Pdf dan Microsoft word. Untuk memperkuat kunci enkripsinya dapat dienkripsi Kembali dengan menggunakan kriptografi kunci asimetri yaitu RSA yang menggunakan kunci public untuk menenkripsi dan kunci private untuk mendekripsi file.

Kinerja Algoritma AES dan RSA sangat cepat sehingga tidak mempengengaruhi kinerja sistem aplikasi secara signifikan. Secara keseluruhan proses enkripsi maupun dekripsi hybrid kriptgrafi AES dan RSA memerlukan waktu rata-rata sebesar 295,8774 bytes / microsecond. Proses enkripsi lebih cepat 0,633537 kali dibandingkan dengan proses dekripsi. kecepatan rata-rata untuk proses enkripsi sebesar 229,5011 bytes / micro second. sedangkan kecepatan rata-rata proses dekripsi sebesar 362,2537 bytes / microsecond.

DAFTAR PUSTAKA

Andy Wicaksono -, P. (2008). Enkripsi Menggunakan Algoritma RSA.

Asriyanik. (2017). Studi Terhadap Advanced Encryption Standard AES Dan Algoritma Knapsack Dalam Pengamanan Data (Vol. 7, Issue 1).

Chillia Furda Chudhrotus. (2017). Sistem Pendukung Keputusan Pemilihan Sistem Berprestasi di SMK Muhammadiyah 1 Lamongan dengan metode AHP dan Topsis. Universitas Muhammadiyah Gresik.

Indra Nugraha, A., Kusumaningsih, D., & Alawy, M. (2018). Enkripsi File Dokumen AES PT. MNC. Telematika MKOM, 10(1).

Meko, D. A. (2018). Perbandingan Algoritma DES, AES, IDEA Dan Blowfish dalam Enkripsi dan Dekripsi Data. Jurnal Teknologi Terpadu, 4(1).

Muharram, F. (2018). Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard. Prosiding Seminar Nasional Ilmu Komputer Dan Teknologi Informasi, 3(2).

Munir, R. (2018). Algoritma RSA. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2017-2018/Algoritma-RSA-(2018).pdf

Nurnaningsih, D., & Permana, A. A. (2018). Rancangan Aplikasi Pengaman Data Dengan Algoritma Advanced Encryption Standard AES. JURNAL TEKNIK INFORMATIKA, 11(2), 177–186. https://doi.org/10.15408/jti.v11i2.7811

Peraturan Pemerintah No 71 Tahun 2019 Tentang Penyelenggaraan Sistem dan Transaksi Elektronik, (2019).

Rinanto, A., Sutopo, W., Mojo No, J., & Karangasem Kec Laweyan, K. (2017). Perkembangan Teknologi Rapid Prototyping: Study Literatur. In Jurnal Metris (Vol. 18). http://ojs.atmajaya.ac.id/index.php/metris

Sahi, A. (2020). Aplikasi Test Potensi Akademik Seleksi Saringan Masuk LP3I Berbasis Web Online Menggunakan Framework CodeIgniter (Vol. 7, Issue 1). http://www.php.net.

Suhartini, Sadali, M., & Putra, Y. K. (2020). Sistem Informasi Berbasis Web SMA Al-Mukhtariyah Mamben Lauk Berbasis PHP Dan Mysql. Infotek : Jurnal Informatika Dan Teknologi, Vol. 3(No. 1,), 79–83.

Tumini, & Fitria, M. (2021). Penerapan Metode Scrum pada E-Learning STMIK CIKARANG. Jurnal Informatika SIMANTIK, 6(1). https://www.simantik.panca-sakti.ac.id

Widodo, B. E., & Purnomo, A. S. (2020). Implementasi Advanced Encryption Standard pada Enkripsi Dokumen Rahasia DitIntelkam Polda DIY. Jurnal Teknik Informatika (Jutif), 1(2), 69–77. https://doi.org/10.20884/1.jutif.2020.1.2.21

William, C. J. (2023). Implementasi Kriptografi Hybrid dalam Pengelolaan Properti Rahasia pada Aplikasi Spring Boot. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/Makalah2/Makalah2-Kriptografi-2023%20(13).pdf