Implementasi *Hybrid* Kriptografi *Diffie Hellman* dan *ChaCha20* Pada Keamanan *File* Dokumen

1st Nikko Yudha Asmara Adi
Program Studi Informatika
Universitas Telkom, Kampus Surabaya
Surabaya, Indonesia
nikkoyudha@student.telkomuniversity.ac.id

2nd Rizky Fenaldo Maulana
Program Studi Informatika
Universitas Telkom, Kampus Surabaya
Surabaya, Indonesia
rizkyfenaldo@telkomuniversity.ac.id

3rd Fandisya Rahman
Program Studi Informatika
Universitas Telkom, Kampus Surabaya
Surabaya, Indonesia
fandisya@telkomuniversity.ac.id

Abstrak — Keamanan data menjadi isu krusial di era digital, terutama bagi dokumen elektronik yang sering menjadi target serangan siber. Penelitian ini bertujuan mengembangkan sistem pengamanan *file* dokumen dengan masukan berupa *file* teks dan keluaran berupa *file* terenkripsi yang aman dari intersepsi pihak tidak sah. Meskipun algoritma seperti *AES* telah banyak digunakan, tantangan dalam pengelolaan kunci dan kebutuhan akan efisiensi tinggi di perangkat tanpa akselerasi *hardware* masih menjadi hambatan. Saat ini, sistem belum mampu menggabungkan kecepatan dan keamanan secara optimal, khususnya untuk *file* dengan berbagai ukuran. Penelitian ini mengusulkan metode *hybrid* kriptografi yang menggabungkan *Diffie–Hellman* sebagai algoritma pembentukan kunci rahasia dan *ChaCha20* sebagai *stream cipher* untuk enkripsi data. Sistem ini memanfaatkan keunggulan *Diffie–Hellman* dalam pertukaran kunci aman serta efisiensi tinggi dari *ChaCha20* dalam proses enkripsi dan dekripsi. Hasil menunjukkan bahwa kombinasi *Diffie-Hellman–ChaCha20* memiliki waktu enkripsi tercepat (0.1906 detik) dan *throughput* tertinggi (13651.4704 KB/s). Sistem ini terbukti unggul dalam kecepatan, efisiensi, dan ketahanan terhadap skenario *file* berukuran besar, menjadi solusi optimal dalam pengamanan *file* digital.

***Kata kunci*— *ChaCha20*, *Diffie-Hellman*, Efisiensi Enkripsi, *Hybrid* Kriptografi, Keamanan Data.**

I. PENDAHULUAN

Keamanan data menjadi isu krusial di era digital, terutama bagi dokumen elektronik yang sering menjadi target serangan siber. Meskipun algoritma seperti *AES* telah banyak digunakan, tantangan dalam pengelolaan kunci dan kebutuhan akan efisiensi tinggi di perangkat tanpa akselerasi *hardware* masih menjadi hambatan. *Hybrid* kriptografi—penggabungan kriptografi simetris dan asimetris—hadir sebagai solusi untuk mengatasi kelemahan masing-masing metode [1]. Namun, sebagian besar implementasi *hybrid* masih menghadapi tantangan dalam efisiensi dan manajemen kunci.

Diffie-Hellman menjadi solusi untuk pembentukan kunci rahasia secara aman, meski tidak digunakan untuk enkripsi langsung [2]. Sementara itu, *ChaCha20* dikenal sebagai algoritma enkripsi modifikasi dari *stream cipher Salsa20* yang lebih cepat daripada *AES* dan algoritma lainnya, dengan waktu enkripsi hanya 39,8 milidetik [3] [4]. Penelitian ini mengusulkan kombinasi *Diffie-Hellman* dan *ChaCha20* untuk membentuk sistem *hybrid* kriptografi yang efisien dan aman dalam mengamankan *file* dokumen. Tujuan utama penelitian ini adalah mengimplementasikan metode *hybrid* tersebut dan mengevaluasi performanya dalam hal kecepatan enkripsi, dekripsi, dan *throughput*.

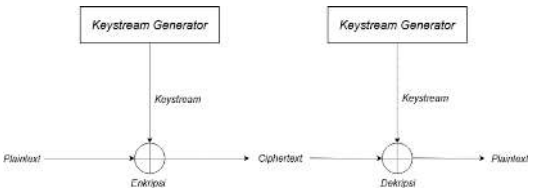
II. KAJIAN TEORI

A. Kriptografi

Kriptografi merupakan ilmu dan seni menjaga keamanan pesan saat pesan dikirim dari satu pihak ke pihak lain [1]. Kriptografi merupakan metode untuk mengacaukan dan menyandikan pesan menjadi kode rahasia atau *ciphertext* guna melindungi pesan rahasia [5]. Putera dan Siahaan (2022) menambahkan bahwa kriptografi bertujuan agar informasi hanya dapat diakses oleh pengirim dan penerima yang sah [6]. Proses kriptografi melibatkan tahapan penentuan kunci, enkripsi (*plaintext* menjadi *ciphertext*), dan dekripsi (*ciphertext* kembali menjadi *plaintext*) [2] [7].

B. *Stream cipher*

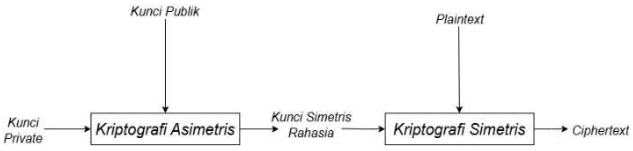
Stream cipher adalah metode enkripsi yang mengubah *plaintext* menjadi *ciphertext* secara bit per bit atau per byte [8]. Proses ini menggunakan *keystream* yang di-XOR dengan *plaintext* untuk menghasilkan *ciphertext*, dan sebaliknya. Nilai *keystream* sangat menentukan hasil enkripsi. *Stream cipher* pertama kali diperkenalkan melalui algoritma *Vernam cipher* [9].



GAMBAR 1
(KONSEP *STREAM CIPHER)*

C. *Hybrid* Kriptografi

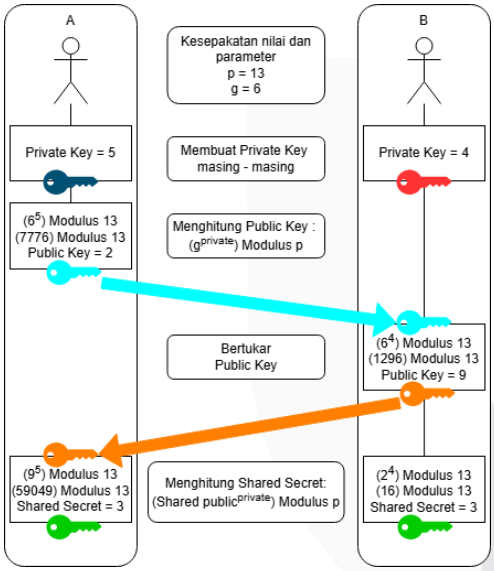
Hybrid kriptografi merupakan teknik yang menggabungkan dua atau lebih algoritma kriptografi yang berbeda untuk memanfaatkan keunggulan masing-masing metode [5]. Biasanya menggabungkan algoritma simetris dan asimetris, di mana algoritma simetris digunakan untuk mengenkripsi data dan algoritma asimetris digunakan untuk mengenkripsi kunci [1] [10].



GAMBAR 2
(KONSEP *HYBRID* KRIPTOGRAFI)

D. Diffie Hellman

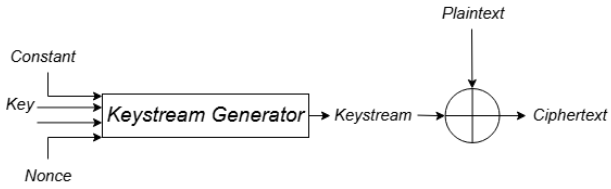
Diffie-Hellman adalah algoritma kunci publik pertama yang diperkenalkan oleh Diffie dan Hellman dan digunakan untuk menghasilkan serta bertukar kunci secara aman melalui jaringan publik [11] [6]. Metode ini tidak melakukan enkripsi/dekripsi langsung terhadap data, tetapi membentuk *public* dan *private key* yang menghasilkan *shared secret key* melalui perhitungan matematika [2].



GAMBAR 3
(SIMULASI *DIFFIE-HELLMAN)*

E. *ChaCha20*

ChaCha20 merupakan versi modifikasi dari algoritma Salsa20 yang dirancang oleh Daniel J. Bernstein, dan dikenal sebagai algoritma *stream cipher* yang cepat dan efisien [4]. *ChaCha20* memiliki waktu enkripsi rata-rata 39,8 milisekon (ms), lebih cepat dari *AES* (51 ms), *Twofish, Blowfish*, dan *Salsa20* [3]. Efisiensinya disebabkan oleh transformasi kolom ganda dalam setiap putaran [12], serta diimplementasikan secara luas, termasuk oleh Google [13].



GAMBAR 4
(KONSEP *CHACHA)*

F. *Throughput*

Throughput merupakan ukuran kecepatan proses enkripsi, dihitung sebagai perbandingan antara ukuran data (Tp) dengan waktu yang diperlukan untuk enkripsi (Et) [11]. *Throughput* dapat dirumuskan sebagai berikut:

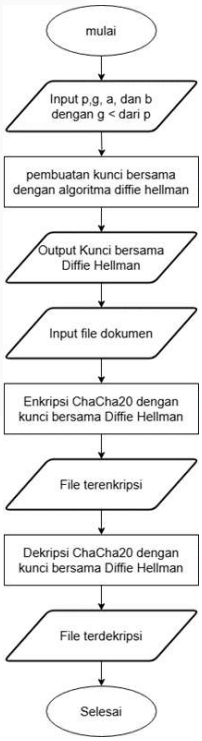
$$Throughput = \frac{Tp}{E_t} \quad (1)$$

Keterangan:
Tp = Total ukuran data (dalam kilobyte)
Et = Waktu enkripsi (dalam detik)

III. METODE

A. Alur Penelitian

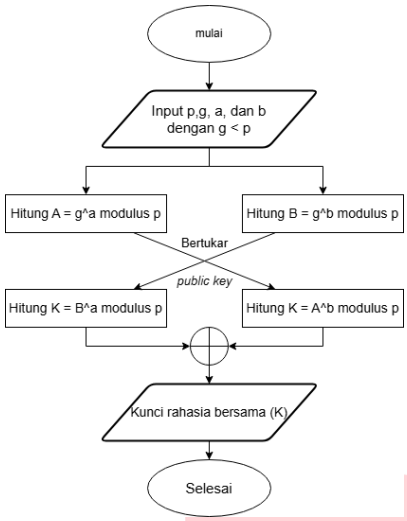
Keamanan *file* dokumen dilakukan menggunakan metode *hybrid* kriptografi dengan mengombinasikan algoritma *Diffie-Hellman* untuk membuat kunci simetris dan algoritma *ChaCha20* untuk enkripsi serta dekripsi data. Alur penelitian dapat dilihat pada gambar berikut.



GAMBAR 5
(ALUR PENELITIAN)

B. Proses Pembuatan Kunci

Pembuatan *shared secret* dilakukan dengan menggunakan algoritma *Diffie-Hellman*.

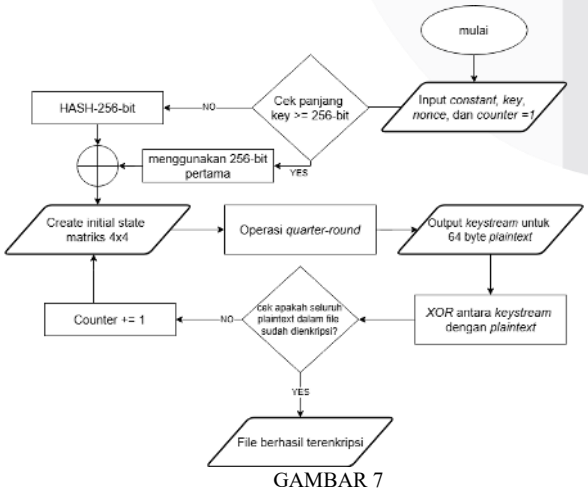


GAMBAR 6
(ALUR *DIFFIE-HELLMAN)*

Sistem akan dimulai dengan menentukan nilai dari parameter p, g, a, dan b dengan ketentuan g lebih kecil dari p. Parameter a dan b masing masing menggambarkan *private key* dari setiap individu, sedangkan parameter p dan g digunakan sebagai nilai yang disepakati oleh kedua individu a dan b untuk melakukan operasi perhitungan dari algoritma *Diffie-Hellman*. Nilai parameter p akan memiliki panjang 256-bit dan parameter a dan b akan memiliki Panjang 128-bit. Tahap berikutnya adalah pembuatan *public key* oleh masing masing individu. Hasil dari perhitungan tersebut adalah nilai *public key*. Kemudian, individu a dan b bertukar *public key*. Setelah pertukaran dilanjutkan dengan perhitungan *shared secret*.

C. Enkripsi dan Dekripsi *ChaCha20*

Proses enkripsi *ChaCha20* dilakukan menggunakan kunci *shared secret* yang dihasilkan dari *Diffie-Hellman*. *Shared secret* dipastikan panjangnya 256-bit. Jika kurang dari 256-bit akan dilakukan hash menggunakan *SHA-2*. *Shared secret* akan dimasukkan ke dalam blok *initial state* untuk melakukan operasi *quarter round* dan menghasilkan *keystream*.



GAMBAR 7
(ALUR *CHACHA20)*

Proses dekripsi dilakukan dengan cara yang sama. Namun sebelum memulai pembuatan *keystream*, data asli *file* yang terenkripsi akan dipisahkan terlebih dahulu untuk mengambil *nonce* yang diperlukan untuk proses pembuatan *keystream*.

D. Skenario Pengujian

Pengujian dilakukan untuk mengevaluasi implementasi *hybrid* kriptografi *Diffie-Hellman* dan *ChaCha20* dalam pengamanan *file* dokumen. Jenis *file* yang diuji meliputi .doc, .pdf (berisi teks/gambar/tabel), .pdf hasil *scan*, .xlsx, dan .pptx. Detail *file* pengujian disajikan pada Tabel berikut.

TABEL 1
(DETAIL *FILE)*

FILE TYPE	*FILE SIZE (KB)*
.doc, .pdf, .xlsx, .pptx	101
	501
	1024
	5128
	10258
.pdf (*scan*)	106
	507
	1062
	5142
	10452

File terenkripsi disimpan sebagai output.txt, kemudian didekripsi kembali untuk menguji integritas. Simulasi komunikasi dilakukan pada satu perangkat fisik menggunakan dua *virtual machine* (VM). VM pertama bertindak sebagai pengirim. *File* terenkripsi dikirim ke VM kedua yang berperan sebagai penerima, dan didekripsi menggunakan kunci *shared secret* yang sama. Hasil dekripsi dibandingkan dengan *file* asli untuk mengukur integritas data. Pemeriksaan dilakukan dengan mencocokkan hash value dari *file* asli dan *file* hasil dekripsi, dengan tiga algoritma hash: *CRC32* (untuk mendeteksi perubahan bit), *MD5* (untuk mendeteksi kerusakan acak), dan *SHA-1* (untuk identifikasi unik terhadap konten *file*).

E. Evaluasi Performa

Evaluasi performa dilakukan untuk membandingkan efisiensi algoritma *hybrid* kriptografi *Diffie-Hellman–ChaCha20* dengan algoritma lain, yaitu *AES*-256, *Twofish*, *ChaCha20*, dan *Diffie-Hellman–AES*-256 (DH-*AES*). Setiap algoritma dikonfigurasi dengan panjang kunci 256-bit sesuai standar yang digunakan. Detail karakteristik masing-masing algoritma ditampilkan pada Tabel berikut.

TABEL 2
(DETAIL ALGORITMA)

Algoritma	Ukuran Blok (bits)	Standard Ukuran Kunci (bits)	Ukuran Kunci yang digunakan (bits)	Tipe Cipher
AES-256	128	128, 192, 256	256	*Block*
Twofish	128	128, 192, 256	256	*Block*
ChaCha20	64	128, 256	256	*Stream*
DH-AES	128	128, 192, 256	256	*Block*
DH-ChaCha20	64	128, 256	256	*Stream*

Pengukuran performa dilakukan menggunakan tiga metrik utama:

1. Waktu Enkripsi

   Mengukur durasi konversi *plaintext* menjadi *ciphertext*. Semakin kecil waktu yang dibutuhkan, semakin efisien algoritma tersebut.

$$E_t = EN_t - ST_t \quad (2)$$

2. Waktu Dekripsi

   Mengukur waktu yang dibutuhkan untuk mengubah *ciphertext* kembali ke *plaintext*.

$$D_t = EN_t - ST_t \quad (3)$$

3. *Throughput*

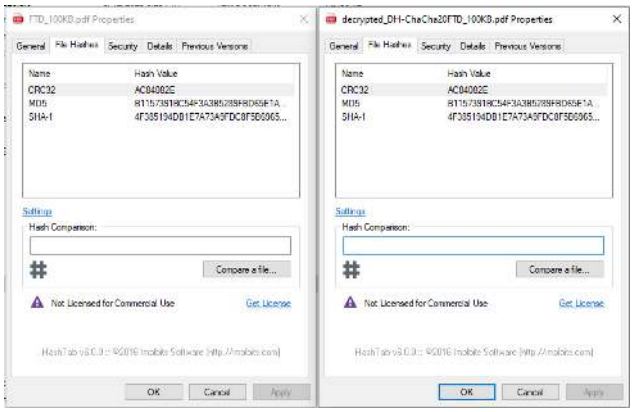
   Mengukur kecepatan enkripsi dan dekripsi berdasarkan jumlah data yang diproses per detik, dihitung dalam kilobyte per detik.

$$Throughput = \frac{Tp}{E_t} \quad (4)$$

IV. HASIL DAN PEMBAHASAN

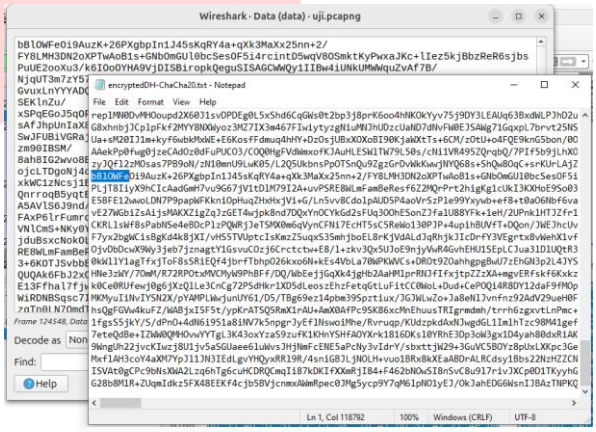
A. Hasil Integritas *File*

Setelah mengimplementasikan setiap algoritma yang telah dibahas sebelumnya. Fokus utama dari penelitian ini adalah integritas *file* asli dengan *file* terdekripsi pada kombinasi *Diffie-Hellman* dan *ChaCha20* serta performa setiap algoritma meliputi waktu enkripsi, waktu dekripsi, *throughput* enkripsi dan *throughput* dekripsi. Hasil implementasi menunjukkan bahwa integritas *file* asli dengan *file* terdekripsi *CRC32*, *MD5*, dan *SHA-1* adalah sama. Detail dapat dilihat pada gambar berikut.



GAMBAR 8
(PERBANDINGAN *HASH VALUE FILE)*

File juga telah terenkripsi dengan baik berupa format .txt seperti gambar berikut.



GAMBAR 9
(POTONGAN DATA TERENKRIPSI)

B. Hasil Performa Algoritma

Hasil menunjukkan bahwa algoritma kombinasi *Diffie-Hellman* dan *ChaCha20* menjadi algoritma yang tercepat dibandingkan dengan algoritma lainnya. Terutama *Twofish* sebagai algoritma terlama dari semua algoritma dan *file* yang diuji. Hasil disajikan dalam gambar. Gambar 10 menunjukkan waktu yang dibutuhkan pada setiap algoritma untuk mengenkripsi *file* tertentu. Gambar 11 menunjukkan *throughput* enkripsi yang didapatkan pada setiap algoritma untuk mengenkripsi *file* setiap detiknya. Sedangkan Gambar 12 menunjukkan waktu yang dibutuhkan pada setiap algoritma untuk mendekripsi *file* tertentu. Gambar 13 menunjukkan *throughput* dekripsi yang didapatkan pada setiap algoritma untuk mendekripsi *file* setiap detiknya.

FILE TYPE	*FILE SIZE* (KB)	Waktu Enkripsi (detik)				
		DH-ChaCha20	DH-AES	ChaCha20	AES	Twofish
.pdf	101	0.018	0.0208	0.0176	0.0176	0.1733
	501	0.0521	0.0536	0.0483	0.0576	0.3728
	1028	0.0718	0.0805	0.0735	0.0759	0.6325
	5178	0.2592	0.2805	0.265	0.2701	3.0433
	10822	0.5425	0.5431	0.5348	0.5803	6.3005
.doc	101	0.0175	0.0181	0.0164	0.0265	0.0966
	501	0.0447	0.0465	0.057	0.0589	0.3201
	1024	0.0708	0.0777	0.0786	0.0797	0.644
	5128	0.2741	0.2766	0.2645	0.264	3.2925
	10258	0.5013	0.5444	0.5014	0.5051	6.2906
.xlsx	101	0.0141	0.0193	0.0189	0.0227	0.0964
	501	0.0499	0.0659	0.0596	0.0607	0.3526
	1024	0.07	0.1074	0.0903	0.1001	0.6654
	5128	0.2677	0.3317	0.3017	0.2935	3.0788
	10258	0.4559	0.6258	0.5764	0.61	6.1699
.pptx	101	0.0184	0.0199	0.0196	0.0294	0.1002
	501	0.05	0.0511	0.0587	0.0511	0.3479
	1024	0.0723	0.0861	0.084	0.0773	0.6687
	5128	0.2962	0.3058	0.2793	0.3304	3.0867
	10258	0.5691	0.5835	0.5834	0.522	6.1455
.pdf (*scan*)	106	0.0188	0.0222	0.0196	0.0202	0.1002
	507	0.0471	0.0541	0.0576	0.0536	0.3504
	1062	0.0977	0.1247	0.1082	0.0784	0.6819
	5142	0.2896	0.2945	0.2901	0.2953	3.085
	10452	0.5983	0.7056	0.6641	0.6482	6.2418
AVERAGE		**0.1907**	**0.2136**	**0.2027**	**0.2051**	**2.0935**

GAMBAR 10
(HASIL WAKTU ENKRIPSI *FILE)*

FILE TYPE	*FILE SIZE* (KB)	Waktu Dekripsi (detik)				
		DH-ChaCha20	*DH-AES*	*ChaCha20*	*AES*	*Twofish*
.pdf	134	0.0976	0.1088	0.0438	0.0318	0.188
	667	0.0608	0.1145	0.1167	0.1196	0.3447
	1371	0.1054	0.2016	0.1859	0.1533	0.754
	6903	0.3985	0.9384	0.5602	0.5095	4.4119
	14430	0.8227	1.968	1.2396	1.2019	8.9419
.doc	135	0.0531	0.0747	0.0238	0.063	0.1125
	668	0.0656	0.1497	0.0694	0.1126	0.3615
	1365	0.0939	0.2434	0.186	0.1724	0.6761
	6837	0.545	0.6675	0.5955	0.5878	3.9385
	13677	0.7156	1.352	1.1638	1.1632	8.8155
.xlsx	135	0.025	0.0376	0.0246	0.0353	0.1292
	668	0.0635	0.1063	0.0607	0.0947	0.7416
	1365	0.0914	0.1541	0.128	0.1479	0.9598
	6837	0.3877	0.5938	0.4528	0.629	4.4808
	13677	0.7232	1.122	0.8662	0.9279	8.3114
.pptx	135	0.0256	0.0384	0.0233	0.0565	0.138
	667	0.0688	0.0817	0.0868	0.0754	0.4756
	1365	0.1063	0.1288	0.1223	0.1356	0.9183
	6838	0.4396	0.6596	0.436	0.4841	4.6339
	13677	0.9557	1.532	0.857	1.2473	8.2569
.pdf (scan)	141	0.0252	0.0413	0.0271	0.03	0.1515
	676	0.0832	0.1103	0.0631	0.0783	0.5233
	1416	0.1709	0.2429	0.1414	0.1408	1.0069
	5855	0.5957	0.4715	0.505	0.4234	3.7449
	13936	0.9648	1.425	0.9781	1.0552	8.7648
AVERAGE		**0.307392**	**0.502556**	**0.317344**	**0.38706**	**2.87126**

GAMBAR 12
(HASIL WAKTU DEKRIPSI *FILE)*

FILE TYPE	*FILE SIZE (KB)*	*Throughput* Enkripsi (KB/s)				
		DH-ChaCha20	*DH-AES*	*ChaCha20*	*AES*	*Twofish*
.pdf	101	5579.41	4827.55	5686.38	5678.62	578.11
	501	9596.44	9321.32	10346.79	8675.72	1341.31
	1028	14303.89	12771.5	13975	13542.3	1624.89
	5178	19970.75	18455.26	19537.84	19164.06	1701.1
	10822	19949.73	19927.04	20236.95	18650.12	1717.6
.doc	101	5758.81	5562.59	6143.64	3792.86	1040.75
	501	11189.85	10761.9	8782.66	8493.65	1563.37
	1024	14453.38	13175.15	13021.6	12842	1590.08
	5128	18709.15	18540.36	19386.95	19425.35	1557.47
	10258	20462.67	18843.33	20456.77	20306.32	1630.6
.xlsx	101	7135.5	5226.01	5338.75	4445.82	1044.63
	501	10028.39	7589.84	8391.98	8242.35	1392.19
	1024	14624.23	9525.95	11327.05	10220.02	1537.48
	5128	19153.77	15455.93	16997.57	17470.16	1665.36
	10258	18489.31	16391.93	17794.89	16816.61	1662.51
.pptx	101	5490.56	5061.69	5153.59	3433.97	1006.75
	501	10000.71	9792.19	8522.01	9789.13	1437.77
	1024	14151.31	11878	12180.58	13235.98	1530.02
	5128	17311.48	16766.97	18359.01	15518.97	1661.28
	10258	22497.23	17580.29	17582.6	19649.36	1669.12
.pdf (*scan*)	106	5595.2	4747.16	5380.1	5204.06	1050.23
	507	10738.02	9359.2	8790.08	9451.86	1444.63
	1062	10871.58	8513.52	9814.93	13534.8	1557.12
	5142	17754.72	17457.05	17719.09	17410.76	1666.48
	10452	17470.67	14813.46	15783.63	16123.52	1674.52
AVERAGE		**13651.470**	**12093.808**	**12668.418**	**12444.734**	**1453.814**

GAMBAR 11
(HASIL *THROUGHPUT* ENKRIPSI *FILE)*

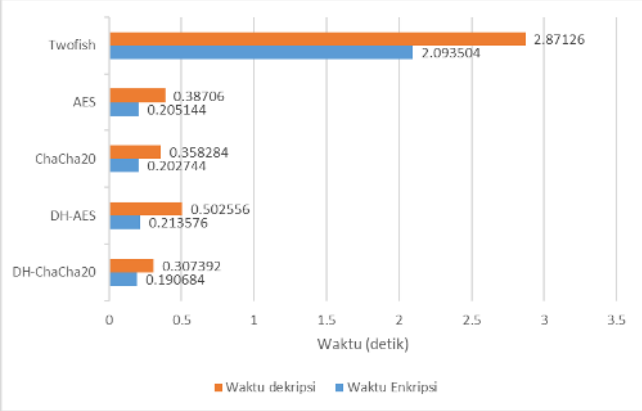
FILE TYPE	*FILE SIZE (KB)*	*Throughput* Dekripsi (KB/s)				
		DH-ChaCha20	*DH-AES*	*ChaCha20*	*AES*	*Twofish*
.pdf	141	1026.37	1221.92	3058.69	4210.66	712.91
	676	10974.92	5764.3	5713.88	5575.18	1934.89
	1416	13004.11	6822.03	7374.83	8943.24	1818.51
	6855	17322.91	7426.03	12323.07	13547.95	1564.75
	13936	17538.6	7786.73	11641.65	12006.43	1613.92
.doc	141	1892.63	1806.52	5668.75	2142.95	1199.77
	676	10179.01	4462.13	9627.42	5934.06	1847.94
	1416	14539.16	5608.22	7340.48	7919.23	2018.98
	6855	9409.51	10241.95	11480.88	11632.04	1736.18
	13936	19112.06	10115.96	11753.17	11758.94	1551.76
.xlsx	141	5368.69	3590.65	5470.03	3822.92	1044.63
	676	10500.44	6283.37	10999.08	7052.55	900.78
	1416	14924.73	8854.71	10658.86	9228.6	1422.22
	6855	17634.63	11513.93	15098.99	10869.92	1525.96
	13936	18912.57	12191.12	15788.8	14739.93	1645.57
.pptx	141	5261.83	3518.56	5767.46	2388.61	978.08
	676	9695.43	8165.92	7687.64	8842.04	1402.45
	1416	12826.7	10594.79	11152.67	10064.42	1486.47
	6855	15551.7	10366.45	15682.3	14126.91	1475.48
	13936	14310.85	8927.6	15958.99	10965.47	1656.38
.pdf (*scan*)	141	5569.3	3418.04	5188.45	4697.36	930.72
	676	8108.69	6127.17	10694.51	8634.28	1291.91
	1416	8283.46	5830.78	10008.48	10059.5	1406.14
	6855	11506.83	12417.52	13573.25	13828.95	1563.6
	13936	14444.78	9782.18	14247.64	13209.75	1590
AVERAGE		**11515.997**	**7313.543**	**10158.399**	**9048.076**	**1452.8**

GAMBAR 13
(HASIL *THROUGHPUT* DEKRIPSI *FILE)*

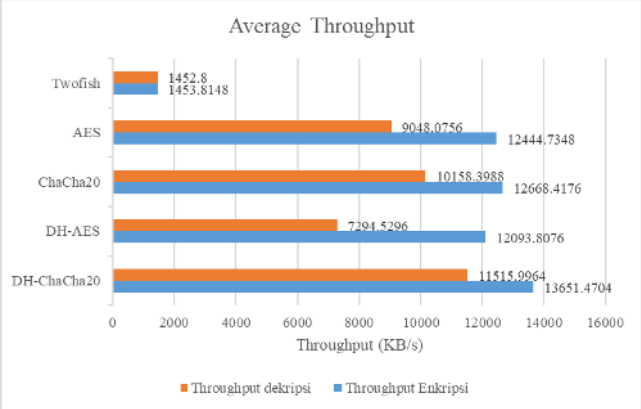
Gambar 14 perbandingan waktu enkripsi dan dekripsi dari lima algoritma kriptografi, yaitu *Twofish*, *AES*, *ChaCha20*, *Diffie-Hellman AES*, dan *Diffie-Hellman ChaCha20*. Berdasarkan hasil yang ditampilkan, algoritma *Twofish* memiliki performa paling lambat, dengan waktu enkripsi sebesar 2,09 detik dan dekripsi mencapai 2,87 detik. Sebaliknya, algoritma DH-*ChaCha20* menunjukkan kinerja paling cepat, dengan waktu enkripsi 0,19 detik dan dekripsi 0,30 detik. Algoritma *AES* dan *ChaCha20* tanpa kombinasi juga menunjukkan efisiensi tinggi, masing-masing berada di bawah 0,25 detik. Sementara DH-*AES* berada pada posisi

menengah, lebih cepat dari *Twofish* namun sedikit lebih lambat dari *ChaCha20* murni. Secara umum, hasil ini menunjukkan bahwa kombinasi *Diffie-Hellman* dengan *ChaCha20* mampu meningkatkan efisiensi proses kriptografi tanpa mengurangi keandalan, sedangkan algoritma *Twofish* memiliki waktu proses yang relatif tinggi dan kurang optimal dalam hal kecepatan.

Gambar 15 perbandingan nilai rata-rata *throughput* dari lima algoritma kriptografi, baik saat proses enkripsi maupun dekripsi. Hasil pengujian memperlihatkan bahwa algoritma DH-*ChaCha20* menghasilkan nilai *throughput* tertinggi, yaitu 13.651 KB/s saat enkripsi dan 11.515 KB/s saat dekripsi, mengungguli algoritma lainnya dalam hal kecepatan pemrosesan data. Algoritma *ChaCha20* murni menempati posisi kedua dengan *throughput* enkripsi sebesar 12.668 KB/s dan dekripsi 10.158 KB/s. Disusul oleh *AES* dengan *throughput* enkripsi 12.444 KB/s dan dekripsi 9.048 KB/s. Sementara itu, DH-*AES* berada di bawahnya dengan nilai 12.093 KB/s (enkripsi) dan 7.294 KB/s (dekripsi).



GAMBAR 14
(*AVERAGE* WAKTU)



GAMBAR 15

(*AVERAGE THROUGHPUT)*

V. KESIMPULAN

Implementasi *hybrid* kriptografi menggunakan algoritma *Diffie-Hellman* dan *ChaCha20* terbukti efektif dalam mengamankan *file* dokumen. Keamanan sistem telah terverifikasi melalui pengujian menggunakan Wireshark, di mana *file* telah terenkripsi dengan baik saat proses transmisi. Validitas integritas data juga telah dipastikan melalui pengecekan nilai hash antara *file* asli dan *file* hasil dekripsi yang menunjukkan hasil identik. Dari segi performa, algoritma ini menunjukkan efisiensi tinggi, dengan waktu enkripsi sebesar 0.1852 detik dan dekripsi 0.29582 detik, menjadikannya yang tercepat dibandingkan algoritma lain yang diuji. Selain itu, rata-rata *throughput* enkripsi mencapai 13.997,408 KB/s dan *throughput* dekripsi sebesar 11.499,928 KB/s, menegaskan keunggulan algoritma ini dalam hal kecepatan dan konsistensi performa. Dengan demikian, kombinasi algoritma *Diffie-Hellman* dan *ChaCha20* tidak hanya menawarkan keamanan yang kuat, tetapi juga efisiensi dan kecepatan tinggi dalam proses kriptografi dokumen digital.

REFERENSI

[1] J. A. Hutabarat, “Implementasi Kriptografi Hibrida Dan Steganografi Ihwt Dalam Pengamanan Data Teks,” *Jurnal Pelita Informatika*, vol. 8, no. 3, 2020.

[2] N. S. Atmaja, “Symmetric Cryptography Modification Using Diffie Hellman On RDBMS,” *Journal of Information Technology, computer science and Electrical Engineering (JITCSE)*, vol. 1, no. 1, pp. 40–48, 2024, doi: 10.30596/jitcse.v1i1.xxxx.

[3] R. K. Muhammed *et al.*, “Comparative Analysis of *AES*, Blowfish, *Twofish*, Salsa20, and *ChaCha20* for Image Encryption,” *Kurdistan Journal of Applied Research*, vol. 9, no. 1, pp. 52–65, May 2024, doi: 10.24017/science.2024.1.5.

[4] S. Barbero, D. Bazzanella, and E. Bellini, “Rotational Cryptanalysis on ChaCha *Stream cipher*,” *Symmetry (Basel)*, vol. 14, no. 6, Jun. 2022, doi: 10.3390/sym14061087.

[5] M. W. Saputra, A. Sapitri, and M. A. Putri, “PENERAPAN KRIPTOSISTEM *HYBRID* UNTUK MENGENKRIPSI PESAN MENGGUNAKAN ALGORITMA RSA CIPHER,” 2023. [Online]. Available: https://jurnal.ittc.web.id/index.php/jct/

[6] A. Putera and U. Siahaan, “Pembangkitan Kunci pada Algoritma Hill Cipher menggunakan Teknik Distribusi Angka *Diffie-Hellman*,” *Nasional Teknologi Informasi dan Komputer)*, vol. 6, no. 1, 2022, doi: 10.30865/komik.v6i1.5775.

[7] Dhea Agustina Akmal, Dicha Mutia Dhani, and Febby Syahila, “Kombinasi Kriptografi Modern Dalam Keamanan Pesan Teks,” *Saturnus : Jurnal Teknologi dan Sistem Informasi*, vol. 2, no. 3, pp. 119–128, Jul. 2024, doi: 10.61132/saturnus.v2i3.204.

[8] E. Ariyanto and T. Indah Pravitasari, “ANALISA IMPLEMENTASI ALGORITMA *STREAM CIPHER* SOSEMANUK DAN DICING DALAM PROSES ENKRIPSI DATA,” *Seminar Nasional Informatika*, 2008.

[9] K. Hidayatuloh, Y. Yustantina, and K. Kusmadi, “PERBANDINGAN METODE *STREAM CIPHER* DAN HILL CIPHER DALAM KEAMANAN DATA,” *Infotronik : Jurnal Teknologi Informasi dan Elektronika*, vol. 6, no. 1, p. 27, Jun. 2021, doi: 10.32897/infotronik.2021.6.1.647.

[10] M. G. Ar Romandhon, A. Junaidi, and A. N. Sihananto, “PENERAPAN *HYBRID* CRYPTOGRAPHY MENGGUNAKAN CAMELLIA DAN DUAL MODULUS RSA PADA PERTUKARAN *FILE*,” *Jurnal Informatika dan*

Teknik Elektro Terapan, vol. 12, no. 3S1, Oct. 2024, doi: 10.23960/jitet.v12i3S1.5218.

[11] L. Nisa, T. Indriyani, and M. Ruswiansari, "Aplikasi Enkripsi Citra dan Teks Menggunakan Algoritma *Diffie-Hellman* dan ElGamal," 2020.

[12] V. R. Kebande, "Extended-*ChaCha20 Stream cipher* With Enhanced Quarter Round Function," *IEEE Access*, vol. 11, pp. 114220–114237, 2023, doi: 10.1109/ACCESS.2023.3324612.

[13] M. S. Mahdi, N. F. Hassan, and G. H. Abdul-Majeed, "An improved chacha algorithm for securing data on IoT devices," *SN Appl Sci*, vol. 3, no. 4, Apr. 2021, doi: 10.1007/s42452-021-04425-7.