Prosiding Seminar Nasional Teknologi Informatika
Volume 2 Nomor 1
November 2019
ISBN : 978-602-50006-1-4

Perbandingan Algoritma Kriptografi *Hash MD5* dan *SHA-1*

M. Hamdani Santoso
Universitas Medan Area
hamdanisantos123@gmail.com

Nardianti Dewi Girsang
Universitas Medan Area
nardiantidewigirsang@gmail.com

Heldawaty Siagian
Universitas Medan Area
siagianheldawaty@gmail.com

Agung Wahyudi
Universitas Medan Area
agungwah006@gmail.com

Bunaya Arthavia Sitorus
Universitas Medan Area
arthaviasitorus@gmail.com

Abstrak

Keamanan selalu menjadi perhatian di bidang Teknologi Informasi, terutama di lingkungan *cloud*. Salah satu peran utama dalam keamanan informasi adalah *hash* Kriptografi. Algoritma *hash*, yang juga disebut algoritma intisari pesan digunakan untuk menghasilkan intisari pesan khusus untuk pesan acak. Algoritma *Hashing* diklaim sebagai elemen penting dalam bidang kriptografi dan praktik keamanan. *Hashing* memiliki properti satu arah, dan karena properti inilah mereka diangap memiliki peran yang besar dalam memberikan integritas pesan dan penyimpanan kata sandi. Algoritma *hash* digunakan secara luas terutama dalam otentikasi *login* dan memverifikasi integritas pesan. Dalam hal ini, algoritma *hash* dapat membantu menjaga integritas pesan. Pada penelitian ini digunakan algoritma *hash* MD5 dan SHA-1. Algoritma *hash* MD5 memiliki 32 bit karakter,

Prosiding Seminar Nasional Teknologi Informatika
Volume 2 Nomor 1
November 2019
ISBN : 978-602-50006-1-4

sedangkan SHA-1 memiliki 40 bit karakter. Dalam ilmu kriptografi, dua algoritma *hash* tersebut digunakan untuk melindungi informasi yang rawan disalahgunakan terutama *username* dan *password*. Algoritma *hash MD5* memiliki waktu proses lebih cepat dari pada algoritma hash *SHA-1* dengan perbandingan waktu kecepatan rata-rata untuk *MD5* adalah 0.1635s dan *SHA-1* adalah 0.164s.

Kata Kunci : Kriptografi, *Hash, MD5, SHA-1, Message Digest*

I. PENDAHULUAN

Kriptografi seharusnya sudah banyak diimplementasikan terutama di bidang Teknologi untuk keamanan dan proteksi. Hampir setiap kali mengakses internet, kriptografi digunakan tanpa sepengetahuan penggunanya. Misalnya untuk berkomunikasi atau bertukar informasi, informasi tersebut akan di enkripsi sebelum dikirimkan dan dideskripsi kembali sebelum di terima oleh penerima. Contoh sederhananya, untuk *login* ke *email* saja membutuhkan fungsi *hash* untuk memeriksa *password* nya benar atau tidak.

Mekanisme keamanan dalam kriptografi adalah mekanisme *hashing* yang menggunakan fungsi hash dan menghasilkan nilai *hash*. Keamanan algoritma kriptografi tergantung pada jenis teks/kalimat seperti *plaintext*, teks sandi proses *encrypt* dan *decrypt*, panjang kunci. Jika panjang kunci lebih dari itu algoritma dianggap lebih aman. Ada dua cara menggunakan kunci dalam kriptografi. Salah satunya menggunakan kunci yang sama untuk proses enkripsi dan dekripsi yang disebut kriptografi kunci simetris. Lainnya menggunakan kunci yang berbeda satu untuk enkripsi dan lainnya untuk dekripsi yang disebut asimetrik kunci kriptografi.

Biasanya dalam kriptografi, ketika mengenkripsi suatu pesan harus dapat di dekripsi dengan sempurna. Sebaliknya, untuk fungsi hash justru tidak harus didekripsi[1]. Fungsi hash yang konseptual adalah fungsi yang hasil *hash value* tidak dapat didekripsi kembali. Hal ini dimanfaatkan untuk kerahasiaan. Ada berbagai macam algoritma untuk melakukan *hash.* Namun, karena semakin banyak ahli kriptanalisis, banyak algoritma terdahulu dianggap *strong* ternyata telah ditemukan celahnya.

Algoritma yang masih dianggap *strong* hingga kini adalah MD5 dan SHA-1. MD5 adalah *Message-Digest Algorithm.* MD5 sudah sangat umum digunakan sebagai fungsi *hash* kriptografi. MD5 sering digunakan untuk memastikan kepaduan data. Hasil yang didapat dari penggunaan MD5 adalah angka *hexadecimal* dengan panjang 32 karakter. Walaupun MD5 banyak dipakai secara umum untuk aplikasi keamanan, tetapi sudah ada yang menemukan celah dari MD5. Sehingga, para kriptografi sudah mulai menganjurkan penggunaan algoritma SHA-1 dibandingkan dengan MD5[2].

Dalam penelitian ini dilakukan perbandingan terhadap algoritma *hash* MD5 dan algoritma *hash* SHA-1. Yang bertujuan untuk mengetahui algoritma *hash* mana yang lebih unggul dalam sisi kecepatan maupun keamanannya.

Menurut penelitian Zhong[1] yang membahas tentang "Javaweb Login Authentication Based On Improved Md5 Algorithm". Hasil penelitiannya menyimpulkan bahwa berdasarkan algoritma enkripsi MD5 dapat ditingkatkan secara efektif dengan melindungi informasi kata sandi pengguna itu yang telah dibuktikan oleh beberapa percobaan, karena kata sandi disimpan dalam basis data bukan lagi kata sandi yang sebenarnya, tetapi kata sandi yang sudah ditambahkan huruf, angka, maupun simbol.

Menurut penelitian Bhandari[3] yang membahas tentang "Enhancement of MD5 Algorithm for Secured Web Development", hasil penelitiannya yaitu perlu penambahan fungsi *hash* baru karena algoritma MD5 menunjukkan banyak kelemahan dan kerentanan terhadap berbagai serangan seperti *birthday attack*, *brute force*, dan *rainbow table*, dapat dengan mudah dikurangi dengan menggunakan algoritma hibrida dengan panjang keluaran variabel dan teknik kunci.

Menurut penelitian Kurniawan[4] yang membahas tentang "Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost", hasil penelitiannya yaitu Kinerja dari algoritma hash SHA-3 dapat dikatakan lebih baik daripada algoritma SHA-1 berdasarkan pada hasil pengujian yang telah dilakukan. Pada saat pengujian *brute-force*, waktu yang ditempuh untuk mendapatkan *plaintext* 8, 9 dan 10 karakter dari hash SHA-3

ISBN : 978-602-50006-1-4

memakan waktu yang sangat berbanding jauh dengan algoritma SHA-1, membuat algoritma SHA-3 ini sangat tahan terhadap serangan dibandingkan dengan SHA-1.

II. TINJAUAN PUSTAKA

A. Algoritma Hash

Fungsi *hash* adalah sebuah fungsi yang menerima input berupa *string* lalu diproses sesuai dari standar fungsi *hash* tersebut yang kemudian akan mengeluarkan *output* berupa *string* dengan panjang yang tetap dan panjang tersebut tidak tergantung pada ukuran awal dari *string input* [5]. Fungsi kriptografi *hash* adalah sebuah fungsi matematika yang digunakan mencerna pesan, artinya dibutuhkan sebuah pesan sebagai input dan menghasilkan output sebagai nilai hash, ukuran dari nilai output hash tergantung dari algoritma yang digunakan seperti 64 bit, 128 bit dan 256 bit. *Hash* merupakan enkripsi satu arah, satu arah berarti tidak mempunyai fungsi untuk melakukan pengembalian nilai yang sudah di enkripsi [6].
Beberapa Algoritma *hashing* yang telah berkembang adalah sebagai berikut :
1. MD4(*Message-Digest algortihm 4*)
2. MD5 (*Message-Digest algortihm 5*)
3. MD5 (*$pass.$salt*)
4. MD5 (*$salt.$pass*)
5. SHA-1 (*Secure Hash Algorithm*)
6. SHA-256 (*Secure Hash Algorithm*)
7. SHA-512 (*Secure Hash Algorithm*)
8. Base64

B. Algoritma Hash MD5

MD5 (*Message Digest* 5) adalah Hash MD5 diperkenalkan oleh Profesor Ronald L. Rivest dan merupakan bagian dari kriptografi modern. Algoritma Hash MD5 adalah fungsi hash searah dengan nilai hash 128 bit. Hash dikatakan sebagai fungsi searah karena pesan yang dimasukkan akan diubah menjadi pesan pendek atau "intisari pesan" dan sulit untuk kembali ke pesan awal. Fungsi Hash dapat mengubah pesan input yang memiliki panjang berubah-ubah, menjadi pesan singkat yang panjangnya selalu diperbaiki.[7]

Langkah-langkah untuk menghasilkan *message digest* adalah sebagai berikut:

1. Penambahan bit-bit pengganjal

Pesan mula-mula ditambahkan dengan bit-bit pengganjal sehingga panjang pesan (dalam satuan bit) paralel dengan modulo 512. Setelah panjang data menjadi kelipatan 512, selanjutnya akan dikurangi 64 bit.

2. Penambahan nilai panjang pesan semula

Representasi sebesar 64 bit tersebut akan ditambahkan ke pesan.

3. Inisialisasi penyangga (*buffer*) MD5.

MD5 membutuhkan 4 buah register sebagai *buffer* dengan panjang masing-masing 32 bit, sehingga total panjang *buffer* sebesar 128 bit. Keempat register ini diberi nama A, B, C dan D yang diinisialisasi dalam nilai heksa sebagai berikut:
A = 01 23 45 67
B = 89 ab cd ef
C = fe dc ba 98
D = 76 54 32 10

4. Pengolahan pesan dalam blok bernilai 512 bit.

Dalam langkah ini, awalnya didefinisikan 4 buah fungsi dengan inputan 3 buah *word* 32 bit dan menghasilkan 1 buah *word* 32 bit. Inisialisasi fungsi dinyatahkan dalam nama F, G, H dan I adalah sebagai berikut:
F (X, Y, Z) = XY or (~X)Z
G (X,Y,Z) = XZ or Y(~Z)
H (X, Y, Z) = X xor Y xor Z
I (X, Y, Z) = Y xor (X or (~Z))

Fungsi-fungsi tersebut masing-masing berisikan 16 kali operasi dasar terhadap inputan yang memanfaatkan suatu nilai elemen T. nilai T adalah sebuah array berisi 64 elemen yang didapat dengan perhitungan tertentu menggunakan fungsi sinus. Pesan yang diinput dibagi menjadi n buah blok berukuran 512 bit. Setiap blok tersebut diproses bersama dengan buffer sehingga menghasilkan output sebesar 128 bit. Dalam proses ini terdiri dari 4 kali proses, masing-masing proses melakukan operasi tiap-tiap fungsi 16 kali banyaknya dan memakai elemen operasi dasar T. Hasil akhir dari algoritma MD5 ini akan menghasilkan output (*message digest*) dengan panjang yang tetap sebesar 32 karakter.

C. Algoritma Hash SHA-1

SHA dikembangkan oleh *National Institute of Standards and Technology (NIST)* dan dipublikasikan sebagai *Federal Information Processing Standards (FIPS 180)* pada tahun 1993. *Secure Hash Standard (SHS)* menspesifikasikan SHA-1 untuk menghitung nilai *hash* dari sebuah *message* atau *file*. SHA-1 memiliki panjang pesan maksimal 264 bits dan memiliki keluaran sebesar 160 bits yang dinamakan *message digest* atau *hash code. Message digest* tersebut dapat dimanfaatkan

ISBN : 978-602-50006-1-4

sebagai *input* untuk *Digital Signature Algorithm (DSA),* yang digunakan untuk menghasilkan *signature* untuk memvalidasi pesan tersebut[4].

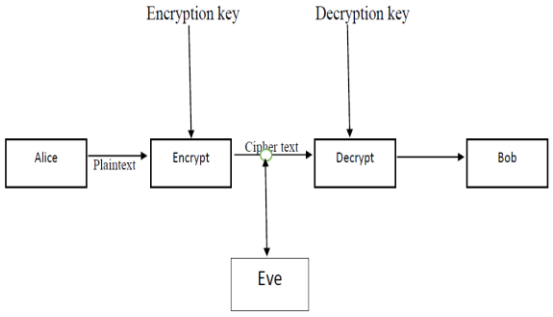
Pada saat ditemukan kelemahan pada algoritma SHA-0, berbagai revisi dan perbaikan dilakukan untuk membuat suatu algoritma yang lebih baik. Hasil dari perbaikan tersebut diterbitkan pada tahun 1995 dan dijadikan acuan untuk pembuatan algoritma SHA-1.

Algoritma SHA-1 merupakan revisi teknis dari algoritma SHA. Algoritma SHA-1 dapat dikatakan aman karena proses perhitungannya tidak memungkinkan untuk menemukan pesan yang sebenarnya dari *message digest* yang dihasilkan. Setiap perubahan yang terjadi pada pesan saat perjalanan akan menghasilkan *message digest* yang berbeda. Algoritma SHA-1 berbasis pada algoritma MD4 dan rancangannya sangatlah mirip terhadap algoritma tersebut.

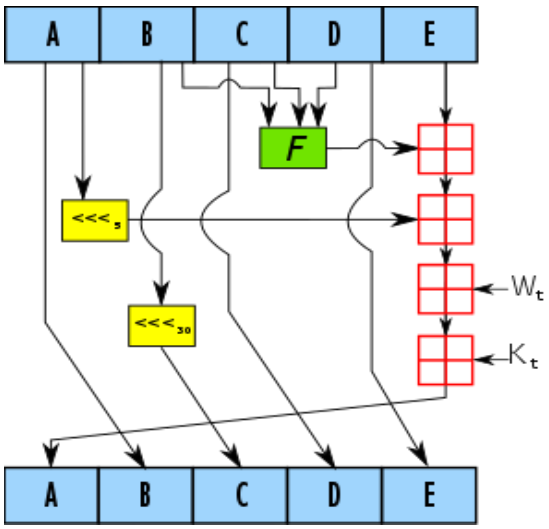
III. METODE PENELITIAN

Metode penelitian yang digunakan adalah *Library Research*. Metode ini dilakukan dengan cara mengumpulkan semua teori-teori yang berhubungan dengan *Enkripsi* dan *Hashing*, mengumpulkan penelitian-penelitian sebelumnya yang berhubungan dengan objek yang diteliti dan menganalisa serta membahas contoh implementasi *hash* MD5 dan SHA-1 pada sistem lalu membandingkannya.

Proses paling dasar dari penyampaian pesan terenkripsi ditunjukkan oleh diagram berikut :

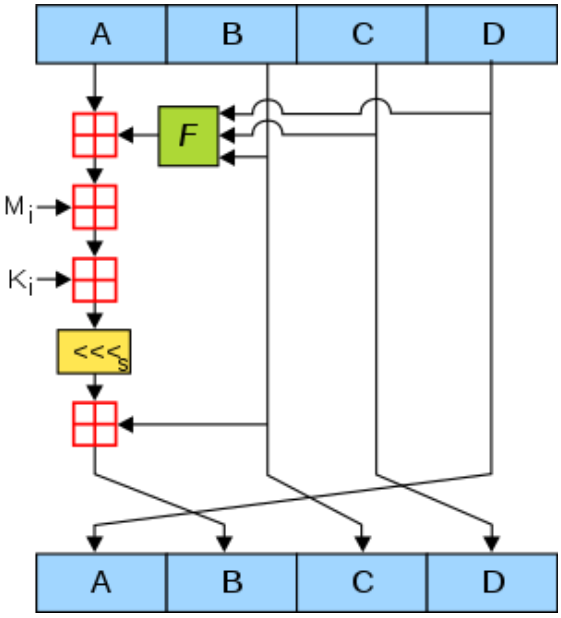


Gambar 1. Proses Enkripsi dan Deskripsi pada sebuah Pesan



Gambar 2. Fungsi Kompresi pada SHA-1 (sumber:http://crppit.epfl.ch/documentation/Hash_Function/Documentation/)

A, B, C, D, E adalah pesan 32-bit. *F* adalah fungsi nonlinear yang berbeda-beda. <<<*n* menandakan perubahan bit kiri sesuai *n*. Wt adalah kata pesan yang diperluas dari putaran *t*. Kt adalah konstanta putaran dari putaran *t*. ⊞ menunjukkan penambahan modulus 2^{32}.



Gambar 3. Fungsi Kompresi pada MD5 (sumber:http://crppit.epfl.ch/documentation/Hash_Function/Documentation/)

ISBN : 978-602-50006-1-4

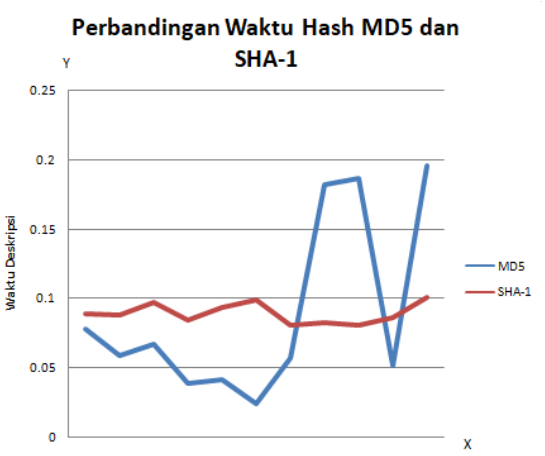
MD5 terdiri dari 64 operasi ini, dikelompokkan dalam empat putaran dari 16 operasi. *F* adalah fungsi nonlinear, satu fungsi digunakan di setiap putaran. M_i menunjukkan blok 32-bit dari input pesan, dan K_i menunjukkan konstanta 32-bit, berbeda untuk setiap operasi. $<<<s$ menandakan perubahan bit kiri sesuai tempat *s*. *s* bervariasi untuk setiap operasi. $\boxplus$ menunjukkan penambahan modulus 2^{32}.

IV. HASIL DAN PEMBAHASAN

Perbandingan dilakukan dengan mengeksekusi dan mengimplementasi kedua algoritma kriptografi hash MD5 dan SHA-1 kemudian menghitung waktu deskripsinya masing-masing. Untuk menghitung waktu eksekusi, dalam penelitian ini menggunakan web *decrypter* yaitu https://md5decrypt.net. Selain dapat mendekripsi *hash* MD5 dan SHA-1, web tersebut juga otomatis menghitung waktu yang dibutuhkan untuk mendeskripsikan kembali *hash* tersebut. Berikut table perbandingan eksekusi yang dapat diperoleh :

Tabel 1 Perbandingan Eksekusi MD5 dan SHA-1

No	Plain Text	Jumlah Karakter	Waktu MD5	Waktu SHA-1
1	z	1 Karakter	0.078s	0.089s
2	z0	2 Karakter	0.059s	0.088s
3	0p!	3 Karakter	0.067s	0.097s
4	dan1	4 Karakter	0.039s	0.085s
5	helda	5 Karakter	0.041s	0.094s
6	bunaya	6 Karakter	0.024s	0.099s
7	wahyudi	7 Karakter	0.057s	0.081s
8	pr0kl@m^s!	10 Karakter	0.182s	0.083s
9	qwerty123007	11 Karakter	0.187s	0.081s
10	kriptografik	12 Karakter	0.051s	0.086s
11	@dm!n!5tr6t0r!@#	16 Karakter	0.196s	0.101s



Gambar 3 Grafik Perbandingan Waktu Hash MD5 dan SHA-1

V. KESIMPULAN

Dari pembahasan diatas, maka dapat diambil beberapa kesimpulan, yaitu :

1. Algoritma kriptografi hash MD5 dan SHA-1 memiliki karakter yang berbeda dari sisi keamanan datanya karena jumlah *cipher* yang berbeda diantara keduanya.
2. Implementasi MD5 untuk kemanan data sebuah *username* atau *password* tidak dianjurkan saat ini, karena MD5 rentan terhadap *collision* maka perlu diperbaharui dengan algoritma lain sebagai pendukung.
3. Pengacakan karakter pada SHA-1 lebih banyak jumlahnya sehingga algoritmanya juga lebih rumit dari MD5, ini yang membuat SHA-1 menjadikan lebih unggul dari MD5
4. Selisih waktu deskripsi SHA-1 yang tidak banyak berbeda menunjukkan bahwa, SHA-1 konstan dipecahkan meskipun *plain text* memiliki string, numeric, maupun *symbol*.
5. Untuk beberapa kasus MD5 membutuhkan waktu lebih lama dideskripsi, namun dalam sisi keamanan MD5 masih jauh tertinggal dari SHA-1
6. Algoritma *hash MD5* memiliki waktu proses lebih cepat dari pada algoritma hash *SHA-1* dengan perbandingan waktu kecepatan rata-rata untuk *MD5* adalah 0.1635s dan *SHA-1* adalah 0.164s.

ISBN : 978-602-50006-1-4

VI. REFERENSI

[1] L. Zhong, W. Wan, and D. Kong, "Javaweb login authentication based on improved MD5 algorithm," *ICALIP 2016 - 2016 International Conference on Audio, Language and Image Processing - Proceedings*. pp. 131–135, 2017.

[2] H. Dobbertin, "Cryptanalysis of MD5 Compress," *Present. rump Sess. Eurocrypt'96*, pp. 5–6, 1996.

[3] A. Bhandari, M. Bhuiyan, and P. W. C. Prasad, "Enhancement of MD5 Algorithm for Secured Web Development," *J. Softw.*, vol. 12, no. 4, pp. 240–252, 2017.

[4] F. Kurniawan, A. Kusyanti, and H. Nurwarsito, "Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 9, pp. 803–812, 2017.

[5] N. Kishore and B. Kapoor, "Attacks on and advances in secure hash algorithms," *IAENG Int. J. Comput. Sci.*, vol. 43, no. 3, pp. 326–335, 2016.

[6] P. P. Pittalia, "A Comparative Study of Hash Algorithms in Cryptography," vol. 8, no. 6, pp. 147–152, 2019.

[7] N. Khairina, M. K. Harahap, and J. H. Lubis, "The Authenticity of Image using Hash MD5 and Steganography Least Significant Bit," *IJISTECH (International J. Inf. Syst. Technol.*, vol. 2, no. 1, p. 1, 2018.

[8] P. V. Rao, S. G. Rao, P. C. Reddy, G. R. Sakthidharan, and Y. M. Kumar, "Improve the integrity of data using hashing algorithms," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 2018–2023, 2019.

[9] N. F. Ginting and M. Ginting, "Perbandingan Kriptografi RSA dengan Base64," *J. Tek. Inform. Unika St. Thomas*, vol. 02, no. 02, pp. 47–52, 2017.

[10] P. Ora and P. R. Pal, "Data security and integrity in cloud computing based on RSA partial homomorphic and MD5 cryptography," *IEEE Int. Conf. Comput. Commun. Control. IC4 2015*, 2016.

[11] S. Gupta, N. Goyal, and K. Aggarwal, "A Review of Comparative Study of MD5 and SSH Security Algorithm," *Int. J. Comput. Appl.*, vol. 104, no. 14, pp. 1–4, 2014.

[12] J. B. Sanger, "DESAIN DAN IMPLEMENTASI MEKANISME TANDA TANGAN DIJITAL DALAM PERTUKARAN DATA DENGAN HASH MD5 DAN ENKRIPSI / DEKRIPSI MENGGUNAKAN ALGORITMA RSA," vol. 12, no. 2, 2015.

[13] E. V. Maliberan, A. M. Sison, and R. P. Medina, "A new approach in expanding the hash size of MD5," *Int. J. Commun. Networks Inf. Secur.*, vol. 10, no. 2, pp. 374–379, 2018.

[14] G. Gupta, "What is Birthday attack??"

[15] K. D. Wandani and S. Sinurat, "Implementasi Secure Hash Algoritma Untuk Pengamanan Pada File Video," *Maj. Ilm. INTI*, vol. 13, pp. 165–168, 2018.

[16] Saputra Sandi & Susandri, "Pada Sistem Keamanan Dokumen Dalam Folder," *J. Ilm. Media Process.*, vol. 11, no. 1, pp. 689–696, 2016.

[17] H. Agung, "Kriptografi Menggunakan Hybrid Cryptosystem dan Digital Signature," *J. Tek. Inform. dan Sist. Inf.*, vol. 3, no. 1, pp. 34–45, 2016.

[18] M. Arief and N. Ikhsan, "Kriptografi Rsa Pada Aplikasi File Transfer Client- Server Based," *J. Ilm. Teknol. Inf. Terap.*, vol. I, no. 3, pp. 3–7, 2015.