Research *: Journal of Computer, Information System, & Technology Management* **Online** ISSN: 2615-7357
Vol. 4, **No. 1**. **April 2021, Pages 84-89** **Print** ISSN: 2615-7233

Penerapan Hashing SHA1 dan Algoritma Asimetris RSA untuk Keamanan Data pada Sistem Informasi berbasis Web

Implementation Of Hashing SHA1 And RSA Asymetric Algorithm For Data Security In Web-Based Information Systems

Ridho Pamungkas*[1], Ferdinand Wahyu Zulaiman Zaney [2]
[1,2,3] *Sistem Informasi; Universitas PGRI Madiun*
[1, 2, 3] *Madiun, Indonesia*
e-mail: ***[1]ridho.pamungkas@**unipma**.ac.id**, [2]ferdinandferlanda@gmail.com

Abstrak *- Penggunaan teknologi informasi dalam memperoleh data di era society 5.0 merupakan hal yang tidak dapat dihindari. Hampir semua data yang kita gunakan tersimpan dalam database pada sebuah sistem informasi yang dapat di akses secara online dan dapat dipergunakan oleh sembarang pengguna. Sistem Informasi menjadi sangat rentan apabila data yang kita miliki dipergunakan atau diakses oleh orang lain dan mempergunakan data tersebut untuk hal yang merugikan bagi pemilik data. Melihat hal tersebut, keamanan data dalam sebuah sistem informasi menjadi sangatlah penting. Penelitian ini bertujuan untuk merancang sebuah keamanan data pada database sistem informasi dengan penerapan Hasing SHA1 dan Algoritma Asimetris RSA. Algoritma Asimetris RSA digunakan untuk mengacak suatu kata dari sebuah data yang digunakan oleh user dan fungsi Hashing SHA1 mengenskripsi data yang telah di acak sebelumnya. setelah melalui proses enskripsi data, barulah hasil enskripsi data tersebut selanjutnya disimpan dalam database. Data yang tersimpan dalam database tidak dapat terbaca secara langsung oleh pengguna lain karena data tersebut menjadi kode – kode yang acak. Dengan penerapan Hasing SHA1 dan Algoritma Asimetris RSA data yang tersimpan akan terlindungi dari pihak – pihak yang ingin menyalahgunakan data tersebut.*

Kata kunci – Algoritma Asimetris RSA; Keamanan Sistem Informasi; SHA1

Abstract *- The use of information technology in obtaining data in the era of society 5.0 cannot be avoided. Almost all of the data we use is stored in a database on an information system that can be accessed online and can be used by any user. Information systems are very vulnerable if the data we have is used or accessed by other people and uses the data for things that are detrimental to the data owner. Seeing this, data security in an information system is very important. This study aims to design a data security in an information system database with the application of Hashing SHA1 and RSA Asymmetric Algorithm. RSA Asymmetric Algorithm is used to scramble a word from a data used by the user and the SHA1 Hashing function encrypts the data that has been randomized before. After going through the data encryption process, the results of the data encryption are then stored in the database. The data stored in the database cannot be read directly by other users because the data becomes random codes. With the application of Hashing SHA1 and RSA Asymmetric Algorithm, stored data will be protected from parties who want to misuse the data.*

Keywords – Information System Security; RSA Asymmetric Algorithm; SHA1

I. PENDAHULUAN

Perkembangan teknologi informasi saat ini sangat pesat. Kebutuhan manusia akan akses informasi yang cepat menuntut kita untuk memanfaatkan teknologi yang ada saat ini. Informasi saat ini sangat berharga. Sejak adanya internet, informasi tidak lagi dibatasi. Fungsi internet sebagai gudang informasi adalah menyediakan informasi apapun, seperti informasi tentang apa saja yang ada di seluruh penjuru dunia, dan penggunaan teknologi informasi dan komunikasi sekarang menjadi cara transmisi informasi yang efektif dan efisien[1][2].

Masalah keamanan data merupakan masalah yang sangat serius dalam kegiatan penggunaan sistem informasi di era Society 5.0. Hampir semua data yang kita gunakan tersimpan dalam database pada sebuah sistem informasi yang dapat di akses secara online dan dapat dipergunakan oleh sembarang pengguna[3]. Database merupakan sekumpulan informasi yang disimpan di dalam komputer secara sistematik yang dapat digunakan melalui sebuah program komputer tertentu untuk menjalankannya[4]. Keamanan database

menjadi solusi terakhir pada saat aplikasi komputer mengalami gangguan yang disebabkan oleh pihak luar setelah berhasil melewati keamanan jaringan komputer dan keamanan aplikasi komputer[5][6].

Dalam penggunaan sistem informasi, keamanan database menjadi masalah utama setelah menjamin keamanan pada jaringan komputer dan aplikasi komputernya[2]. Contohnya pada kegiatan bisnis penjualan peralatan elektronik yaitu database akan menyimpan data-data penjualan, seperti data produk dan data pelanggan serta hak akses pelanggan, dan jika kegiatan bisnis tersebut terintegrasi dengan identitas data lainya antara lain kartu kredit maka harus menjamin kerahasiaan identitas data karena menyangkut privasi pelanggan[7]. Semua yang menyangkut data pribadi pelanggan harus dijamin kerahasiaannya bahkan dari karyawan sekalipun tidak mempunyai hak untuk mengakses atau melihat data identitas pelanggang tersebut. Database yang aman akan menjadi tolok ukur sebuah perusahaan dalam keberlangsungan kegiatan bisnis yang dilakukan perusahaan tersebut[8].

Keamanan database dapat dilakukan dengan beberapa cara contohnya pembatasan hak akses pada database tersebut, pengunaan nama field data yang hanya pahami oleh pemilik aplikasi dan tidak terdapat pegawai yang dapat mengakses database dan memahami alur database yang ada sehingga terhindar dari manipulasi data dan lainnya, serta menerapkan metode kriftografi pada aplikasi terhadap field data di dalam databasenya dengan tujuan field data yang disimpan menjadi lebih terjamin privasinya dan tidak dapat dimengerti oleh pihak luar mapun pihak dalam.

Kriptografi sendiri merupakan ilmu dan sekaligus seni untuk mengamankan data yang didalamnya terdapat algoritma tertentu yang bertujuan sebagai confusion atau pembingungan, dengan cara mengubah teks polos (plaintext) menjadi teks yang tidak bisa dibaca artinya secara langsung oleh manusia atau teks rahasia (ciphertext)[9]. Kriptografi mempunyai proses enkripsi dimana dapat mengubah teks atau data (plaintext) menjadi teks rahasia (ciphertext), kemudian sebaliknya proses deskripsi yang dapat mengembalikan teks rahasia (ciphertext) menjadi teks atau data (plaintext)[10]. Dalam proses ini digunakan kunci rahasia, semakin banyak kunci rahasia yang digunakan maka semakin bagus. Algoritma kriptografi diklasifikasikan menjadi dua yaitu algoritma simetris dan algoritma asimetris. Contoh algoritma kriptografi Asimetris yaitu algoritma RSA.

Secure Hash Algorithm (SHA) merupakan fungsi Hash yang bersifat "tidak dapat diubah kembali" menjadi pesan semula (satu arah) yang akan menghasilkan sebuah checksum atau fingerprint dari data tersebut. Umumnya dipergunakan untuk data integration dan authentication. Kelebihan fungsi Hash yaitu menjaga integritas data, hemat dalam waktu pengiriman serta menormalkan panjang data yang beraneka ragam[11][12]. Fungsi Hash yang dipakai untuk penelitian ini adalah SHA1 yang memetakan inputan string dengan panjang sembarang menjadi suatu nilai hash dengan panjang tetap yaitu 160 bit. Ukuran internal state pada SHA 1 adalah 160 bit, sedangkan ukuran bloknya adalah 64 bytes.

Diharapkan dengan Penerapan Hashing SHA1 dan Algoritma Asimetris RSA untuk Keamanan Data pada Sistem Informasi yang dapat menjadi pijakan dasar dalam pengembangan keamanan website kampus di masa mendatang.

II. LANDASAN TEORI

Kriptografi adalah istilah yang berasal dari bahasa Yunani dan terdiri dari dua kata yaitu kata cryptos dan graphein, kata cryptos berarti bersembunyi, sedangkan graphein berarti menulis. Fokus bidang ilmu komputer dan matematika adalah teknologi yang digunakan untuk mengamankan komunikasi antara dua pihak di hadapan pihak ketiga[9][13]. Oleh karena itu, penggunaan teknologi enkripsi dapat melindungi data dari serangan atau pencurian data oleh pihak yang tidak bertanggung jawab. Ada dua jenis enkripsi yaitu enkripsi klasik (algoritma yang hanya menggunakan satu kunci untuk melindungi data) dan enkripsi modern, dimana enkripsi modern mempunyai kompleksitas yang sangat kompleks karena proses enkripsi dan dekripsi dioperasikan oleh komputer.

Secure Hash Algorithm (SHA) merupakan fungsi Hash yang bersifat "tidak dapat diubah kembali" menjadi pesan semula (satu arah) yang akan menghasilkan sebuah checksum atau fingerprint dari data tersebut[14]. Umumnya dipergunakan untuk data integration dan

authentication. Kelebihan fungsi Hash yaitu menjaga integritas data, hemat dalam waktu pengiriman serta menormalkan panjang data yang beraneka ragam. Fungsi Hash yang dipakai untuk paper ini adalah SHA1 yang memetakan inputan string dengan panjang sembarang menjadi suatu nilai hash dengan panjang tetap yaitu 160 bit. Ukuran internal state pada SHA 1 adalah 160 bit, sedangkan ukuran bloknya adalah 64 bytes.

RSA adalah algoritma kriptografi asimetris. Ini pertama kali ditemukan pada tahun 1977 oleh Ron Rivest, Adi Shamir dan Leonard Adleman. Nama RSA sendiri diambil dari singkatan Ron Rivest, Adi Shamir dan Leonard Adleman. Nama RSA sendiri diambil dari singkatan nama ketiga penemunya. Sebagai algoritme kunci publik, RSA memiliki dua kunci, kunci publik dan kunci privat[7][15]. Kunci publik dapat diketahui siapa saja dan digunakan dalam proses enkripsi.

Sedangkan kunci pribadi hanya pihak - pihak tertentu saja yang boleh mengetahuinya, dan digunakan untuk proses dekripsi. Algoritma RSA masih digunakan hingga pada saat ini seperti yang diuraikan M. Zaki Riyanto dan Ardhi Ardhian: Keamanan sandi RSA terletak pada sulitnya memfaktorkan bilangan yang besar. Sampai saat ini RSA masih dipercaya dan digunakan secara luas di internet[16].

Untuk membangkitan pasangan kunci RSA, digunakan algoritma sebagai berikut:

1. Dipilih dua buah bilangan prima sembarang yang besar, p dan q.
   Nilai p dan q harus dirahasiakan.
2. Dihitung n = p x q.
   Besaran n tidak perlu dirahasiakan.
3. Dihitung fungsi Euler's totient
   $(n) = (p - 1)(q - 1)$
4. Dipilih sebuah bilangan bulat sebagai kunci publik, disebut namanya e, yang relatif prima terhadap (n). e relatif prima terhadap (n) artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut gcd (e, (n)) = 1.
5. Dihitung diterminasi d dengan rumus
   $d = e^{-1} \pmod{(n)}$,
   d adalah multiplikasi invers dari e mod (n)
6. d sebagai komponen kunci private sehingga e * d mod (n) = 1
7. Kunci pablik mengandung modulo n dan eksponen e, sehingga (e,n)
8. Kunci privat mengandung modulo n dan eksponen d, sehingga (d,n)

Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan non prima menjadi faktor primanya, yang dalam hal ini n = p x q. Jika n berhasil difaktorkan menjadi p dan q, maka m = (p – 1)(q – 1) dapat dihitung. Dan karena kunci enkripsi e telah diumumkan (tidak dirahasiakan),

maka kunci dekripsi d dapat dihitung melalui persamaan (d x e) mod n = 1. Selama belum ditemukan cara untuk memfaktorkan bilangan besar menjadi faktor-faktor primanya, maka selama itu pula keamanan algoritma RSA terjamin.

Penemu algoritma RSA menyarankan nilai p dan q panjangnya lebih dari 100 digit. Dengan demikian hasil kali n = p x q akan berukuran lebih dari 200 digit. Dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mem-punyai kecepatan 1 milidetik, menurut Rivest dan kawan-kawan, usaha untuk men-cari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun. Adapun Proses Enkripsi dan dekripsi RSA antara lain sebagai berikut :

1. Proses Enkripsi
   a. Nyatakan pesan menjadi blok-blok plainteks: m_1, m_2, m_3, … (harus dipenuhi persyaratan bahwa nilai m_i harus terletak dalam himpunan nilai 0, 1, 2, …, $n - 1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan)
   b. Hitung blok cipherteks c_i untuk blok plainteks p_i dengan persamaan
   $c_i = m^e \pmod{n}$ dalam hal ini,
   e adalah i Kunci Publik.

2. Proses Dekripsi

Proses dekripsi dilakukan dengan menggunakan persamaaan $m_i = c_i \pmod{n}$, yang dalam hal ini, d adalah kunci private.

III. METODE

Metode yang digunakan dalam penelitian ini adalah metode eksperimental dan studi pustaka. Penelitian dimulai dengan studi dokumen yang diperoleh dari buku, e-book dan materi terkait penelitian. kemudian Eksperimen dilakukan dengan mengimplementasikannya sebagai

bahasa pemrograman untuk mewujudkan algoritma sha1 hash dan rsa asimetris untuk keamanan data pada sistem informasi berbasis web.

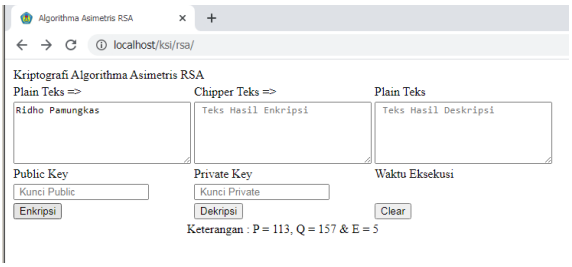
IV. HASIL DAN PEMBAHASAN

Hasil dari penerapan hashing sha1 dan algoritma asimetris rsa untuk keamanan data pada sistem informasi berbasis web. Implementasi Algoritma Asimetris RSA bisa dilihat pada sebagian source code berikut.

Fungsi.php

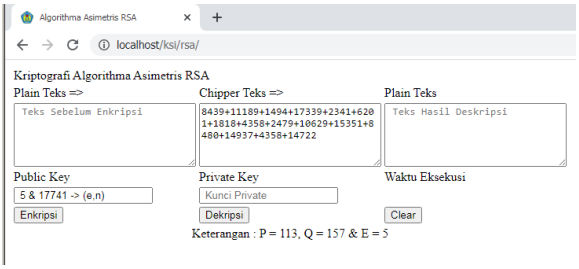
```
  <?php
  /*
  -- keterangan Masing Masing Fungsi yang dipake dari Library gmp & PHP --

  gmp_mod    = Sisa Hasil Bagi;
  gmp_pow        = Raise number into power;

  ord            = Mengembalikan Nilai Karakter Ke ASCII;
  strlen          = Dapatkan Panjang String;

  gmp_strval   =   Convert   Nomer   ke String;

  chr         = Mengembalikan ke Nilai Karakter;

  explode    = Memecah String;

  */

  // Proses Enkripsi
  function enkripsi($plain, $n, $e){
  //Pesan dikodekan menjadi kode ASCII, Kemudian di Enkripsi Per Karakter
       $hasil='';

     for($i=0;      $i<strlen($plain); ++$i){

          //Rumus     Enkripsi     ---> ChiperTeks = <pesan>^<e>mod<n>

$hasil.=gmp_strval(gmp_mod(gmp_pow(ord ($plain[$i]),$e),$n));

          //Antar     Tiap     Karakter dipisahkan dengan "+"
          if($i!=strlen($plain)-1){
              $hasil.="+";

          }
      }

  return $hasil;
  }

  // Proses Dekripsi
  function deskripsi($chiper, $d, $n){
      $time_start = microtime(true);
      //Pesan Enkripsi dipecah menjadi array dengan batasan "+"
       $hasildekrip = '';

       $hasil = '';

      $dekrip=explode("+", $chiper);

      foreach($dekrip as $nilai){
       //Rumus Deskripsi ---> PlainTeks = <enkripsi>^<d>mod<n>
       $gm1                              = gmp_pow($nilai,gmp_strval($d));

       $gm2 = gmp_mod($gm1,$n);

       $gm3 = gmp_strval($gm2);

      $hasildekrip.=chr($gm3);
      }
       $time_end = microtime(true);

       // Waktu Eksekusi
       $time = $time_end - $time_start;

       $hasil       =       array($time, $hasildekrip);

  return $hasil;
  }

?>
```

Uji Coba Algoritma Asimetris RSA
Pertama kita masukkan kata yang akan kita encrypt atau bisa disebut dengan Plain Teks pada kolom Plain Teks sesuai gambar 1.



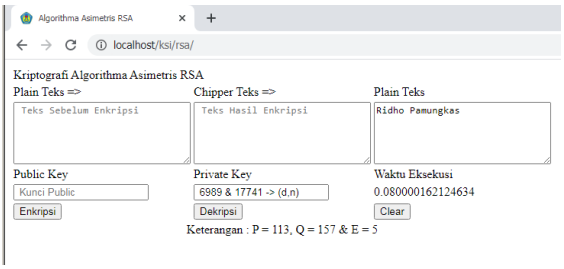
Gambar 1. Input Plain Teks

Setelah itu klik tombol encrypt, maka akan muncul data Chipper Teks atau karakter hasil dari enkripsi beserta Public Key dari Algoritma Asimetris RSA pada gambar 2.



Gambar 2. hasil dari enkripsi

Setelah data terenkripsi, pada saat data tersebut akan digunakan maka harus di deskripsi agar kembali lagi ke data aslinya. Untuk mengembalikan ke data aslinya, klik tombol dekripsi yang terlihat pada gambar 3. Lalu munculah data asli pada kolom plain Teks gambar 3.

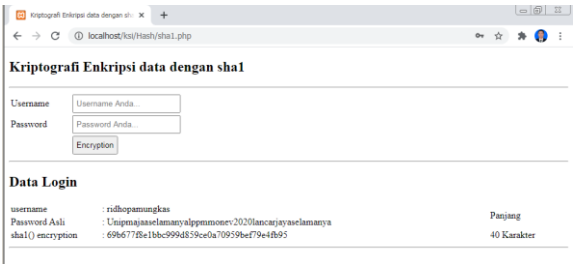


Gambar 3. Dekripsi Data Uji Coba Hashing SHA1

Pertama kita masukkan data Login Username dan Password yang akan kita encrypt menggunakan Teknik hashing dengan SHA1 pada gambar 4. Setelah kita klik encryption, maka akan muncul hasil data login yang asli dan yang sudah terencrypt pada gambar 4. Terlihat bahwa dari kata ridhopamungkas menjadi sebuah karakter acak dengan Panjang 40 karakter.

Gambar 4. Hasil encrypt dengan SHA1

Selanjutnya kita uji coba data input dengan Panjang lebih dari 40 karakter. Gambar 5 menunjukkan hasil dari encrypt tetap sama, yaitu karakter acak dengan Panjang tetap 40 karakter.



Gambar 5. Hasil encrypt dengan SHA1 menggunakan data lebih dari 40 karakter.

V. KESIMPULAN

Hasil dari implementasi dan uji coba penerapan hashing sha1 dan algoritma asimetris RSA pada data sistem informasi, data telah berhasil ter-*encrypt* dari data aslinya. Dengan penggunaan Hashing SHA1 dapat me-encrypt data user untuk login dengan karakter acak sepanjang 40 karakter dengan nilai tetap, tidak akan melebihi 40 karakter walaupun karakter aslinya lebih panjang dari 40 karakter. Pada algoritma asimetris RSA digunakan untuk mengencrypt data dengan rumus menggunakan bilangan primer tertentu yang sudah ditentukan terlebih dahulu lalu menyimpannya ke dalam database sesuai dengan data yang telah terencrypt dan melakukan decrypt pada saat data digunakan kembali, sehingga data yang terlihat dalam database tidak seperti data asli.

DAFTAR PUSTAKA

[1] R. Pamungkas and S. Saifullah, "Evaluasi Kualitas Website Program Studi Sistem Informasi Universitas PGRI Madiun Menggunakan Webqual 4.0," *INTENSIF J. Ilm. Penelit. dan Penerapan Teknol. Sist. Inf.*, vol. 3, no. 1, p. 22, Feb. 2019, doi: 10.29407/intensif.v3i1.12137.

[2] R. Pamungkas, "OPTIMALISASI QUERY DALAM BASIS DATA MY SQL MENGGUNAKAN INDEX," *Res. J. Comput. Inf. Syst. Technol. Manag.*, vol. 1, no. 1, pp. 27–31, 2018.

[3] B. U. Customer, "Analisis Pengaruh Sistem Keamanan Informasi Perbankan pada Nasabah Pengguna Internet Banking," vol. 3, no. 1, pp. 1–9, 2019.

[4] R. Pamungkas, "Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index," *Res. Comput. Inf. Syst. Technol. Manag.*, Apr. 2018, doi: 10.25273/research.v1i1.2453.

[5] C. Asiminidis, G. Kokkonis, and S. Kontogiannis, "DATABASE SYSTEMS PERFORMANCE EVALUATION FOR IOT APPLICATIONS," *Int. J. Database Manag. Syst. (IJDMS*, vol. 10, no. 6, 2018, doi: 10.5121/ijdms.2018.10601.

[6] M. S. Ramadhan and F. Ariyani, "PENINGKATAN KEAMANAN LOGIN WEBSITE DENGAN IMPLEMENTASI ONE TIME PASSWORD MENGGUNAKAN ALGORITMA SHA1 DAN MD5 BERBASIS MOBILE," May 2018.

[7] S. Susanto and A. A. Trisusilo, "Penerapan Algoritma Asimetris Rsa Untuk Keamanan Data Pada Aplikasi Penjualan Cv. Sinergi Computer Lubuklinggau Berbasis Web," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 9, no. 2, pp. 1043–1052, 2018, doi: 10.24176/simet.v9i2.2537.

[8] F. Ramadhani, U. Ramadhani, P. Nasution, and L. Basit, "Combination Of Hybrid Cryptography In One Time Pad (OTP) Algorithm And Keyed-Hash Message Authentication Code (HMAC) In Securing The Whatsapp Communication Application," Mar. 2020.

[9] Y. Bin Idris, S. Adli Ismail, N. F. Mohd Azmi, A. Azmi, and A. Azizan, "Enhancement Data Integrity Checking Using Combination MD5 and SHA1 Algorithm in Hadoop Architecture," *J. Comput. Sci. Comput. Math.*, vol. 7, no. 3, pp. 99–102, 2017, doi: 10.20967/jcscm.2017.03.007.

[10] A. B. Nasution, "Implementasi Pengamanan Data Dengan Menggunakan Algoritma Caesar Cipher Dan Transposisi Cipher," *J. Teknol. Inf.*, vol. 3, no. 1, p. 1, 2019, doi: 10.36294/jurti.v3i1.680.

[11] D. Tiwari, A. Singh, and A. Prabhakar, "Performance Analysis of AES, RSA and Hashing Algorithm Using Web Technology," Springer, Singapore, 2020, pp. 413–418.

[12] R. Prasetyo and A. Suryana, "Aplikasi Pengamanan Data dengan Teknik Algoritma Kriptografi AES dan Fungsi Hash SHA-1 Berbasis Desktop," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 5, no. 2, p. 61, 2016, doi: 10.32736/sisfokom.v5i2.40.

[13] P. Chyan, "Penerapan Sistem Kriptografi Enkripsi Jamak Dan Tanda Tangan Digital Dalam Mendukung Keamanan Informasi," *J. Temat.*, vol. 6, no. 1, pp. 39–46, 2018.

[14] F. G. N. Larosa, J. F. Naibaho, and R. M. Tarigan, "Web Storage Berbasis Private Cloud Menggunakan Enkripsi Sha1," *J. METHOMIKA*, vol. 4, no. 1, pp. 56–59, 2020, [Online]. Available: http://www.methomika.net/index.php/jmika/article/view/142/81.

[15] Sumarno, I. Gunawan, H. S. Tambunan, and E. Irawan, "Analisis Kinerja Kombinasi Algoritma Message-Digest Algorithm 5 (MD5), Rivest Shamir Adleman (RSA) dan Rivest Cipher 4 (RC4) pada Keamanan E-Dokumen," *JUSIKOM PRIMA (Jurnal Sist. Inf. Ilmu Komput. Prima)*, vol. 2, no. 1, pp. 41–48, 2018.

[16] M. Z. Riyanto and A. Ardian, "Kriptografi Kunci Publik: Sandi RSA.," *J. Kelompok Stud. Sandi*, 2008.