EULER: Jurnal Ilmiah Matematika, Sains dan Teknologi
Journal Homepage: http://ejurnal.ung.ac.id/index.php/Euler
DOI: https://doi.org/10.37905/euler.v11i2.21169
Desember 2023, Vol. 11, No. 2, pp. 205-215

Penerapan Algoritma *Rivest-Shamir-Adleman* (RSA) pada Enkripsi *Uniform Resource Locator* (URL) *Website* untuk Keamanan Data

Theodora Tantri Trisnawati[1*], Sherli Yurinanda[2], Wardi Syafmen[3], Cut Multahadah[4]

[1,2,4]Program Studi Matematika, Universitas Jambi, Mendalo Darat 36361, Jambi, Indonesia
[3]Program Studi Pendidikan Matematika, Universitas Jambi, Mendalo Darat 36361, Jambi, Indonesia

**Penulis Korespondensi. Email: trisnawatitantri25@gmail.com*

Abstrak

PT. Rezeki Surya Gasindo merupakan salah satu perusahaan yang memanfaatkan website untuk menyimpan data-data penting perusahaan salah satunya data pribadi pelanggan. *Website* PT. Rezeki Surya Gasindo dilindungi oleh sistem *login*, namun dengan sistem *login* saja belum cukup untuk melindungi data-data yang disimpan dalam *website* tersebut dari kasus pencurian data oleh pihak ketiga. Adapun salah satu solusi untuk menangani masalah ini adalah dengan melakukan enkripsi pada *Uniform Resource Locator* (URL) *website* demi meningkatkan tingkat keamanan data yang disimpan dalam *website* tersebut. Dalam penelitian ini algoritma yang digunakan untuk proses enkripsi adalah Algoritma *Rivest-Shamir-Adleman* (RSA). Tujuan dari penelitian ini yaitu untuk mengetahui proses penerapan algoritma *Rivest-Shamir-Adleman* (RSA) pada enkripsi *Uniform Resource Locator* (URL) *website* PT. Rezeki Surya Gasindo. Keberhasilan pengaplikasian enkripsi dengan Algoritma RSA diamati dari perubahan *value* parameter *get* yang muncul pada URL *bar*. Adapun pesan yang dienkripsi adalah *customer*, yang merupakan parameter *get* yang muncul pada menu Customer yang berisi data-data pribadi konsumen. Dengan memilih dua buah bilangan prima besar yaitu 151 dan 173, dan mengambil salah satu kunci publik/kunci enkripsi yaitu 16379, diperoleh hasil bahwa parameter *get* pada URL *bar* telah berubah menjadi deretan kode 5a9cb05811aa6e4c. Artinya, Algoritma RSA telah berhasil diterapkan pada URL *website*.

Kata Kunci: Website; Enkripsi; URL; Algoritma RSA

Abstract

PT. Rezeki Surya Gasindo is a company that uses websites to store important company data, one of which is customer personal data. PT. Rezeki Surya Gasindo's website is protected by a login system. However, a login system alone is not enough to protect the data stored on the website from cases of data theft by third parties. One solution to this problem is to encrypt the website's Uniform Resource Locator (URL) to increase the security level of data stored on the website. In this research, the Rivest-Shamir-Adleman (RSA) algorithm is used for the encryption process. The aim of this research is to determine the process of applying the Rivest-Shamir-Adleman (RSA) algorithm to the Uniform Resource Locator (URL) encryption of the PT. Surya Gasindo's website. The success of applying encryption with the RSA algorithm is observed from changes in the get parameter value that appears in the URL bar. The encrypted message is the customer, the get parameter in the Customer menu, which contains the consumer's personal data. By choosing two large prime numbers, namely 151 and 173, and taking one of the public keys/encryption keys, namely 16379, the result is that the get parameter in the URL bar has changed to the code string 5a9cb05811aa6e4c. The RSA algorithm has been successfully applied to the website URL.

Keywords: *Website; Encryption; URL; RSA Algorithm*

1. Pendahuluan

Proses penginputan data, pencarian data serta pembuatan laporan secara manual mempunyai resiko kesalahan yang cukup tinggi apalagi dalam menangani data–data yang cukup kompleks dan

Submit: 20 Juli 2023, Direvisi: 5 Oktober 2023, Disetujui: 15 Oktober 2023

cukup besar. Proses pencarian data dengan cara konvensional memerlukan waktu yang lama. Oleh karena itu, diperlukan sebuah sistem yang lebih praktis, yaitu dengan memanfaatkan sistem yang sudah terkomputerisasi, dimana semua urusan dapat dengan cepat dan efisien dikerjakan menggunakan komputer[1]. Salah satu sistem terkomputerisasi yang digunakan perusahaan dalam mengolah data mereka adalah *website*. *Website* menjadi salah satu perangkat yang sangat diperlukan oleh perusahaan di era teknologi saat ini. Penggunaan *website* menjadi salah satu media pemasaran, penjualan dan pelayanan pada suatu bisnis karena faktor biaya yang murah, kemudahan akses hingga 24 jam penggunaan[2]. Dengan berbagai macam kelebihan yang ditawarkan oleh *website*, terdapat beberapa kelemahan dari *website* yaitu keamanan data yang tersimpan didalamnya. Dalam dunia bisnis, data bersifat sangat rahasia. Informasi data pribadi seseorang merupakan hal yang sangat rahasia sehingga dalam penggunaanya diperlukan kehati-hatian yang sangat ketat agar orang lain tidak dapat menyalahgunakannya. Keamanan data konsumen juga sangat mempengaruhi kepuasan konsumen terhadap layanan perusahaan tersebut[3]. Masalah keamanan data sangatlah komplek. Seringkali masalah keamanan data dapat melibatkan aspek hukum, sosial atau etika. Keamanan data berkaitan dengan perlindungan data terhadap ancaman yang disengaja atau tidak disengaja, dengan menggunakan elemen kontrol peralatan komputasi atau yang tidak[4]. Karena masalah keamanan data ini, data organisasi dapat berada dalam bahaya karena potensi ancaman, kehilangan data, dan manipulasi data. Penyerang dapat bekerja di *browser* sistem dan menghancurkan data yang terkait. Kurangnya fitur keamanan perangkat memungkinkan penyerang melacak detail sistem dan melakukan serangan dengan memanipulasi atau mencuri data. Oleh karena itu, penggunaan data antar sistem memerlukan pengamanan dari serangan *cyber*[5].

Dalam pemrograman *web* terdapat dua metode dalam mengirimkan data dari *client* ke *server* yaitu metode POST dan GET[6]. Metode POST mengirimkan data pada sebuah permintaan yang mengeksekusi data yang tidak terlihat pengguna, sedangkan metode GET menampilkan *value* data pada URL *bar* yang menyebabkan metode GET rentan terhadap serangan *cyber* [7], [8]. Data pribadi konsumen yang tersimpan dalam *website* menjadi rawan terhadap penyalahgunaan oleh pihak yang tidak bertanggung jawab, contohnya seperti tindak kriminal pembobolan atau pencurian data warga Indonesia yang terjadi pada tahun 2022 lalu oleh *hacker* asal Australia, Bjorka. Salah satu kasus pencurian yang dilakukan Bjorka adalah pencurian data pelanggan IndiHome sebanyak 26 juta data yang diperjualbelikan di BreachForums pada Agustus 2022, data yang dicuri berupa histori pencarian, *keyword*, *user* info mencakup email, nama, jenis kelamin, hingga NIK [9].

Kasus pembobolan data dapat memberikan dampak yang sangat besar bagi perusahaan seperti berpengaruh pada kinerja bisnis, menurunnya reputasi perusahaan, menurunnya tingkat kepercayaan pelanggan, dan dalam beberapa kasus berakhir dengan jatuhnya perusahaan tersebut[10]. Melihat beragam kasus pencurian data dan kerugian yang ditimbulkan, maka dapat disimpulkan bahwa keamanan data sangatlah penting terlebih untuk sebuah perusahaan atau instansi. Oleh sebab itu, perlu adanya pengamanan untuk mengatasi solusi ini.

Salah satu ilmu yang berkaitan dengan keamanan data adalah kriptografi. Kriptografi merupakan ilmu matematika yang mempelajari tentang pengacakan pesan yang bertujuan untuk menjaga keamanan pesan tersebut. Enkripsi merupakan cara menubah data atau pesan yang dikirimkan menjadi deretan kode dengan menggunakan algoritma yang sistematis untuk memrosesnya. Sedangkan dekripsi merupakan kebalikannya, yaitu proses mengembalikan deretan kode yang diterima menjadi pesan seperti semula. Secara umum, kriptografi memiliki dua jenis kunci, yaitu kunci simetris dan kunci asimetris. Kunci simetris menggunakan kunci yang sama untuk proses enkripsi dan dekripsi, sedangkan kunci asimetris menggunakan kunci yang berbeda untuk melakukan proses enkripsi dan dekripsi[11]. Kelemahan kunci simetris terletak pada proses pendistribusian kunci dari pengirim ke si penerima, karena kunci yang digunakan untuk proses enkripsi dan dekripsi sama, sehingga jika kunci tersebut dapat dipecahkan maka deretan kode cipherteks dapat dengan mudah dipecahkan. Sedangkan, kunci asimetris menggunakan dua buah kunci yang berbeda pada proses enkripsi dan dekripsi, oleh sebab itu kunci asimetris lebih aman dibandingkan kunci simetris [12].

Salah satu algoritma kriptografi yang paling tekenal adalah algoritma RSA. Algoritma RSA merupakan algoritma krunci asimetris, yang mana menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi. Algoritma RSA merupakan algoritma pertama yang cocok untuk *digital signatur* seperti halnya enkripsi dan merupakan algoritma yang paling maju dibandingkan kunci asimetris lainnya[13]. Algoritma RSA terkenal dengan tingkat keamanannya, dikarenakan untuk mendekripsi pesan yang telah dienkripsi maka terlebih dahulu harus menemukan kunci rahasia. Kunci rahasia sendiri dibentuk dari perkalian dua buah bilangan prima dan aritmetika modulo. Dan untuk memecahkan kunci rahasia maka perlu memfaktorkan dua buah bilangan prima dari sebuah bilangan non-prima. Pada dasarnya, memfakorkan bilangan non-prima menjadi faktor primanya bukanlah pekerjaan yang mudah dan hingga kini belum ditemukan algoritma yang efisien untuk memfaktorkannya. Semakin besar bilangan non-primanya, maka semakin sulit pula untuk difaktorkan[14].

Salah satu penelitian yang terkait dengan keamanan data dan kriptografi, yaitu penelitian yang dilakukan oleh Gat pada tahun 2018 dengan judul "Mencegah Exploit URL *Website* Sensitek STMIK Pontianak dengan Algoritma Blowfish". Pada penelitian ini telah dilakukan proses enkripsi dan dekripsi pada parameter GET yang muncul pada URL *bar* dengan algoritma Blowfish. Dan diperoleh hasil bahwa algoritma Blowfish telah berhasil diterapkan untuk enkripsi dan dekripsi URL *website*. Hasil ini diperoleh dengan menerapkan algoritma Blowfish pada *website* uji coba Sensitek STMIK. Dalam kasus ini yang dienkripsi hanya value dari parameter GET pada URL tersebut. Keberhasilan enkripsi dan dekripsi ini dilihat dari perubahan pada *value* parameter GET pada URL.

Dengan demikian, penelitian ini mengambil studi kasus pada PT. Rezeki Surya Gasindo (RSG) merupakan salah satu perusahaan yang memanfaatkan *website* sebagai alat untuk menyimpan data-data penting perusahaan, salah satunya data pribadi pelanggan. *Website* PT. RSG telah dilindungi oleh sistem *login*, namun sayangnya *website* tersebut memiliki parameter GET yang muncul pada URL *bar* yang mengakibatkan *website* tersebut sangat rentan terhadap pembobolan data. Dalam penelitian Fajrin et al. pada tahun 2023 yang berjudul "Analisis Performa dari Algoritma Kriptografi RSA dan ElGamal dalam Enkripsi dan Dekripsi Pesan" disebutkan bahwa Algoritma RSA lebih unggul dibandingkan dengan algoritma kunci asimetris lainnya yaitu ElGamal, baik dalam waktu pengoperasian hingga pemakaian ruang memori, RSA unggul dalam kedua hal ini dibandingkan ElGamal. Dalam penlitian tersebut juga disebutkan bahwa algoritma RSA akan sulit untuk dipecahkan karena harus menemukan faktor dari dua buah bilangan prima yang besar. Oleh sebab itu, pada penelitian ini dilakukan proses enkripsi pada parameter GET yang muncul pada URL *bar* dengan menggunakan algoritma RSA untuk meningkatkan keamanan data yang tersimpan didalamnya. Tujuan dari penelitian ini adalah untuk mengetahui proses enkripsi dengan algoritma RSA yang diterapkan pada parameter GET yang muncul pada URL *bar*.

2. Metode Penelitian

Penelitian ini dilakukan di PT. RSG yang merupakan salah satu perusahaan yang memanfaatkan *website* sebagai media penyimpanan perusahaan dan terdapat parameter GET pada *URL bar* yang menyebabkan *website* milik PT. RSG rentan terhadap kasus pembobolan. Adapun batasan-batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Website yang digunakan adalah website uji coba/localhost yang dibentuk berdasarkan website asli milik PT. Rezeki Surya Gasindo.
2. Enkripsi hanya dilakukan pada menu *Customer*, karena hanya pada menu ini yang berisi data-data pribadi konsumen.
3. URL website yang telah dienkripsi tidak didekripsi kembali, untuk menjaga keamanan *website* tersebut.

Pada penelitian ini, proses enkripsi dengan algoritma RSA diterapkan pada menu *Customer* yang berisi data-data pribadi konsumen. Keberhasilan penerapan algoritma ini dapat diamati perubahan yang terjadi pada parameter GET yang muncul di *URL bar*. Adapun pesan yang dienkripsi

atau plainteks pada penelitian ini adalah *customer* yang merupakan parameter GET yang muncul di *URL bar* pada menu *Customer*.

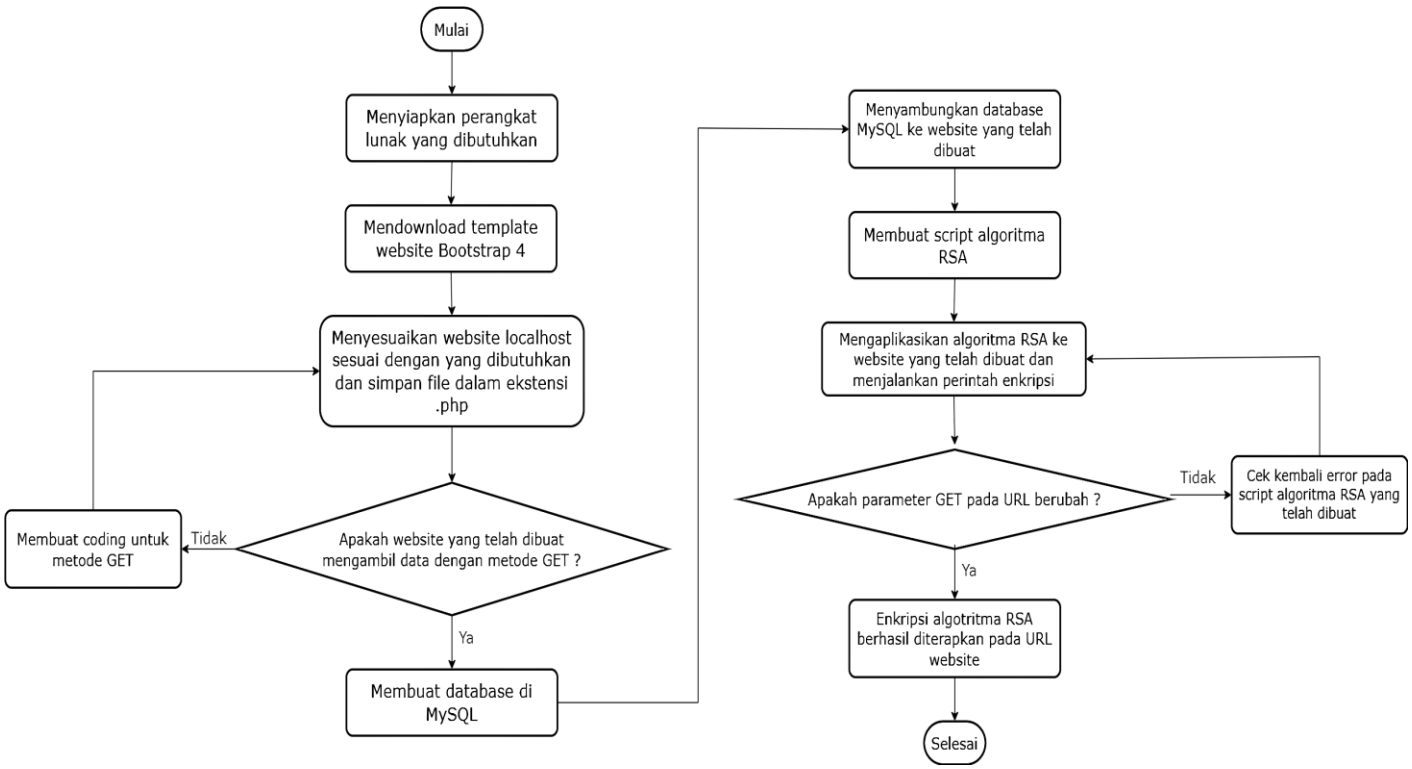
Berikut algoritma enkripsi dengan RSA yang digunakan pada penelitian ini :

1. Pilih nilai *p* dan *q*, dengan p dan q merupakan bilangan prima yang lebih besar dari 100.
2. Hitung nilai $n = p \times q$.
3. Hitung $\phi(n)$.
4. Pilih kunci publik *h* dengan *h* relatif prima terhadap $\phi(n)$, $\left(\gcd\left(h, \phi(n)\right) = 1\right)$ dan $1 < h < \phi(n)$.
5. Ubah plainteks ke dalam bentuk desimal ASCII berdasarkan Tabel ASCII, kemudian pecah plainteks menjadi beberapa blok $m_1, m_2, \ldots, m_i$, dimana setiap m_i mempresentasikan nilai yang berada dalam interval $0 \leq m_i \leq n - 1$.
6. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus :
$$c_i = m_i^h \ mod \ n$$
7. Ubah cipherteks ke bentuk heksadesimal ASCII berdasarkan Tabel ASCII.

Sebelum menerapkan algoritma RSA, terlebih dahulu perlu dibentuk sebuah *website localhost* yang menjadi *website* uji coba pada penelitian ini. Berikut merupakan alur penelitian dalam penelitian ini :

1. Menyiapkan perangkat lunak pendukung yang dibutuhkan untuk proses enkripsi URL *website*.
2. Mendownload template *website localhost* yang tersedia secara gratis melalui situs *online* yaitu Bootstrap 4.
3. Menyesuaikan *website localhost* sesuai dengan yang dibutuhkan, dan menyimpan file dalam ekstensi file .php.
4. Memastikan *website* yang telah dibuat mengambil data dengan metode GET. Hal ini dapat dilihat pada bar URL *website*, apakah terdapat parameter GET atau tidak. Jika URL *website* sudah terdapat parameter GET maka penelitian dapat dilanjutkan ke tahap selanjutnya. Jika tidak, maka harus membuat *coding* untuk mengambil data dengan metode GET terlebih dahulu dan sisipkan pada file *website* yang telah dibuat pada langkah ke-3.
5. Jika langkah ke-4 sudah terpenuhi, langkah selanjutnya adalah membuat *database* di MySQL.
6. Menyambungkan *database* MySQL ke file *website* yang telah dibuat sebelumnya.
7. Membuat *script* yang berisi algoritma RSA menggunakan PHP.
8. Mengaplikasikan *script* algoritma RSA ke file *website* yang telah dibuat dan menjalankan perintah enkripsi pada file *website*.
9. Selanjutnya, mengecek keberhasilan program enkripsi ini dengan melihat perubahan *value* parameter GET pada URL *website*. Jika *value* parameter GET telah berubah menjadi cipherteks maka proses enkripsi telah berhasil diterapkan. Jika belum berubah maka kembali ke langkah-6 dengan melakukan pengecekan error pada script algoritma RSA yang telah dibuat.

Diagram alir penelitian ini dapat dilihat pada Gambar 1.



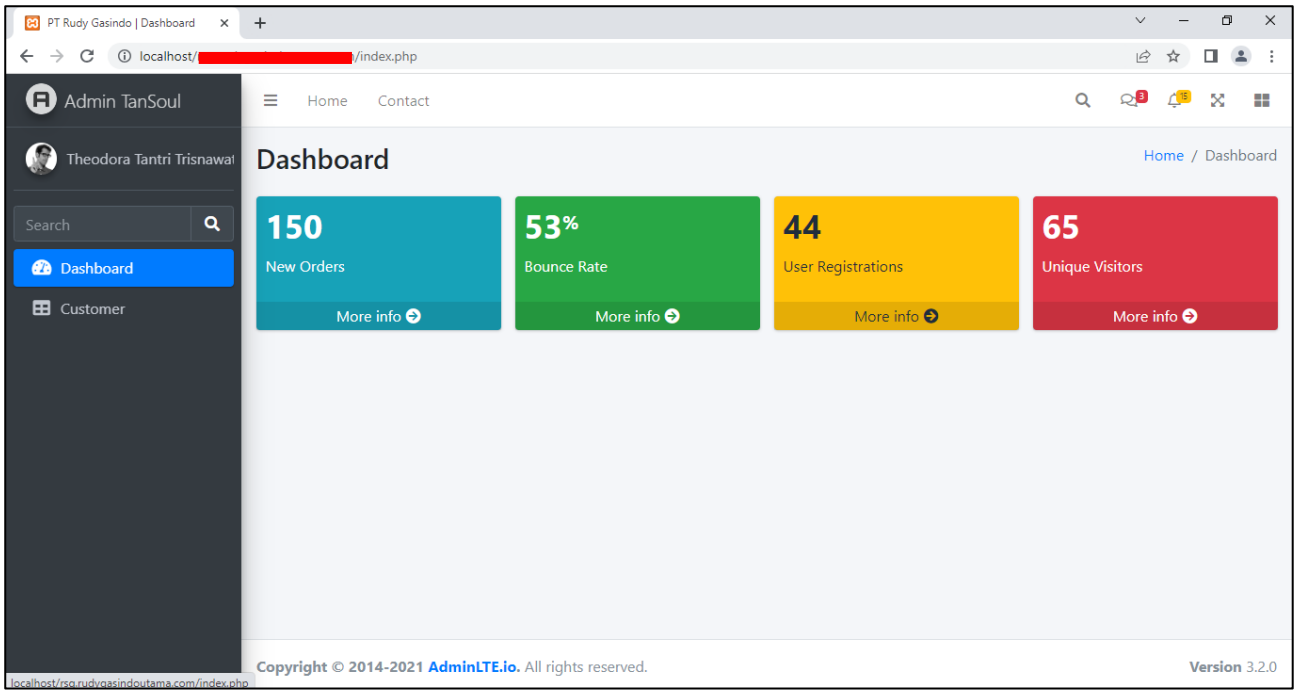
Gambar 1. Diagram Alir Penelitian

3. Hasil dan Pembahasan

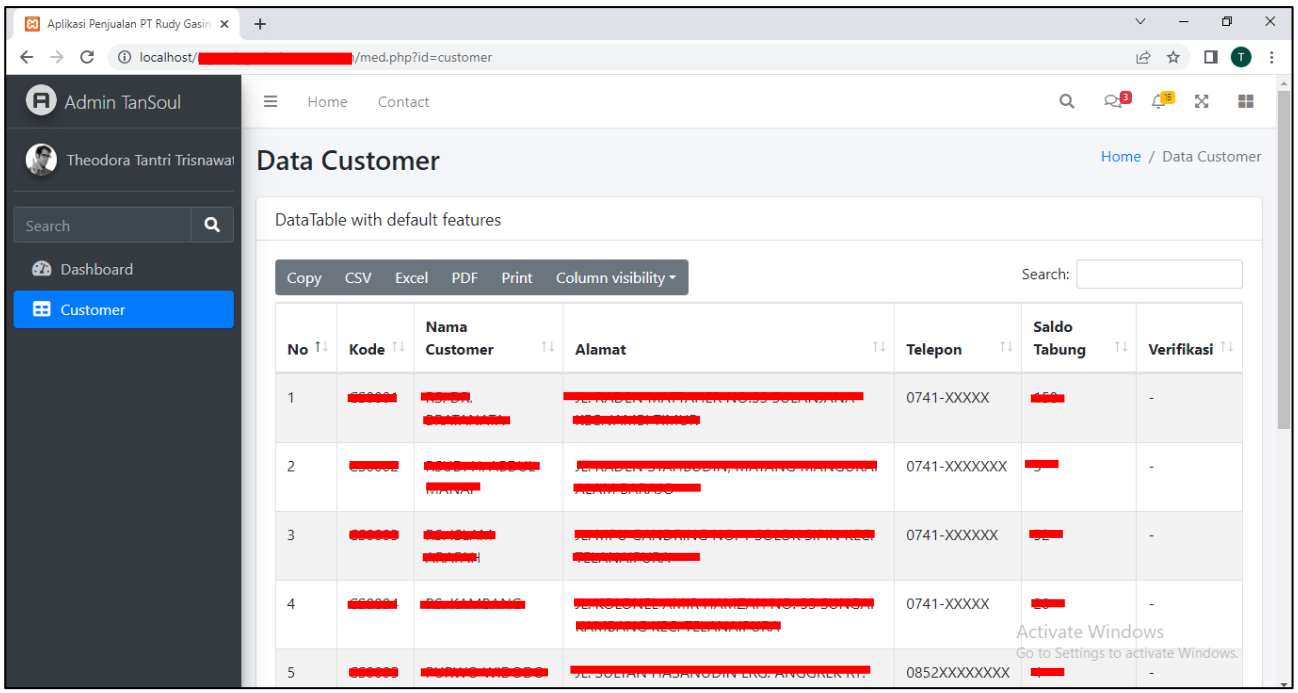
PT. Rezeki Surya Gasindo (RSG) merupakan pabrik sekaligus depot yang menyediakan gas kesehatan dan gas industri. PT. RSG sendiri sudah memiliki perkembangan bisnis yang stabil dan memiliki jumlah *customer* yang cukup banyak, yaitu 1.964 *customer*, dengan total data pribadi milik *customer* yang perlu dilindungi sebanyak 9.820 termasuk diantaranya data berupa kode pelanggan, nama, alamat dan nomor telepon. PT. RSG menyimpan data konsumen mereka dalam sebuah *website*. Dengan tujuan meningkatkan tingkat keamanan data konsumen pada *website* tersebut, maka penulis melakukan enkripsi pada URL *website* PT. Rezeki Surya Gasindo, bagian menu Customer, karena hanya pada menu ini yang berisi data-data pribadi konsumen, yaitu http://rsg.rudysgasindoutama.com/med.php?mod=customer. Namun, *website* yang digunakan pada penelitian ini adalah *website localhost* sebagai *website* uji coba. Maka sebelum melakukan enkripsi terlebih dahulu perlu dibentuk *website localhost* yang digunakan sebagai *website* uji coba.

3.1 Pembangunan Website Localhost

Dalam membangun *website localhost*, diperlukan *template website* untuk membantu dalam membuat *website*. Template *website* yang digunakan adalah *Bootstrap 4*. Setelah ter-*instal* maka selanjutnya tinggal menyesuaikan sesuai dengan yang dibutuhkan. Berikut tampilan menu *Dashboard* dan *Customer* :



Gambar 2. Tampilan Website Menu Dashboard



Gambar 3. Tampilan Website Menu Customer

3.2 Pengaplikasian Algoritma RSA

Berikut adalah langkah-langkah proses enkripsi yang digunakan dalam penelitian ini yang dilakukan secara manual :

1. Pada penelitian ini, dipilih nilai $p = 151$ dan $q = 173$.
2. Nilai $n = p \times q = 151 \times 173 = 26123$.
3. Nilai $\phi(n)$

   $\phi(n) = (p-1)(q-1)$

   $= (151-1)(173-1) = 25800$

4. Pilih kunci publik h dengan h relatif prima terhadap $\phi(n)$, $\left(\gcd\left(h,\phi(n)\right)=1\right)$ dan $1<h<\phi(n)$. Maka dipilih $h=16397$, jelas bahwa 16397 berada pada interval $(1,25800)$. Selanjutnya untuk membuktikan bahwa $h=16397$ relatif prima terhadap $\phi(n)=25800$, maka perlu menemukan $\gcd(16397,25800)$. Dan terbukti bahwa 16397 relatif prima terhadap 25800, karena $\gcd(16397,25800)=1$.

5. Ubah plainteks ke dalam bentuk desimal ASCII berdasarkan Tabel ASCII, kemudian pecah plainteks menjadi beberapa blok $m_1, m_2, \dots, m_i$, dimana setiap m_i mempresentasikan nilai yang berada dalam interval $0 \le m_i \le n-1$.

Plainteks	:	customer										
Pecah Plainteks	:	c	-	u	-	s	-	t	-	o	-	m	-	e	-	r
Bentuk Desimal	:	99		117		115		116		111		109		101		114

6. Setiap blok m_i dienkripsi menjadi blok c_i dengan rumus :

$$c_i = m_i^h \ mod\ n \Leftrightarrow c_i = m_i^{16397} \ mod\ 26123$$

Diperoleh hasil sebagai berikut :

$c = 99^{16397} \ mod\ 26123 = 7258$
$u = 117^{16397} \ mod\ 26123 = 15516$
$s = 115^{16397} \ mod\ 26123 = 18864$
$t = 116^{16397} \ mod\ 26123 = 24408$
$o = 111^{16397} \ mod\ 26123 = 2065$
$m = 109^{16397} \ mod\ 26123 = 15786$
$e = 101^{16397} \ mod\ 26123 = 24942$
$r = 114^{16397} \ mod\ 26123 = 2380$

7. Ubah cipherteks ke bentuk heksadesimal ASCII berdasarkan Tabel ASCII. Karena kode ASCII yang digunakan merupakan ASCII 256-bit, sedangkan cipherteks yang dihasilkan bernilai lebih besar dari 256, sehingga cipherteks yang masih berbentuk numerik harus dimodulokan dengan 256 terlebih dahulu.

$c = 7258 \ mod\ 256 = 90$
$u = 15516 \ mod\ 256 = 156$
$s = 18864 \ mod\ 256 = 176$
$t = 24408 \ mod\ 256 = 88$
$o = 2065 \ mod\ 256 = 17$
$m = 15786 \ mod\ 256 = 170$
$e = 24942 \ mod\ 256 = 110$
$r = 2380 \ mod\ 256 = 76$

Selanjutnya, hasil tersebut diubah kembali ke bentuk heksadesimal ASCII.

Bentuk Desimal	:	90	-	156	-	176	-	88	-	17	-	170	-	110	-	76
Bentuk Heksadesimal	:	5a	-	9c	-	b0	-	58	-	11	-	aa	-	6e	-	4c
Cipherteks	:	5a9cb05811aa6e4c														

Jadi, diperoleh cipherteks = 5a9cb05811aa6e4c, dari plainteks = customer, dengan memilih $p = 151$ dan $q = 173$, dan kunci publik 16397.

3.3 Pengaplikasian Algoritma RSA pada Website

Pada penelitian ini, algoritma RSA dibagi dalam 2 file berbeda, yaitu file *publickey.php* yang digunakan untuk membangkitkan kunci publik dan file *rsa.php* yang digunakan untuk menjalankan perintah enkripsi dengan algoritma RSA. Sumber *coding* yang digunakan adalah

https://github.com/sainihitesh/RSA-Encription-in-php. Berikut merupakan tahap-tahap dalam mengaplikasikan algortima RSA :

1. Menjalankan file *publickey.php*
   a) Input nilai $p = 151$ dan $q = 173$ pada file *publickey.php*.
   b) Jalankan file *publickey.php* pada *browser*, dengan mengetik http://localhost/publickey.php.
   c) Diperoleh *output* sebagai berikut :
      $p = 151$
      $q = 173$
      $n = 26.123$
      $\phi(n) = 25.800$
      $h\ (Kunci\ Publik) = \{7,11,13, \dots ,25799\}$
      Banyaknya Kunci Publik = 6719
      Artinya, ada sebanyak 6719 kunci publik yang terbentuk dari pemilihan bilangan prima $p = 151$ dan $q = 173$, dan dalam penelitian ini telah diambil sebuah kunci publik *h* = 16397.
2. Mengaplikasikan file *rsa.php* pada *website* yang telah dibuat
   a) Berikut merupakan isi dari file *rsa.php* :

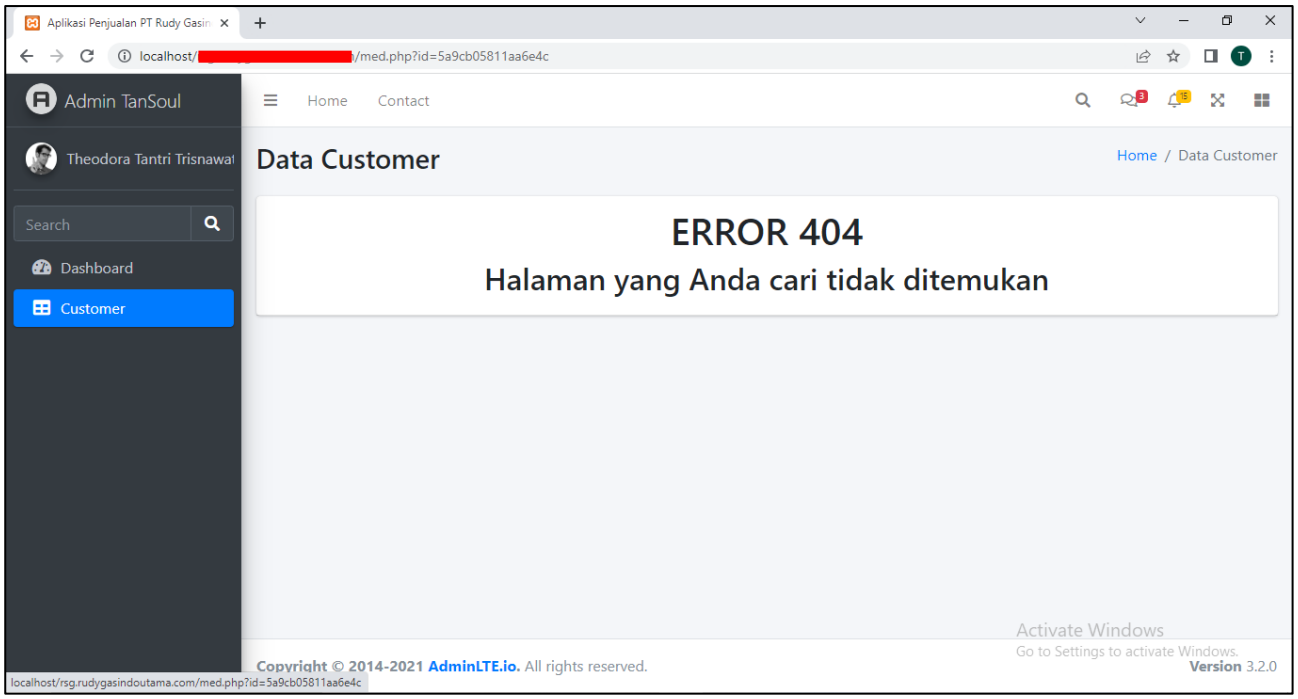
```php
<?php
//fungsi untuk menghitung fast modular exponential
function exp_count($c, $n, $d)
{
            if ($d % 2 == 0) $g = 1; else $g = $c;
            for ($i = 1; $i <= $d / 2; $i++)
            {
                $f = $c * $c % $n;
                $g = $f * $g % $n;
            }
            return $g;
}
//fungsi untuk enkripsi RSA
function RSA_Encrypt($data,$N,$public_key)
{
  $enc_val=array();
  $ascii_val=str_split($data);
  for ($i=0; $i < count($ascii_val); $i++)
  {
    $enc_val[$i]=chr(exp_count(ord($ascii_val[$i]),$N,$public_key));
  }
  $acc=implode('',$enc_val);
  $fin=bin2hex($acc);
  return $fin;
}

?>
```

   b) Berikut *coding* untuk meng-*include* algoritma RSA pada *website* yang telah dibuat :

```php
<?php
   include 'rsa.php';
   $id = "customer";
   $p= 151;
   $q= 173;
   $n= 26123;
   $public_key = 16397;
   $rsa_encrypt= RSA_Encrypt($id,$n,$public_key);
```

```
?>
<a href="med.php?id=<?php echo $rsa_encrypt ?>" class="nav-link active">
```

Selanjutnya, masuk pada tahap pengujian sistem yang telah dibuat. Dalam mengecek apakah Algoritma RSA telah berhasil diterapkan dapat dilakukan dengan mengakses *website localhost* yang telah dibuat. Sebelum diaplikasikan algoritma RSA, tampilan *website* seperti pada Gambar 3. Berikut tampilan *website* setelah diterapkan algoritma RSA :



Gambar 4. Tampilan Website setelah diterapkan Algoritma RSA dengan Kunci 16397

Pada Gambar 4 terlihat bahwa Algoritma RSA sudah berhasil diterapkan pada *website*. Dapat diperhatikan perubahan pada *value* parameter GET *'id'*, sebelumnya adalah customer. Setelah diterapkan algoritma RSA maka berubah menjadi deretan cipherteks 5a9cb05811aa6e4c yang tidak dapat dimengerti oleh pihak ketiga.

Karena terdapat 6719 kunci publik, maka hasil enkripsi atau cipherteks yang dihasilkan juga ada sebanyak 6719 kombinasi. Berikut beberapa contoh hasil cipherteks dari plainteks *customer* dengan mengambil beberapa kunci publik yang berbeda secara random :

Tabel 1. Beberapa Contoh Hasil Cipherteks dari Kunci yang Berbeda

No.	Kunci	Cipherteks
1.	7	b458a97536d2e15b
2.	19	c9aae772737dc8c5
3.	97	acc0af534084c19c
4.	101	bba774765680a8cc
5.	997	2d7e5fd5ddf23dec
6.	1021	a6970cb0d8419723
7.	1997	a3171642ffd9ced9
8.	2023	1395da60a8618209
9.	4003	21d1fe49e59d3fce
10.	5003	2a1fcd9db225b6ff
11.	11131	62a43814f11e3695
12.	11141	57f4f5f5198f7a75
13.	25799	e073a6661c6de6f9

4. Kesimpulan

Dengan menggunakan Kriptografi Algoritma RSA telah dilakukan enkripsi pada *website localhost* yang menjadi *website* tiruan atau uji coba dari *website* asli milik PT. Rezeki Surya Gasindo. Algoritma RSA ini digunakan untuk mengenkripsi parameter GET yang muncul pada URL *bar*, yaitu *customer*. Keberhasilan pengaplikasian Algoritma RSA dapat diamati dengan memperhatikan perubahan *value* parameter GET pada URL *bar*, yaitu *customer* berubah menjadi deretan kode atau cipherteks. Berdasarkan penelitian yang telah dilakukan, dengan mengambil nilai untuk $p = 151$ dan $q = 173$, kemudian menghitung $n = p \times q = 26123$ dan hitung $\phi(n) = (p-1)(q-1) = 25800$. Selanjutnya, bentuk kunci publik *h* dengan syarat *h* harus relatif prima terhadap $\phi(n)$ dan terletak pada interval $1 < h < \phi(n)$. Maka diperoleh sebanyak 6719 kunci publik yang berhasil terbentuk. Dengan mengambil salah satu kunci publik yaitu 16397, diperoleh hasil bahwa *value* parameter GET pada URL *bar* yaitu *customer* telah berubah menjadi 5a9cb05811aa6e4c, artinya proses enkripsi dengan Algoritma RSA telah berhasil diterapkan pada *website localhost* yang telah dibentuk sebelumnya. Karena terdapat 6719 kunci publik, maka hasil enkripsi atau cipherteks yang dihasilkan juga ada sebanyak 6719 kombinasi.

Referensi

[1] M. M. Purba and C. Rahmat, "Perancangan Sistem Informasi Stok Barang Berbasis Web di PT Mahesa Cipta," *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, vol. 8, no. 2, pp. 123–158, 2021, doi: https://doi.org/10.35968/jsi.v8i2.721.

[2] J. Saputra Irsandi, I. Fitri, N. D. Nathasia, and K. Kunci, "Sistem Informasi Pemasaran dengan Penerapan CRM (Customer Relationship Management) Berbasis Website menggunakan Metode Waterfall dan Agile," *Jurnal Teknologi Informasi dan Komunikasi)*, vol. 5, no. 4, p. 2021, 2021, doi: 10.35870/jti.

[3] A. P. Aryani and L. E. Susanti, "Pentingnya Perlindungan Data Pribadi Konsumen dalam Transaksi Online pada Marketplace terhadap Kepuasan Konsumen," *Ahmad Dahlan Legal Perspective*, vol. 2, no. 1, pp. 20–29, 2022, doi: https://doi.org/10.12928/adlp.v2i1.5610.

[4] G. Susilo, "Keamanan basis data pada sistem informasi di era global," *Transformasi*, vol. 12, no. 2, 2017, doi: https://doi.org/10.56357/jt.v12i2.70.

[5] N. I. Putri, R. Komalasari, and Z. Munawar, "Pentingnya keamanan data dalam intelijen bisnis," *J-SIKA| Jurnal Sistem Informasi Karya Anak Bangsa*, vol. 2, no. 02, pp. 41–48, 2020.

[6] R. Andriyanto, K. Khairijal, and D. Satria, "Penerapan Kriptografi AES Class Untuk Pengamanan URL WEBSITE Dari Serangan SQL INJECTION," *Jurnal Unitek*, vol. 13, no. 1, pp. 34–48, 2020, doi: https://doi.org/10.52072/unitek.v13i1.153.

[7] I. P. A. E. Pratama, "Pengujian performansi lima back-end javascript framework menggunakan metode GET dan POST," *Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 6, pp. 1216–1225, 2020, doi: https://doi.org/10.29207/resti.v4i6.2675.

[8] U. D. R. Z. M. Luthfansa and U. D. U. Rosiani, "Pemanfaatan Wireshark untuk Sniffing Komunikasi Data Berprotokol HTTP pada Jaringan Internet," *Journal Information Engineering and Educational Technology) ISSN*, vol. 2549, p. 869X, 2021.

[9] CNN Indonesia, "10 Kasus Kebocoran Data 2022 : Bjorka Dominan, Ramai-Ramai Bantah," 2022. Accessed: Mar. 15, 2023. [Online]. Available: https://www.google.com/amp/s/www.cnnindonesia.com/teknologi/20221230125430-192-894094/10-kasus

[10] A. R. Kelrey and A. Muzaki, "Pengaruh Ethical Hacking Bagi Keamanan Data Perusahaan," *Cyber Security dan Forensik Digital*, vol. 2, no. 2, pp. 77–81, 2019, doi: https://doi.org/10.14421/csecurity.2019.2.2.1625.

[11] D. Ariyus, *Pengantar Ilmu Kriptografi : Teori Analisis dan Implementasi*. Yogyakarta: CV. ANDI Offset, 2008.

[12] M. A. Al-Shabi, "A survey on symmetric and asymmetric cryptography algorithms in information security," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 3, pp. 576–589, 2019, doi: http://dx.doi.org/10.29322/IJSRP.X.X.2018.pXXXX.

[13] M. Y. Simargolang, "Implementasi Kriptografi Rsa Dengan Php," *(JurTI) Jurnal Teknologi Informasi*, vol. 1, no. 1, pp. 1–10, 2017.

[14] R. Munir, *Matematika Diskrit*, 3rd ed. Bandung: INFORMATIKA Bandung, 2010.