APLIKASI PENGAMANAN BASIS DATA DENGAN ALGORITMA RSA DAN WAKE BERBASIS DESKTOP

Peter Jaya[1)], **Safitri Juanita**[2)]
[1]Program studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : peterjaya3@gmail.com[1)], safitri@gmail.com[2)]

Abstrak

UD Bintang Timur adalah perusahaan penyewaan alat berat yang melayani penyewaan mesin giling, mesin excavator dan lainnya. Perusahaan ini memiliki banyak pelanggan yang datanya tersimpan dalam bentuk file basis data bernama data pelanggan. Data pelanggan merupakan data-data yang penting yang berkaitan dengan identitas pelanggan dan bersifat rahasia. Masalahnya adalah data ini tersimpan di basis data yang belum memiliki keamanan tinggi sehingga memiliki kemungkinan untuk diambil atau dirusak oleh pihak yang menginginkan data tersebut, sehingga dibutuhkan perangkat lunak untuk mengamankan data tersebut yaitu suatu aplikasi pengamanan basis data yang berjudul "Aplikasi Pengamanan Basis data dengan Algoritma RSA Dan WAKE Berbasis Desktop". Metode yang digunakan untuk membangun perangkat lunak adalah metode Waterfall. Bahasa pemrograman yang digunakan adalah Java, dengan basis data MySQL. Aplikasi ini mengeimplementasikan metode kriptografi dengan algoritma RSA dan WAKE, algoritma RSA yang digunakan untuk proses enkripsi dan dekripsi data dan algoritma WAKE yang digunakan untuk proses pembuatan public key dan private key pada saat melakukan Generate Key. Kesimpulan dari penelitian ini adalah Dengan adanya aplikasi enkripsi basis data ini, data penting yang dimiliki oleh UD Bintang Timur dapat terjamin keamanan dan kerahasiaannya. Aplikasi ini menggunakan algoritma RSA dan WAKE yang digunakan untuk proses enkripsi dan dekripsi basis data. Diimplementasikan menggunakan bahasa pemrograman Java dengan menggunakan algoritma RSA dan WAKE untuk proses enkripsi dan dekripsi

Kata kunci: *Kriptografi, Algoritma RSA, Algoritma WAKE, Data Pelanggan*

1. PENDAHULUAN

1.1. Latar Belakang

UD Bintang Timur adalah perusahaan penyewaan alat berat yang melayani penyewaan-penyewaan alat berat seperti mesin giling, mesin excavator dan lainnya. UD Bintang Timur juga menggunakan basis data sebagai media penyimpanan data penting yang berkaitan dengan data pelanggan.

Keamanan data pada UD Bintang Timur sangatlah penting seperti data pelanggan. Data pelanggan merupakan sebuah dokumen-dokumen penting berkaitan dengan identitas. Data tersebut merupakan data yang bersifat rahasia dan tidak bisa dirubah oleh pihak yang tidak berhak untuk merubahnya, jika data tersebut diketahui oleh orang yang tidak bertanggung jawab, maka ada pihak yang dirugikan dalam insiden tersebut. Oleh sebab itu, dibutuhkan keamanan data pelanggan pada UD Bintang Timur. Maka dari itu penulis bermaksud untuk membuat aplikasi penyandian Basis Data dengan judul "Aplikasi Pengamanan Basis Data dengan Algoritma RSA Dan Wake Berbasis Desktop".

Metode yang digunakan adalah metode enkripsi asimetris Rivest Shamir Adleman (RSA) dan simetris WAKE (*Word Auto Key Encryption*).

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Metode WAKE (*Word Auto Key Encryption*) proses penyelesaian metode ini cukup rumit dan sulit untuk dikerjakan secara manual karena algoritmanya yang cukup panjang dan kompleks.

1.2. Batasan Masalah

Dari beberapa uraian latar belakang diatas, maka penulis merumuskan masalah yang akan dibahas adalah sebagai berikut :

a. Bagaimana mengimplementasikan Algoritma RSA dan WAKE untuk mengamankan data yang disimpan dalam bentuk file Basis Data agar terjaga kerahasiaannya?

1.3. Tujuan

Adapun tujuan yang ingin dicapai dari penulisan tugas akhir ini adalah:

a. Mengembangkan suatu aplikasi pengamanan basis data menggunakan algoritma kriptografi RSA dan WAKE untuk mengamankan informasi atau data yang sifatnya rahasia.
b. Mengamankan data yang disimpan dalam bentuk file basis data agar tidak dapat diketahui oleh orang yang tidak bertanggung jawab

melalui file basis data dalam satu aplikasi dengan algoritma RSA dan WAKE.

c. Menghasilkan aplikasi enkripsi basis data yang diharapkan mudah dimengerti dan digunakan oleh pengguna.

2. METODOLOGI PENELITIAN

Dalam penulisan Tugas Akhir ini, penulis menggunakan metode *Waterfall*. Menurut Presmann (2001) metode *waterfall* meliputi tahapan perencanaan, analisis, desain, implementasi, pengujian dan pemeliharaan. Berikut ini adalah rincian tahapan dalam pembuatan aplikasi pengamanan basis data yaitu:

a. Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data dengan cara :

1) Observasi

Observasi dilakukan pada bulan Oktober dan dilakukan di UD. Bintang Timur.

2) Wawancara

Melakukan serangkaian wawancara kepada pihak terkait yang menunjang pembuatan aplikasi yang dibuat, penulis melakukan wawancara tatap muka dengan pemilik UD. Bintang Timur.

3) Analisis data

Menganalisis Algoritma kriptografi yang digunakan yaitu algoritma RSA dan WAKE.

b. Menganalisa kebutuhan aplikasi

Setelah memperoleh kebutuhan aplikasi kemudian dipelajari dan dianalisa menggunakan *flowchart*.

c. Desain atau Perancangan Program

Desain rancangan layar menggunakan Edraw Max 7.9. Merancang program yang dibangun menggunakan *Software* Netbeans 8.0.

d. Pengkodean

Pengkodean dilakukan untuk memudahkan dalam mengimplementasikan rancangan aplikasi ke dalam algoritma RSA dan WAKE dengan menggunakan bahasa pemrograman Java menggunakan Netbeans 8.0.

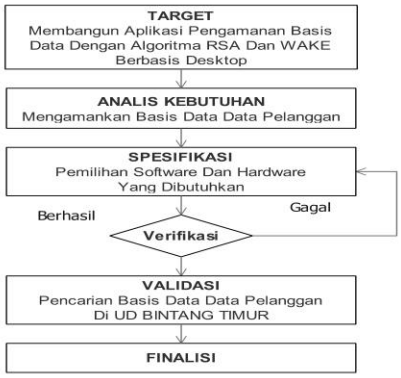
e. Implementasi

Rancangan aplikasi yang sudah dibuat kemudian diimplementasikan berdasarkan analisa masalah

f. Pengujian

Pada tahap ini unit program diintegrasikan dan diuji sistem yang lengkap untuk meyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah diuji coba, sistem disampaikan ke UD. Bintang Timur.

Berikut adalah design atau alur penelitian :



Gambar 1 : Alur Penelitian

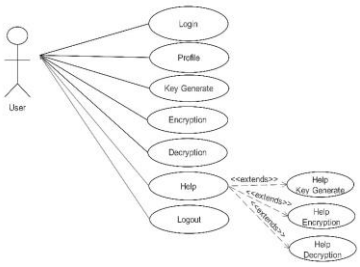
3. HASIL DAN PEMBAHASAN

Program yang akan dibuat terdiri dari beberapa *Form*, yaitu terdiri dari *Form* Menu Utama, *key Generate*, Enkripsi, Dekripsi, Bantuan, dan Profil. Untuk dapat melakukan enkripsi *database* pengguna dapat menggunakan Menu Enkripsi dengan memilih *database* yang akan dienkripsi dan memerlukan *Public Key* untuk dapat mengenkripsi *database*. Selanjutnya akan tampil *output* berupa informasi hasil enkripsi *database* tersebut.

Sedangkan untuk mengembalikan *database* yang sudah dienkripsi menjadi *database* asli, pengguna juga dapat memilih menu dekripsi, namun pengguna harus memiliki *Private Key* untuk mengembalikan *database* tersebut. Serta ada menu bantuan untuk membantu *user* dalam menggunakan program tersebut. Untuk mendapatkan *Public* dan *Private Key* pengguna dapat menggunakan *Menu Key Generate*. Secara umum, rancangan program yang akan dibuat dapat dilihat pada gambar 1 berikut ini :

Gambar 2 : Arsitektur Kerja Aplikasi

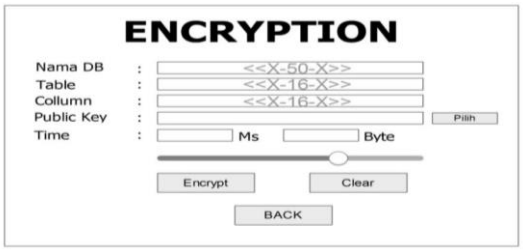
Berikut adalah *use case* diagram dari aplikasi yang akan dibangun :



Gambar 3 *: Use Case Diagram*

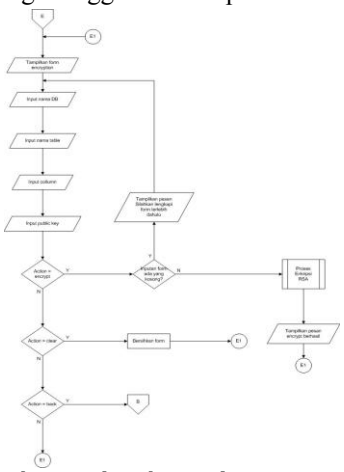
3.1. Proses Enkripsi

Pada aplikasi ini, user harus menginput Nama DB, *table*, *column*, *public key* terlebih dahulu dan selanjutnya mengklik tombol *encrypt* untuk mengubah data asli menjadi data rahasia. Berikut gambar rancangan layar halaman *encryption*:



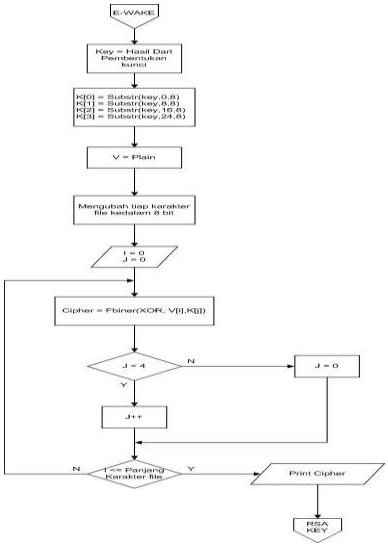
Gambar 4 : Rancangan Layar Halaman *Encryption*

Flowchart proses enkripsi ini menjelaskan tentang alur sub proses enkripsi data yaitu mengubah data asli menjadi sebuah data rahasia. Berikut *flowchart* yang menggambarkan proses *encryption* :



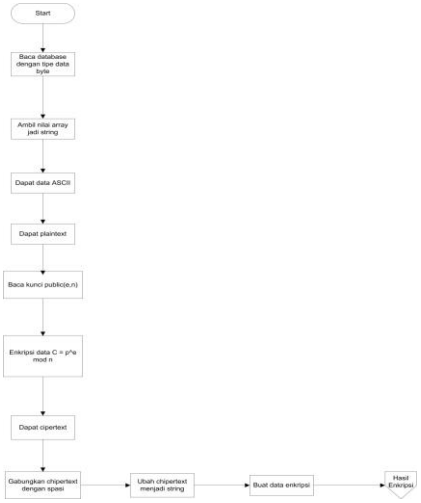
Gambar 5 : Flowchart Halaman *Encryption*

Flowchart proses enkripsi ini menjelaskan tentang alur sub proses enkripsi data yang akan diubah menjadi *ciphertext* dengan menggunakan *key* yang di input pada *public key*. Berikut *flowchart* yang menggambarkan proses enkripsi WAKE:



Gambar 6 : Flowchart Halaman Enkripsi WAKE

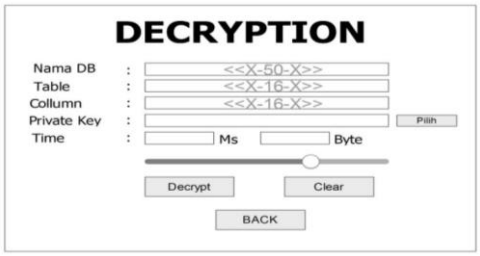
Flowchart proses enkripsi ini menjelaskan tentang alur sub proses enkripsi data yang akan diubah menjadi *ciphertext* dengan menggunakan *key* yang di input pada *public key*. Berikut *flowchart* yang menggambarkan proses enkripsi RSA:



Gambar 7 : Flowchart Halaman Enkripsi RSA

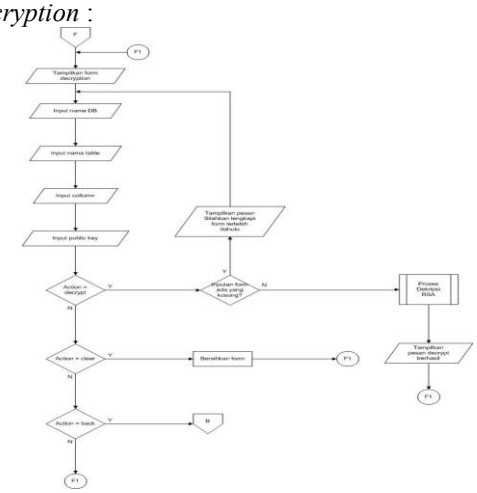
3.2. Proses Dekripsi

Pada aplikasi ini, user harus menginput Nama DB, *table*, *column*, *private key* terlebih dahulu dan selanjutnya mengklik tombol d*ecrypt* untuk mengubah data rahasia menjadi data asli tanpa mengubah isi dari data tersbut. Berikut gambar rancangan layer halaman *decryption*:



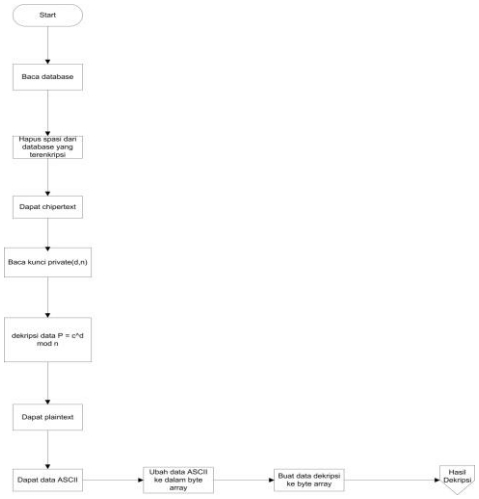
Gambar 8 : Rancangan Layar Halaman *Decryption*

Flowchart proses dekripsi ini menjelaskan tentang alur sub proses dekripsi data yaitu mengubah data rahasia menjadi data asli tanpa mngubah isi dari data. Berikut *flowchart* yang menggambarkan proses *decryption* :



Gambar 9 : Flowchart Halaman *Decryption*

Flowchart proses dekripsi ini menjelaskan tentang alur sub proses dekripsi data yang akan diubah menjadi *plaintext* dengan menggunakan *key* yang di input pada *private key*. Berikut *flowchart* yang menggambarkan proses dekripsi RSA:



Gambar 10 : Flowchart Halaman Dekripsi RSA

4. HASIL DAN PEMBAHASAN

Setelah melalui tahap perancangan, pembuatan hingga uji coba program, maka dapat diperoleh hasil dari uji coba program tersebut.

4.1. Halaman Login

Untuk menggunakan aplikasi ini, *user* dapat login terlebih dahulu ke aplikasi dengan menginput *username* dan *password* pada layar, lalu mengklik tombol Login untuk masuk ke halaman menu utama. Berikut Tampilan login:

Gambar 11 : Tampilan Layar Halaman Login

4.2. Menu Utama

Menu utama aplikasi terdiri menu *profile* untuk mengetahui tentang UD Bintang Timur, menu *generate key* untuk proses pembuatan *public* dan *private key* yang akan digunakan pada saat ekripsi dan dekripsi, menu *encryption* untuk melakukan proses enkripsi data, menu *decryption* untuk melakukan proses dekripsi data, menu *help* untuk melihat cara penggunaan aplikasi dan didalam menu *help* terdapat 3 menu lagi yaitu : menu *help key*, *help encryption*, dan *help decryption* dan *button logout* untuk keluar dari aplikasi.

Gambar 12 : Tampilan Layar Halaman Menu Utama

4.3. Menu profile

Fungsi dari menu *profil* ini adalah untuk memudahkan *user* mengetahui tentang *profile* UD Bintang Timur. Berikut adalah gambar tampilan layar menu *profile*.

Gambar 13 : Tampilan Layar Halaman Menu *Profile*

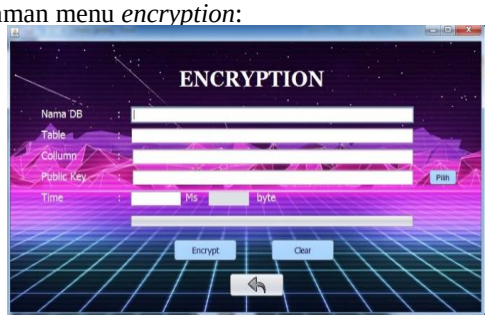
4.4. Menu Key Generate

Fungsi dari menu *generate key* adalah pengguna dapat mengisi *key name* untuk mendapatkan *public* dan *private key*. Berikut adalah gambar tampilan layar menu *key generate:*

Gambar 14 : Tampilan Layar Halaman Generate Key

4.5. Menu *Encryption*

Tampilan menu *encryption* merupakan tampilan layar yang digunakan untuk mengubah data asli menjadi data rahasia. Berikut ini tampilan layar halaman menu *encryption*:



Gambar 15 : Tampilan Layar Halaman Encryption

4.6. Menu *Decryption*

Tampilan menu *decryption* merupakan tampilan layar yang digunakan untuk mengubah data rahasia menjadi data asli tanpa mengubah isi data tersebut. Berikut ini tampilan layar halaman menu *decryption*:

Gambar 13 : Tampilan Layar Halaman Menu

4.7 Menu *Help*

Tampilan menu help merupakan tampilan layar yang memudahkan pengguna mengetahui cara penggunaan aplikasi. Di dalam menu help terdapat 3 menu yaitu menu *help key*, *encryption*, dan *decryption*. Untuk lebih jelasnya berikut gambar layar menu *help*:

Gambar 17 : Tampilan Layar Halaman Menu *Help*

4.8. Proses *Key Generate*

Fungsi dari menu *generate key* adalah sebelum melakukan proses enkripsi dan dekripsi, maka terlebih dahulu untuk membuat *key name*, dan pilih *directory key* untuk menyimpan *public* dan *private key* lalu klik *save*, lalu klik *generate key*. Untuk lebih jelasnya berikut adalah tampilan layar proses pembuatan *Generate Key*:

Gambar 18: Tampilan Layar Proses *Generate Key*

Klik pilih untuk menyimpan *directory* key. Klik save untuk menyimpan *private* dan *public key* yang nanti akan digunakan pada saat enkripsi dan dekripsi.

Gambar 19 : Tampilan Layar *Directory Key*

4.9. Proses Enkripsi

Pada proses enkripsi pengguna meng-*input* nama db, *table*, dan *collumn* yang ingin di-enkripsi, pilih *public key*, lalu klik *encrypt* maka akan muncul time dan pesan berhasil di-enkrip. Untuk lebih jelasnya berikut adalah gambar proses *encryption* :

Gambar 20 : Tampilan Layar Proses Enkripsi

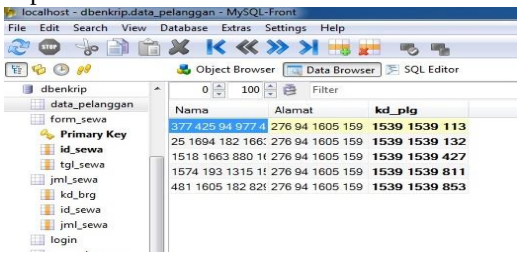
Klik pilih untuk mencari *directory public key* yang telah disimpan pada saat proses pembuatan *key generate*. Dapat dilihat pada gambar berikut :

Gambar 21 : Tampilan Layar *Directory Private Key*

Setelah memilih *public key*, maka selanjutnya klik tombol *Encrypt*. Setelah enkripsi berhasil maka akan tampil time dan pesan berhasil di enkripsi.

Gambar 22 : Tampilan Setelah Mengklik Tombol *Encryption*

Hasil basis data yang telah di-enkrip. Berikut adalah gambar hasil dari basis data yang telah ter-enkripsi:



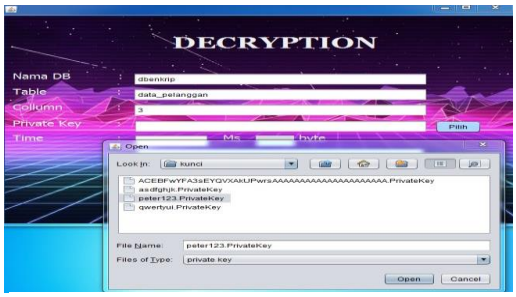
Gambar 23 : Tampilan Hasil Enkripsi

4.10. Proses Dekripsi

Pada proses dekripsi ini pengguna dapat mengubah data yang telah ter-enkripsi menjadi data asli tanpa mengubah isi dari data tersebut. Kemudian pengguna meng-*input* nama db, *table*, dan *collumn* yang ingin di-enkripsi, pilih *private key*, lalu klik *decrypt* maka akan muncul time dan pesan berhasil di-dekrip. Untuk lebih jelasnya berikut adalah gambar layar menu *decryption* :

Gambar 24 : Tampilan Layar Halaman Proses *Decryption*

Klik pilih untuk mencari directory *private key* yang telah disimpan pada saat pembuatan *key generate*.

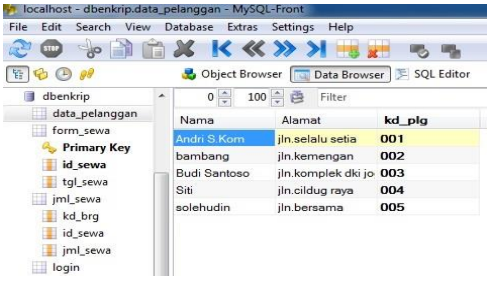


Gambar 25 : Tampilan Layar *Directory Private Key*

Setelah memilih *private key*, maka selanjutnya klik tombol *decrypt*. Setelah dekripsi berhasil maka akan tampil time dan pesan berhasil di dekripsi.

Gambar 26 : Tampilan Layar Saat Klik Tombol *Decrypt*

Hasil basis data yang telah di dekripsi. Berikut adalah gambar hasil dari basis data yang telah di dekripsi:



Gambar 27 : Tampilan Hasil Dekripsi

5. KESIMPULAN

Berdasarkan proses perancangan, pembuatan, dan pengujian aplikasi enkripsi basis data ini, data penting yang dimiliki oleh UD Bintang Timur dapat terjamin keamanan dan kerahasiaannya. Aplikasi ini menggunakan algoritma RSA dan WAKE yang digunakan untuk proses enkripsi dan dekripsi basis data. Diimplementasikan menggunakan bahasa pemrograman *Java*.

6. SARAN

Untuk pengembangan lebih lanjut agar aplikasi ini menjadi lebih baik lagi, adapun saran yang diberikan antara lain:

a. Diharapkan aplikasi ini dapat mengenkripsi basis data per column sehingga aplikasi ini dapat berjalan secara maksimal.

7. DAFTAR PUSTAKA

[1] Busran, at al., 2012. Analisis Komputasi Enkripsi Dan Dekripsi Data Gambar, Teks Dan Audio Dengan Menggunakan Algoritma RC4 Berbasis Visual Basic 6.0, *Jurnal Teknologi Informasi & Pendidikan*, 5(1), ISSN : 2086-4981, pp. 32–45. Available at:http://jurnal-tip.net/jurnalresource/file/3Vol5No1Maret2012-Busran-Putri Mandarani.pdf

[2] Kromodimoeljo, S., 2009. Teori & Aplikasi Kriptografi. Jakarta: SPK IT Consuling.

[3] Munir, I. R., 2010. Algoritma RSA dan ElGamal, *Kriptografi*. Bandung: Informatika.

[4] Purwadi, at al., 2014. Aplikasi Kriptografi Asimetris Dengan Metode Diffie-Hellman Dan Algoritma ElGamal Untuk Keamanan Teks, *Jurnal Ilmiah SAINTIKOM (Sains dan Komputer)*, 13(3), ISSN : 1978-6603, pp. 183–196. Available at: https://lppm.trigunadharma.ac.id/public/fileJurnal/hpPmJurnalPurwadi 2014.pdf.

[5] Simargolang, at al., 2017. Implementasi Kriptografi RSA Dengan PHP, *Jurnal Teknologi Informasi(JurTI)*, 1(1), P-ISSN : 2580-7927, pp. 1–11. Available at: http://jurnal.una.ac.id/index.php/jurti/article/viewFile/1/1.

[6] Wahyadyatmika, at al., 2014. Implementasi Algoritma Kriptografi RSA Pada Surat Elektronik (*E-Mail*), *TRANSIENT*, 3(4), ISSN : 2302-9927. Available at: http://download.portalgaruda.org/article.php?article=270625&val=4717&title=Implementasi Algoritma Kriptografi Rsa Pada Surat Elektronik (*E-Mail)*.