# PENGGUNAAN ALGORITMA RSA UNTUK KEAMANAN TRANSAKSI ONLINE BERBASIS APLIKASI MOBILE

Terry Firasyan
Jurusan Teknik Informatika, Idris Winarno, Yuliana Setiowati
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember Surabaya
Kampus PENS-ITS Keputih Sukolilo Surabaya 60111
Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114
Email: teriyaki15@gmail.com

**Abstrak**

*Perkembangan teknologi mendukung orang beralih ke media online seperti halnya toko online yang dulu membangun toko dengan modal besar. Dan sekarang dengan media toko online memerlukan biaya yang murah. Dengan perkembangan itu banyak orang menggunakannya, di sisi lain semakin banyak juga tindak kejahatan yang bisa menyerang kita. Maka pada makalah ini dibangun aplikasi berbasis android untuk mengamankan transaksi online dengan menggunakan algoritma RSA. Algoritma RSA adalah proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo.*

**Kata kunci** : Enkripsi, kriptografi, RSA, asimetris/public-key, android

## 1. Latar Belakang

Transaksi Online (Transaksi jual beli barang atau jasa secara online) sudah lazim dilakukan, banyaknya toko online yang menawarkan berbagai barang kebutuhan manusia telah mengembangkan metode baru sebagai alat tukar transaksi disamping metode yang sudah umum digunakan. Namun keamanan dalam menggunakan metode ini telah menjadi perhatian utama mengingat jaringan online sangat rawan tindak kejahatan.

Untuk jual-beli online biasanya orang cenderung tidak tahu apakah data input pembelian dari toko online sudah diterima dengan aman atau ada pihak lain yang tahu. Jika ada pihak lain tahu yang bisa menyebabkan data tersebut dimanipulasi bisa berakibat fatal. Karena data inputan bersifat plain teks maka siapapun bisa membacanya. Pada proyek akhir ini mencoba memberikan alternatif untuk menyelesaikan masalah ini.

## 2. Dasar Teori

### a. Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak
Fitur yang tersedia di Android adalah:

- Kerangka aplikasi: itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat mobile.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- SQLite: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3,AAC, AMR, JPG, PNG, GIF)
- GSM, Bluetooth, EDGE, 3G, dan WiFi (hardware dependent)
- Kamera, Global Positioning System (GPS), kompas, dan *accelerometer* (tergantung hardware)

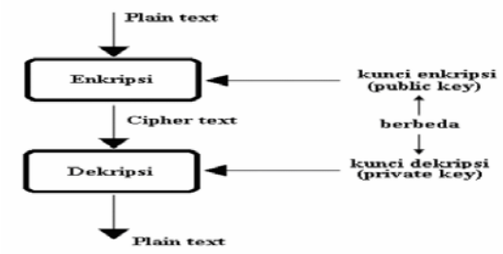
### b. Socket Programing

Setiap aplikasi di jaringan, menggunakan transaksi didasarkan pada konsep *client*-server. Sebuah server dan sebuah atau beberapa *client* yang meminta/request pelayanan ke server. Fungsi server sebagai pengatur resource yang ada, yang menyediakan pelayanan dengan memanfaatkan resource yang untuk kebutuhan *client*.

### c. RSA

Kriptografi berasal dari kata *crypto* yang berarti rahasia dan *graf* yang berarti tulisan, jadi kriptografi adalah suatu cara membuat tulisan rahasia yang aman dari campur tangan pihak ketiga atau pihak yang tidak diharapkan. Algoritma atau hitungan kriptografi dapat disebut juga *chipper* yaitu hitungan matematik yang digunakan untuk membuat suatu enkripsi maupun dekripsinya. Sebuah data yang tidak dienkripsi disebut *plaintext*. Perlu diketahui bahwa hitungan enkripsi ini tidak sama dan diusahakan untuk tidak sama. Hal ini bertujuan untuk mencegah supaya data kita tidak bisa dibaca oleh orang yang bukan seharusnya.

Algoritma kriptografi RSA merupakan algoritma yang termasuk dalam kategori algoritma asimetri (juga disebut sebagai algoritma kunci publik), didesain sedemikian sehingga kunci yang digunakan untuk enkripsi berbeda dari kunci yang digunakan untuk dekripsi. Algoritma disebut kunci publik karena kunci enkripsi dapat dibuat publik yang berarti semua orang boleh mengetahuinya.



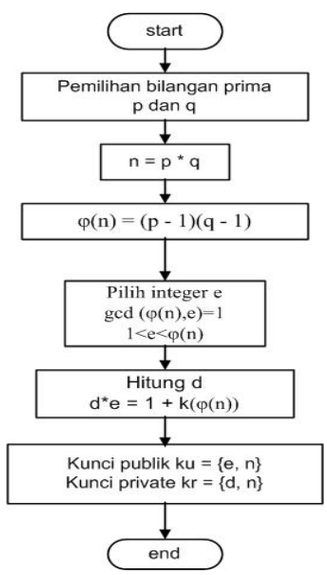

**Gambar 2. 1 Enkripsi RSA**

**Pembuatan Kunci**

Pada bagian ini, terdapat lima tahapan. Proses ini dilakukan oleh pihak server.

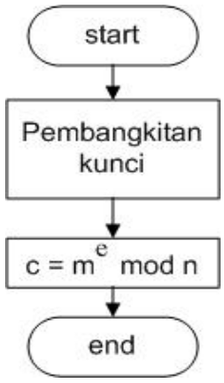
1) Pilih bilangan prima sembarang *p* dan *q*. Kedua nilai ini harus dirahasiakan.Misal bil. prima p = 7 dan q = 11,
2) Hitung $n = p.q$, Besaran n ini tidak perlu dirahasiakan .n = 7.11 =77
3) Hitung $\varphi(n) = (p - 1)(q - 1)$ . $\varphi(n) = (7 - 1)(11 - 1) = 6.10 = 60$
4) Pilih sebarang bilangan e, $1 < e < \phi(n)$, $\Phi(n)=\{1,2,3,4,6,8,..,76\}=\{x|gcd(x, n)=1\}$ misalnya e=17
5) dengan $gcd(\phi(n), e) = 1$ . Pilih e dalam {x|gcd(x, 60)=1}
6) Hitung invers dari e, yaitu $d*e = 1+ k(\phi(n))$. d .e = 1mod 60, d =53 53 . 17 mod 60 = 901 mod 60 = 1 mod 60
7) Kunci publik: (n, e) dan kunci rahasia: (n, d).

Gambar 2.2. di bawah ini adalah flowchart untuk membuat kunci (membangkitkan kunci) untuk enkripsi dan dekripsi.




**Gambar 2. 2 Setup Key**

**Proses Enkripsi**




**Gambar 2. 3 Proses Enkripsi**

Berikut ini adalah proses enkripsi RSA. Dilakukan oleh pihak pengirim. Seluruh perhitungan pemangkatan bilangan modulo dilakukan menggunakan metode *fast exponentiation*.

1) Ambil kunci publik (n,e).
2) Pilih plainteks m, dengan $0 \leq m \leq n - 1$.
3) Hitung $c = m^e \mod n$ .
4) Diperoleh cipherteks c, dan kirimkan.

M = “PESAN”, m = 16 5 19 1 14

•Enkripsi: c = me mod n

- c1 = 1617 mod 77 = 25
- c2 = 517 mod 77 = 3
- c3 = 1917 mod 77 = 24
- c4 = 117 mod 77 = 1
- c5 = 1417 mod 77 = 42

c = 25 03 24 01 42, C = “YCXAp”

**Proses Dekripsi**

Berikut ini adalah proses dekripsi RSA. Dilakukan oleh pihak penerima cipherteks.

1. Ambil kunci publik (n,e) dan kunci rahasia (n,d).
2. Hitung $m = c^d \mod n$.
3. Diperoleh plainteks m.

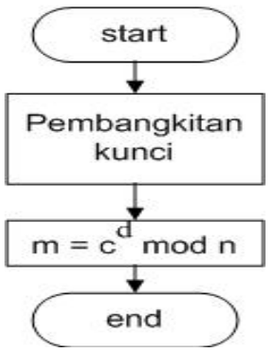
C = “YCXAp”, c = 25 03 24 01 42

•Dekripsi: m = cd mod n

- m1 = 2553 mod 77 = 16
- m2 = 353 mod 77 = 5
- m3 = 2453 mod 77 = 19
- m4 = 153 mod 77 = 1
- m5 = 4253 mod 77 = 14

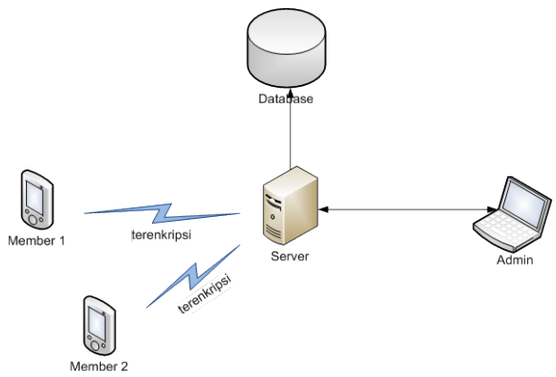
m = 16 5 19 1 14 , M = “PESAN”

Sedangkan penjelasan proses dekripsi RSA secara singkat dapat dilihat pada Gambar 2.8 :




**Gambar 2. 4 Proses dekripsi**

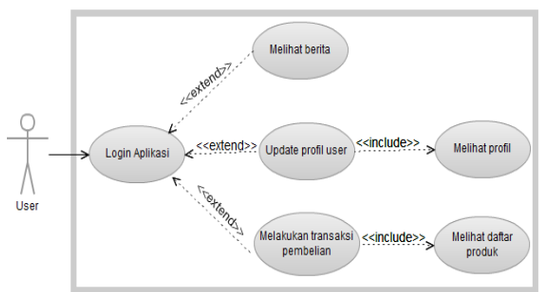
### 3. Perancangan dan Pembuatan Sistem




**Gambar 3. 1 Blok diagram sistem**

Perangkat lunak yang dibuat pada server menggunakan platform berbasis java. Sedangkan klien menggunakan mobile aplikasi android dengan data terenkripsi seperti terlihat pada gambar 3.1. Server akan mendekripsi pesan dari klien dan akan disimpan di database. Admin akan memriksa pesan dari klien dan transaksi akan

**Use case Diagram**

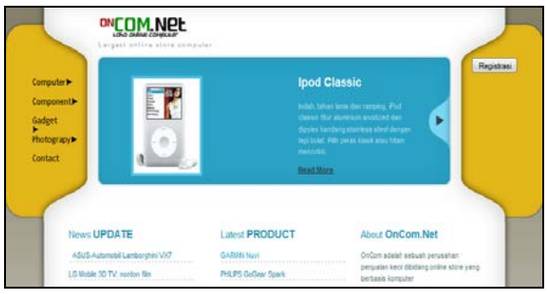



**Gambar 3. 2 Use case digram**

Gambar diatas adalah *use case diagram* yaitu menggambarkan fungsionalitas dari sebuah sistem. Gambar yang dtunjunkan pada 3. 4 adalah kebutuhan sistem dari sudut pandang user . Member diberikan hak akses tiga pilihan yaitu : melihat berita, daftar produk , melihat profil.

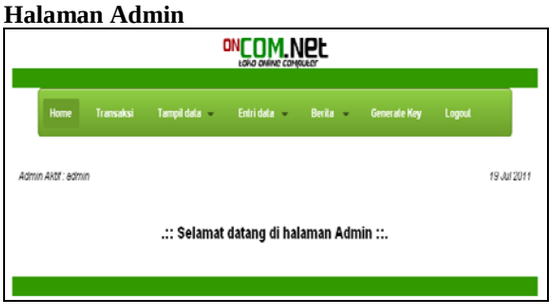
**Antar muka website**

Web yang telah dibuat dihosting secara localhost dengan domain http://localhost/toko. Ketika pertama kali web aplikasi dibuka maka akan diarahkan secara langsung ke http://localhost/toko/index.php rancangan awal halaman web terdapat pada gambar 3.4.




**Gambar 3. 3 Rancangan Halaman Awal**

**Halaman Admin**

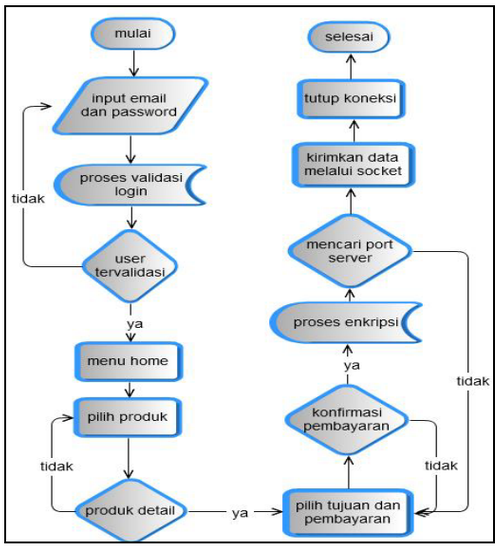



**Gambar 3. 4 Halaman Admin**

Pada halaman admin akan ditampilkan beberapa menu antara lain transaksi,tampil data,entri data,berita,generate key dan logout.

**Flowchart klien**

Untuk merancang mobile android dibutuhkan langkah-langkah atau *flowcart* dimana terlihat pada gambar 3.13 . Untuk masuk kedalam aplikasi klien diwajibkan login terlebih dahulu dengan menggunakan email dan password yang terregistrasi. Selanjutnya akan ditampilkan menu *home* klien harus memilih *tab* produk untuk memulai transaksi. Produk di susun berdasarkan kategori, klien memilih produk kemudian halaman produk detail akan ditampilkan. Pada form selanjutnya klien akan memilih tujuan pengiriman dan pembayaran sebelum adanya konfirmasi untuk pembayaran ke pihak bank yang bersangkutan. Dalam kasus ini diwajibkan membayar sebelum pindah halaman. Proses enkripsi dilakukan kemudian mencari *port* yang telah ditentukan melalui socket. Koneksi akan ditutup ketika proses pengiriman berhasil dilakukan.




**Gambar 3. 5 Flowcart klien**

Dalam aplikasi Android terdapat beberapa tampilan antar muka yang digunakan untuk proses transaksi, adalah sebagai berikut :

1. Login : Aplikasi pertama akan menampilan halaman ini. User diharuskan untuk memasukkan alamat email sebagai username dan password. Rancangan halaman login bisa dilihat pada gambar 3. 6


**Gambar 3. 6 Halaman Login Android**

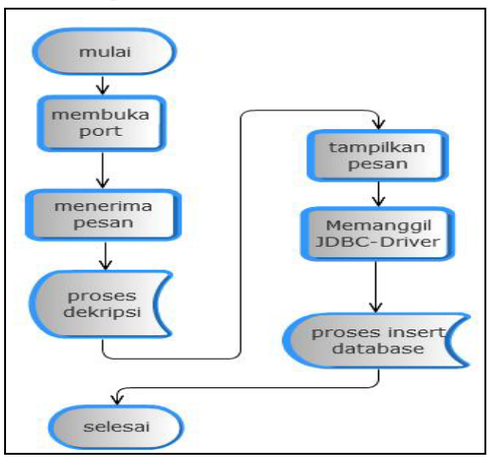
2. Home : Pada halaman ini ada tiga menu yaitu News, Produk, Profile.
3. Produk : Menu ini ditunjukan untuk melakukan transaksi dimana terdapat pilhan-pilihan produk. Pembelian dapat dilakukan dengan memilih salah satu produk. Selanjutnya aplikasi akan menampilkan halaman detail produk.. Rincian transaksi dan pilihan pembayaran akan ditampilkan setelah user menekan tombol *add to cart* sesuai dengan rancangan pada gambar 3.8. Terdapat pilihan metode pembayaran dan tujuan pengiriman pada rincian tersebut.


**Gambar 3. 7 Menu Produk dan detailnya**

Pada halaman ini yang terlihat pada gambar 3.9 klien diwajibkan mengisi field yang tersedia. Proses transaksi bisa dilanjutkan dengan menekan tombol next step. Pada langkah ini sistem akan menampilkan proses persetujan pembayaran. Proses transaksi yang terakhir menampilkan data-data transaksi yang dilakukan sebelumnya yang telah di inputkan oleh member.


**Gambar 3. 8 Menu Pilihan Pengiriman dan Pembayaran**

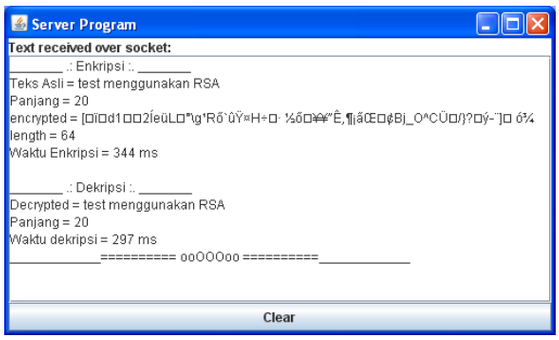
**Perancangan Antar Muka Server**




**Gambar 3. 9 *Flowcart server***

Pada langkah ini akan dijelaskan bagaimana komunikasi server dengan klien. Port akan dibuka ketika server dijalankan. Pesan akan diterima dalam bentuk enkripsi kemudian dilakukan proses dekripsi. Server akan menampilkan hasil dekripsi, panjang karakter, waktu dekripsi dan enkripsi yang terlihat pada gambar 3.12. Setelah server berhasil mendekripsi dengan benar proses terakhir akan

dilanjutkan dengan memanggil JDBC-Driver dan akan disimpan ke database MySQL.



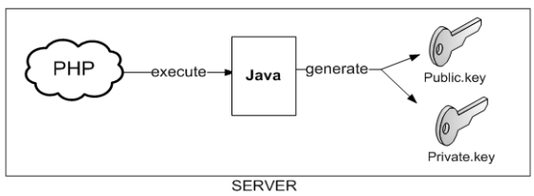

**Gambar 3. 10 Halaman server**

Proses enkripsi dikirim ke server dan di dekripsi seperti pada gambar 3.13. Halaman ini menampilkan beberapa inputan dari klien yaitu :

- Enkripsi: Teks Asli ,Panjang, Encrypted, Length, Waktu enkripsi
- Dekripisi: Decrypted, Panjang, Waktu dekripsi,
- *Clear* : Merupakan tombol untuk menghapus teks enkripsi dan dekripsi ketika form tersebut penuh.
- *Scroll* : Merupakan tombol untuk mengarahkan vertikal, horizontal, atas dan bawah.

**Perancangan Aplikasi Enkripsi dan Dekripsi RSA pada Mobile Android**

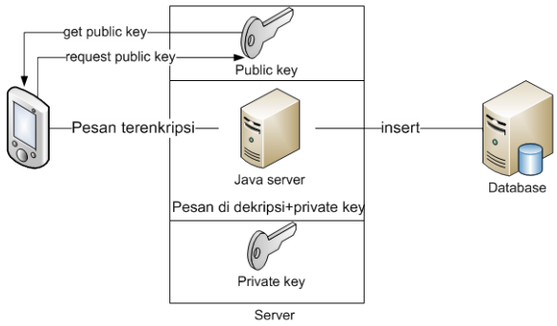
Proses yang ditunjukkan pada gambar 3.14 dibawah ini adalah gambaran proses pembangkitan kunci. Dalam proses ini akan menghasilkan kunci menggunakan berekstensi file `.key` yang pertama `public.key` dan `private.key`. Keduanya disimpan pada server karena lebih memudahkan pihak server jika akan membangkitkan kunci baru.




**Gambar 3. 11 Proses pemgbangkitan kunci**

Diagram proses enkripsi dan dekripsi data dilihat pada gambar 3.15. Klien akan meminta public key dari server yang digunakan untuk proses yang nantinya akan digunakan untuk enkripsi data. Proses enkripsi dilakukan pada klien. Proses ini menghasilkan data terenkripsi atau biasa disebut *chipertext*. Data inilah yang akan dikirimkan ke server yang akan di dekripsi dan proses pengirimannya melalui *socket*.

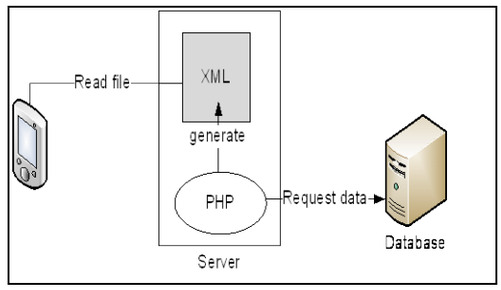
Tahapan selanjutnya pada server adalah dekripsi *chipertext.* Proses ini memerlukan private key yang akan menghasilkan data terdekripsi. Data inilah yang akan dimasukkan ke server database.




**Gambar 3. 12 Alur proses enkripsi dan dekripsi**

**Perancangan Pertukaran Data Pada Aplikasi**

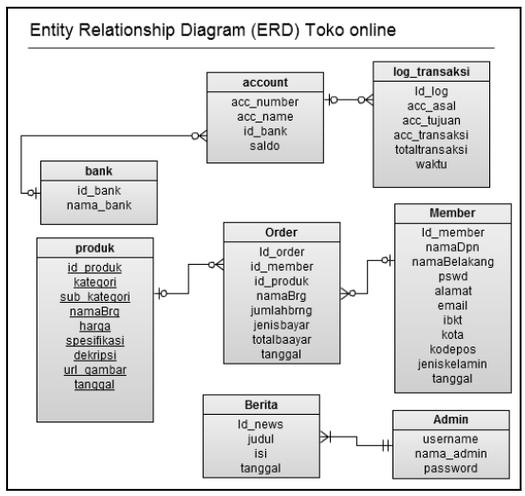
Data-data yang dibutuhkan klien disediakan dalam bentuk file XML. Server PHP akan memperbaharui file ini ketika ada perubahan di *database*. Gambar 3.16 di bawah ini menjelaskan tentang pertukaran data yang dibutuhkan oleh klien.




**Gambar 3. 13 Perancangan klien membaca XML**

**Desain Database**

Desain database merupakan inti dari sebuah aplikasi. Dari desain tersebut, secara tidak langsung kita mengetahui keseluruhan proses sebuah sistem beserta seluruh relasi datanya *(data relatioship).* Pembuatan desain database Entity Relationship Diagram(ERD) gambar 3.17.




**Gambar 3. 14 ERD Diagram toko online**

**Penanganan error pada aplikasi.**

Pada gambar 3.18 ditunjukkan peringatan ketika aplikasi mengalami masalah server tidak aktif. Dengan adanya penanganan ini klien akan mengetahui transaksi untuk sementara waktu belum

bisa diproses dan menunggu beberapa waktu kemudian.


**Gambar 3. 15 Penangan eror**

## 4. Uji Coba dan Analisa

Bagian ini menguraikan lingkungan pengujian untuk aplikasi yang telah dibuat dan analisanya. Lingkungan yang diuraikan meliputi spesifikasi dari perangkat keras dan perangkat lunak yang digunakan dalam menjalankan aplikasi seperti yang terlihat pada Tabel 4.1 di bawah ini :

**Tabel 4. 1 Kebutuhan Uji Coba**

| No | Deskripsi | Spesifikasi |
|---|---|---|
| 1 | CPU | Intel(R) Pentium(R) M processor 1,86 GHz |
| 2 | RAM | 1526 MB |
| 3 | Graphic Card | On Board |
| 4 | Eclipse | Helios |
| 5 | Android SDK | Android Virtual Device, Android Development Tools, Usb Driver |
| 6 | JDBC | mysql-connector-java-5.1.6 |
| 7 | JDK/JRE | Versi 1.6 |
| 8 | Sistem Operasi | Windows XP SP3 |
| 9 | Handphone Android | Samsung Galaxy I551 |
| 10 | Kabel Data+USB Driver | Samsung Kabel Data+Installer |
| 11 | Software Connectify | Hanya tersedia pada Win7 |

**Pelaksanaan Uji Coba dan Analisa**

Pada proyek akhir ini untuk mengetahui berhasil atau tidaknya aplikasi yang telah dibuat ditentukan dari pengujian. Terdapat tiga skenario uji coba yang dilakukan antara lain adalah sebagai berikut:

1. Uji coba enkripsi dan dekripsi dengan berbagai macam data transaksi.
2. Uji coba waktu eksekusi enkripsi dan dekripsi dengan berbagai macam data.
3. Uji coba enkripsi dan dekripsi menggunakan kunci dengan bit yang berbeda.

**Uji Coba Enkripsi dan Dekripsi dengan Berbagai Macam Data**

Uji coba skenario pertama ini ditujukan pada proses transaksi pembelian dengan berbagai macam jenis barang yang berbeda serta klien yang berbeda pula. Tujuan dari uji coba ini adalah untuk mengetahui apakah proses enkripsi data berhasil mengamankan data tersebut. Ukuran aman tidaknya enkripsi adalah pada data yang dihasilkan, apakah berubah dari data aslinya sehingga pesan tidak bisa terbaca secara langsung. Tujuan selanjutnya adalah untuk mengetahui apakah proses dekripsi dapat mengembalikan data sesuai data awal sebelum enkripsi.

Pada uji coba kali dilaksanakan pada dua klien yang berbeda. Masing-masing klien tersebut melakukan transaksi dengan data barang yang berbeda pula. Proses transaksi yang dilakukan oleh setiap klien adalah sebanyak tiga transaksi. Setiap kali proses transaksi akan dicatat pesan/data asli yang akan dienkripsi, hasil enkripsi yang dikirimkan ke server, dan hasil data/pesan yang sudah didekripsi oleh server. Parameter uji coba yang digunakan hanya ada satu yaitu pada tipe kunci yang digunakan. Pada uji coba kali ini tipe kuncinya adalah 2048bits. Hasil uji coba klien pertama ditunjukkan pada tabel 4.2. sedangan untuk klien yang kedua hasilnya ditampilkan pada tabel 4.3.

**Analisa Uji Coba Skenario 1**

Berdasarkan hasil uji coba yang telah dilakukan baik pada klien pertama maupun klien kedua, menunjukkan bahwa data asli dan data yang telah didekripsi memiliki kesamaan. Selain itu data yang telah terenkripsi juga berhasil mengubah data asli menjadi byte-byte karakter yang tidak terbaca secara langsung. Hal ini menunjukkan bahwa proses enkripsi data telah berhasil dilakukan. Uji coba skenario yang pertama ini telah berhasil membuktikan validitas aplikasi yang dibuat. Sehingga algoritma RSA dapat digunakan sebagai alternatif untuk proses enkripsi data pada transaksi mobile.

| No | Data asli | Data hasil enkripsi | Data hasil dekripsi |
|---|---|---|---|
| 1 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Mandiri Transfer:harjo',94312000); | x□ic1□a□□□M□□□=□\|□I"}i□□□`□□+~□□i6J□□□□□O□□s□□M{□□{□□□/□□□□□□Ty□□6□□<0K1&\□ | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Mandiri Transfer:harjo',94312000); |
| 2 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,3,'Ipod 4',1,'Klik BCA:harjo',209000); | Ii□5#>=□L□yR□G□□□□e□EB□4+□?□□F!□e?□n□□□□$□ĶH□□□m□□=□{□□□□1 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,3,'Ipod 4',1,'Klik BCA:harjo',209000); |
| 3 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,4,'core i7 2,4Ghz',1,'Mandiri Transfer:harjo',3009000); | {□□□□□□)A□□□□z□□□,□n□□~□Po□□Q□Y8a"□i□*⌭□-,□□□□□/D□□5□~□□Ö□□Z□□□#□t□fjD□□ | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,4,'core i7 2,4Ghz',1,'Mandiri Transfer:harjo',3009000); |

### 1.1.1. Uji Coba Waktu Eksekusi Enkripsi dan Dekripsi dengan Berbagai Data

Uji coba skenario yang kedua adalah pengujian waktu enkripsi dan dekripsi dengan berbagai macam data. Data yang digunakan sama

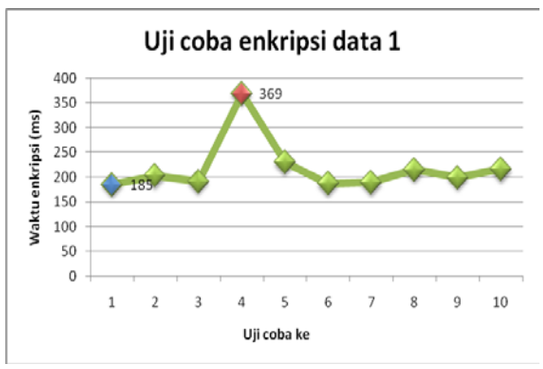
seperti pada uji coba sebelumnya yaitu data transaksi pembelian barang. Tujuan dari uji coba ini adalah untuk mengetahui berapa lama waktu yang diperlukan dalam proses enkripsi dan dekripsi. Parameter tipe kunci yang digunakan 2048 bits.

Langkah uji coba yang dilakukan adalah dengan menggunakan satu data transaksi yang beberapa. Setiap data tersebut diujicoba sebanyak sepuluh kali ulangan. Pada saat uji coba dijalankan akan dicatat data asli yang dienkripsi, panjang data tersebut, waktu enkripsi yang diperlukan, dan waktu dekripsinya. Hasil uji coba disajikan pada table 4.3 dan table 4.4

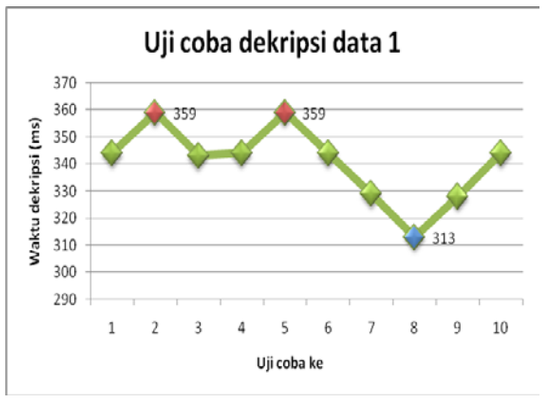
**Tabel 4. 2 (a) Uji coba waktu eksekusi data 1**

| No | Data asli | Pajang data (byte) | Waktu enkripsi (ms) | Waktu dekripsi (ms) |
|---|---|---|---|---|
| 1 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 185 | 344 |
| 2 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 203 | 359 |
| 3 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 191 | 343 |
| 4 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 369 | 344 |
| 5 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 230 | 359 |
| 6 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 187 | 344 |
| 7 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 189 | 329 |
| 8 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 215 | 313 |
| 9 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 199 | 328 |
| 10 | insert into `order`(id_member,id_produk,NamaBrg,jumlahBrg,jenis_bayar,totalbayar) values(1,2,'Ipod 3',1,'Klik BCA:parjo',94309000) ; | 131 | 216 | 344 |

Hasil uji coba ini juga disajikan dalam bentuk grafik. Gambar grafik 4.1 adalah grafik untuk waktu enkripsi, sedangkan grafik 4.2 adalah grafik waktu dekripsi. Waktu uji coba pertama mendapat nilai 185 ms yang merupakan waktu tercepat ditunjukkan dengan warna biru. Enkripsi data yang dilakukan pada uji coba ke empat menghasilkan nilai 369 ms. Data ini merupakan nilai tertinggi dari sepuluh kali percobaan dan ditunjukan dengan warna merah.




**Gambar 4. 1 Grafik uji coba enkripsi data 1**




**Gambar 4. 2 Grafik uji coba dekripsi data 1**

Berdasarkan grafik hasil uji coba 4.2 di atas, terlihat bahwa dari sepuluh kali percobaan yang dilakukan terdapat dua nilai yang sama yaitu pada uji coba ke-2 dan ke-5 yaitu 359 ditandai dengan warna merah. Nilai tersebut adalah waktu eksekusi terlama. Sedangkan waktu yang tercepat dengan nilai 313 ms ditandai dengan warna biru. Nilai ini merupakan hasil uji coba pada ulangan ke-8

**Tabel 4. 3 Summary hasil uji coba skenario 2**

| | Uji coba data 1 (ms) | | |
|---|---|---|---|
| | Rata-rata | Maks | Min |
| Enkripsi | 218.4 | 365 | 189 |
| Dekripsi | 340.7 | 359 | 313 |

Walaupun hasil uji coba ini menunjukkan waktu yang lama untuk beberapa percobaan, namun secara umum hasilnya menunjukkan bahwa waktu eksekusi yang dibutuhkan oleh aplikasi tidak terlalu besar. Proses dekripsi berjalan lebih lama dibandingkan dengan enkripsi karena panjang data input untuk proses dekripsi jauh lebih panjang. Sekalipun proses dekripsi membutuhkan waktu yang lebih lama, namun proses ini hanya dilakukan oleh komputer server sehingga tidak akan membebani komputasi yang dilakukan oleh device mobile. Dengan hasil ini dapat disimpulkan bahwa algoritma RSA memungkinkan untuk dimplementasikan pada data transaksi dalam aplikasi mobile.

### 1.1.2. Uji Coba Menggunakan Kunci dengan Byte yang Berbeda

Pada skenario uji coba yang ketiga ini akan dilakukan uji coba dengan tipe bit kunci yang berbeda-beda. Uji coba ini bertujuan untuk mengetahui hasil enkripsi dengan kunci yang memiliki bit berbeda-beda serta untuk mengetahui batasan panjang data input.

Data yang digunakan untuk uji coba kali ini berbeda dengan uji coba sebelumnya, karena tidak menggunakan data transaksi, namun menggunakan data random. Data uji coba yang digunakan adalah karakter angka yang dibuat dengan ulangan pola tertentu. Langkah uji coba yang dilakukan adalah dengan membuat kunci dengan berbagai bit, dimulai dari 512 bits dan dilanjutkan dengan kelipatan 256 bits sampai dengan bit terbesar 2048 bits. Kemudian uji coba dilakukan dengan memasukkan data dengan byte mencapai panjang maksimum setiap kunci.

Setiap pelaksanaan uji coba akan dicatat besar kunci bit yang digunakan, data input yang digunakan, panjang data input dan panjang cipher text yang dihasilkan. Uji coba dilaksanakan sebanyak tujuh kali dan hasilnya tersaji pada table 4.14 di bawah ini.

| No | Kunci (bits) | Data asli | Panjang data (byte) | Panjang cipher data (byte) |
|---|---|---|---|---|
| 1 | 512 | 1234567890123456 7890123456789012 3456789012345678 90123 | 53 | 64 |
| 2 | 768 | 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 12345 | 85 | 96 |
| 3 | 1024 | 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 34567 | 117 | 128 |
| 4 | 1280 | 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 3456789012345678 9012345678901234 56789 | 149 | 160 |
| 5 | 1536 | 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 78901 | 181 | 192 |
| 6 | 1792 | 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 3456789012345678 90123 | 213 | 224 |
| 7 | 2048 | 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 1234567890123456 7890123456789012 3456789012345678 9012345678901234 5678901234567890 12345 | 245 | 256 |

**Analisa Uji Coba Skenario 3**

Berdasarkan hasil uji coba pada tabel 4.15 di atas terlihat bahwa panjang kunci (bit) yang digunakan dalam proses enkripsi akan mempengaruhi input dan hasil enkripsi. Jika bit kunci semakin besar maka batasan panjang data input yang bisa dimasukkan juga semakin besar, begitu juga dengan data enkripsi yang dihasilkan akan bertambah besar. Berdasarkan analisa 4.3.4 pada uji coba sebelumnya membuktikan bahwa semakin besar data terenkripsi, maka semakin lama juga waktu yang diperlukan untuk mendekripsinya.

**Analisa Kegagalan Transaksi Beserta Solusinya**

Pada saat proses transaksi, sangat memungkinkan terjadinya berbagai macam kegagalan selam proses tersebut. Misalnya server aplikasi sedang dalam keadaan tidak beroperasi, maka secara

otomatis aplikasi klien tidak bisa terhubung ke server. Sehingga proses transaksi tidak bisa dilakukan. Untuk menangani kegagalan seperti ini, user akan diberikan peringatan yang memberitahukan bahwa transaksi tidak bisa dilanjutkan. Penjelasan lebih lengkap mengenai penanganan error pada aplikasi dapat dilihat pada subbab penanganan eror aplikasi

## 5. Kesimpulan dan Saran

### KESIMPULAN

Dari hasil pengujian dan analisa pada bab 4, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Algoritma RSA memungkinkan untuk dimplementasikan pada data transaksi dalam aplikasi mobile.
2. RSA membuktikan bahwa semakin besar data terenkripsi, maka semakin lama juga waktu yang diperlukan untuk mendekripsinya.
3. Metode ini juga membuktikan semakin panjang kunci public dan private semakin lam prosesnya.

### Saran

Hasil dari proyek akhir ini masih belum sempurna, oleh karena itu ada beberapa saran yang mungkin dapat menjadi masukan bagi adik kelas yang ingin mengembangkan aplikasi ini sehingga menjadi sistem yang lebih kompleks yaitu sistem dengan menggunakan enkripsi dua arah yaitu simteris dan asimetris. Metode enkripsi dua arah ini digunakan karena tidak memberatkan klien untuk mengirim data.

## 6. Daftar Pustaka


[1] Arinta Nugrahani Ayuningtyas, “Implementasi metode RSA pada priority dealer untuk layanan penjualan dan pemesanan handphone berbasis J2ME”, Proyek akhir PENS-ITS, 2011.

[2] Hernawan Sulistyanto, “Autentikasi dalam Basis Data Jaringan Menggunakan Kriptosistem Kunci Publik RSA”,Teknik Elektro Universitas Muhamadiyah Surakarta, 2004.

[3] Sarwono Sutikno,*”Kripto Kunci Publik RSA”*, Teknik Elektro-ITB, Bandung,

[4] Prasetyo Andy Wicaksono,*” Studi Pemakaian Algoritma Rsa Dalam Proses Enkripsi dan Aplikasinya”*, Teknik informatika-ITB, Bandung, 2005.

[5] http://www.javamex.com/tutorials/cryptography/rsa_encryption_.shtml waktu akses : 24 Juli 2011, 14:45

[6] Rinaldi Munir, ”*Kriptografi*”, Informatika, Bandung, 2006.

[7] http://www.androidpeople.com/android-xml-parsing-tutorial-using-saxparser waktu akses : 24 Juli 2011, 15:00

[8] http://developer.android.com/guide/practices/design/accessibility.html waktu akses : 24 Juli 2011, 15:45