PENGEMBANGAN AUTHENTIFIKASI QR CODE MENGGUNAKAN ALGORITMA RSA PADA APLIKASI DELIVERY ORDER MAKANAN BERBASIS ANDROID

[1]Ahmat Yavi Yulian (13 1065 1133), [2]Victor Wahanggara, S.Kom., M.Kom,
[1]ahmatyaviyulian82@gmail.com

Universitas Muhammadiyah Jember
Jln. Karimata No. 49, Telp (0331) 336728, Jember

ABSTRAK

Pengiriman makanan (*delivery order*) merupakan salah satu layanan makanan siap saji yang popular. Selain mempermudah konsumen dalam mendapatkan makanan, layanan ini juga membantu meningkatkan penjualan bagi perusahaan (rumah makan). Kebanyakan masyarakat modern saat ini cenderung lebih menyukai memesan makanan untuk diantar kerumah dan menikmatinya di rumah. Kabupaten *Jember* adalah salah satu kota yang memiliki banyak rumah makan jenis apapun, semua rumah makan dikota ini bersaing dalam menghidangkan makanan dan segi pelayanan. Pemberian sandi untuk pengaman data berupa *QR Code* yang telah terenkripsi adalah hal yang paling ampuh dalam mengatasi Penggandaan pesanan atau yang dimaksud order fiktif oleh karena itu konsumen, driver atau pelapak dapat melakukan proses jual beli secara real dan terpantau. Dengan menggunakan sebuah ponsel yang memiliki sistem operasi dan dapat mengakses internet, metode transaksi yang digunakan nantinya adalah dengan memanfaatkan kode batang Qr Code dengan enkripsi algoritma RSA. Jadi, data berupa pesanan konsumen nantinya akan dienkripsi otomatis oleh sistem dan digenerate ke dalam Qr Code. Hasil dari penelitian ini berupa kode batang Qr Code konsumen terenkripsi, sehingga dalam proses transaksi kurir pengantar menscan langsung Qr Code konsumen jika asli sistem akan otomatis memberikan status terbayar, namun jika palsu status akan tetap sehingga kurir akan dapat mengetahui dengan aplikasi scanning khusus yang dimilikinya.

Kata Kunci : *Qr Code, algoritma RSA, kriptografi, Delivery Order.*

1.1 Latar Belakang

Pengiriman makanan (*delivery order*) merupakan salah satu layanan makanan siap saji yang popular. Selain mempermudah konsumen dalam mendapatkan makanan, layanan ini juga membantu meningkatkan penjualan bagi perusahaan (rumah makan). Kebanyakan masyarakat modern saat ini cenderung lebih menyukai memesan makanan untuk diantar kerumah dan menikmatinya di rumah. Kabupaten *Jember* adalah salah satu kota yang memiliki banyak rumah makan jenis apapun, semua rumah makan dikota ini bersaing dalam menghidangkan makanan dan segi pelayanan.

Menurut data pada tahun 2014, pulau jawa dan bali memiliki kontribusi lebih dari 73, 6 persen dari total jumlah restoran dan rumah makan di Indonesia. Peningkatan jumlah rumah makan membuat persaingan antar rumah makan menjadi semakin ketat, sehingga masing-masing rumah makan berusaha untuk meningkatkan kualitas pelayanan. Salah satu usaha yang dilakukan yakni dengan menyediakan jasa *delivery order*. Jasa *delivery order* ditawarkan sejumlah rumah makan, dengan cara melakukan pemesanan melalui ponsel. Seiring dengan berkembangnya teknologi informasi, beberapa rumah makan ternama menyediakan aplikasi-aplikasi *mobile* untuk mempermudah melakukan pemesanan makanan secara *online* (Firmansyah, 2014).

Meskipun *delivery Order* merupakan penyedia layanan yang memberikan kemudahan dalam melakukan komunikasi pemesanan, namun dalam kenyataannya masih banyak terdapat kendala, yaitu masih adanya salah kirim pesanan atau ketidaksesuaian pesanan konsumen bisa dikarenakan informasi yang salah atau duplikasi data informasi itu sendiri sehingga tentu akan berdampak kerugian bagi kedua belah pihak, hal ini tentunya dapat menimbulkan kerugian bagi kedua belah pihak, baik konsumen maupun jasa delivery order.

Berdasarkan latar belakang masalah tersebut, maka akan dibuat suatu aplikasi berbasis *mobile* android, pada delivery order makanan, dengan memanfatkan perangkat android yang terintegrasi kode batang *QR Code* dan nantinya akan diekripsi menggunakan sandi *Algoritma RSA* yang memiliki dua kunci yaitu kunci privat dan kunci publik, sehingga akan memiliki tingkat keamanan yang baik untuk mendukung tugas akhir dengan judul **"*Pengembangan Authentifikasi Qr Code Menggunakan Algoritma RSA Pada Aplikasi Delivery Order Makanan Berbasis Android"*.**

Pada penelitian sebelumnya menggunakan algoritma vigenere untuk proses pengamannya yaitu menggunakan kunci simetris (*one key*) yang sangat mudah untuk di pecahkan dengan menggunakan *brute force*, maka dari itu digunakan algoritma yang lebih aman yaitu menggunakan algoritma RSA yang menggunakan kunci asimetris (kunci privat dan publik).

1.2 *Rumusan* Masalah

Berdasarkan latar belakang yang diuraikan sebelumnya, terdapat beberapa permasalahan yang akan diangkat dalam penelitian ini, antara lain :

1. Bagaimana menerapkan algoritma RSA pada Qr Code pemesanan ?
2. Bagaimana pengujian aplikasi delivery order makanan menggunakan algoritma RSA pada authentifikasi Qr code berbasis android?

1.3 Batasan Masalah

Agar tidak menyimpang jauh dari permasalahan, maka penelitian ini mempunyai batasan masalah sebagai berikut :

1. Area rumah makanan yang digunakan masih sekitar kabupaten Jember (khususnya area kota).
2. Makanan dan minuman memiliki minimum *order* dari setiap rumah makan.
3. Pemesan hanya dapat melakukan satu transaksi pemesanan untuk satu alamat.

4. Pendaftaran atau *Regristrasi* hanya dapat dilakukan oleh admin.
5. Biaya delivery sudah tercantum pada nota pembayaran.
6. Notifikasi pemesanan akan disampaikan kepada pengguna kurang lebih setelah 5 menit pemesanan.
7. Rumah makan yang dipilih dapat memasukkan menu terbaru, promo dan lain-lain secara langsung.
8. Lokasi delivery order hanya kawasan kabupaten jember (khusus area Kota).
9. Aplikasi yang dibangun adalah aplikasi berbasis *mobile.*

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah di atas maka tujuan dari penelitian ini adalah:

1. Menghasilkan QR Code pemesanan yang menerapkan sistem pengamanan menggunakan algoritma RSA sehingga pemesanan lebih aman.
2. Menghasilkan beberapa pengujian dari aplikasi sehingga meminimalkan bug / kesalahan aplikasi serta memberikan kemudahan pengguna.

1.5 Manfaat Penelitian

Penelitian ini dilakukan dengan harapan dapat memberikan manfaat diantaranya sebagai berikut :

1. Mempermudah konsumen atau *user* dalam melakukan proses pemesanan makanan atau minuman.
2. Meningkatkan laba dan dapat menjaring lebih banyak pelanggan bagi rumah makan.
3. Mengurangi kemungkinan konsumen tidak mendapatkan tempat makan karena ramai.

II. LANDASAN TEORI

2.1 Definisi Restoran dan Rumah Makan.

Restoran adalah salah satu jenis usaha jasa pangan yang bertempat di sebagian atau seluruh bangunan yang permanen, dilengkapi dengan peralatan dan perlengkapan untuk proses pembuatan, penyimpanan, penyajian, dan penjualan makanan dan minuman bagi umum di tempat usahanya dan memenuhi ketentuan persyaratan yang ditetapkan dalam keputusan ini (Keputusan Menteri Pariwisata, Pos dan Telekomunikasi Nomor KM. 95/HK.103/MPT-87).

Menurut Marsum (2008), restoran adalah tempat atau bangunan yang diorganisir secara komersial, yang menyelenggarakan pelayanan dengan baik kepada tamunya baik berupa makanan atau minuman. Restoran ada yang berada dalam suatu hotel, kantor, maupun pabrik, dan banyak juga yang berdiri sendiri diluar bangunan itu. Restoran merupakan suatu tempat atau bangunan yang terorganisasi secara komersil, yang menyelenggarakan pelayanan dengan baik kepada semua tamunya baik berupa makan maupun minum.

2.2 Definisi Makanan dan Minuman

Pengertian makanan sehat adalah makanan yang mengandung zat zat yang dibutuhkan oleh tubuh. Makanan sehat mengandung gizi yang seimbang, yaitu makanan yang sarat

gizi dan baik dikonsumsi oleh tubuh. Mengetahui hubungan antara makanan yang dikonsumsi dan air (Hardani,2012).

2.3 Pengertian Layanan Antar *(Delivery Order)*

Pengertian Layanan antar (Home delivery service) menurut beberapa ahli yaitu :

1. "Home-delivery services where meals are delivered at home of the person who orders the meals." Sudhir Andrews (2009).
2. "Where prepared food is brought to the customer's home" Kaye Chon dan Thomas A. Maier (2010).
3. "Delivery service relies heavily on telephone orders, with an increasing number of restaurants accepting delivery order via the internet." Regina S. Barbaran dan Joseph F. Durocher (2010).

Dari beberapa pengertian di atas, maka penulis menyimpulkan bahwa pengertian layanan antar adalah suatu aktivitas dan pemberian jasa dimana customers memesan produk yang disediakan produsen dan biasanya menggunakan media komunikasi melalui telepon atau internet lalu produk yang dipesan akan diantarkan sampai ke tempat tujuan customers tanpa customers perlu untuk datang dan bertemu langsung dengan penjual / produsen.

2.4 Android

Android adalah sistem operasi mobile berbasis *open source* yang di miliki raksasa internet saat ini, *Google*. Android dikembangkan dengan menggunakan kernel linux. Android memungkinkan untuk di modifikasi secara bebas dan di distibusikan oleh pembuat perangkat tersebut. Dengan sifat *open source* tersebut telah banyak mendorong komunitas pengembang aplikasi untuk menggunakan *source code* android sebagai dasar proyek pembuatan aplikasi.

Android dimulai sebagai sebuah start up rahasia pada tahun 2003, dan dibeli oleh Google pada tahun 2005 dan sebagai jalan google untuk memasuki pasar perangkat lunak bergerak. *Handphone* komersil pertama yang menggunakan OS Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008. Dikutip dari okezone.com (2013), terungkap pula sebanyak 4,5 juta smartphone yang berhasil terjual di Indonesia selama Januari sampai Maret 2013, sebanyak 2,28 juta di antaranya menjalankan OS Android.

Sumber : *https://id.wikipedia.org/wiki/Daftar_versi_Android* dan *http://developer.android.com/index.html*

2.5 Quick Response Code

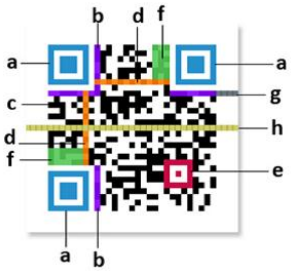
Quick Response Code sering di sebut *Qr Code* atau Kode *QR* adalah semacam simbol dua dimensi yang dikembangkan oleh Denso Wave yang merupakan anak perusahaan dari Toyota sebuah perusahaan Jepang pada tahun 1994. Tujuan dari *Qr Code* ini adalah untuk menyampaikan informasi secara cepat dan juga mendapat tanggapan secara cepat. Pada awalnya *Qr Code* digunakan untuk pelacakan bagian kendaraan untuk *manufacturing*. Namun sekarang, telah digunakan untuk komersil yang ditujukan pada pengguna telepon seluler. *Qr Code* adalah perkembangan dari

barcode atau kode batang yang hanya mampu menyimpan informasi secara horizontal sedangkan *QR Code* mampu menyimpan informasi lebih banyak, baik secara horizontal maupun vertikal.

Gambar 2.1 Contoh *Qr Code"Dokumen Sertifikat Hak Milik Tanah Doni Prayugo Agung Pribadi"*

QR Code biasanya berbentuk persegi putih kecil dengan bentuk geometris hitam (dapat dilihat di gambar 2.1), meskipun sekarang banyak yang telah berwarna dan digunakan sebagai brand produk. Informasi yang dikodekan dalam *QR Code* dapat berupa URL, nomor telepon, pesan SMS, *V-Card*, atau teks apapun (Ashford,2010). *QR Code* telah mendapatkan standarisasi internasional ISO/IEC18004 dan Jepang JIS-X-0510 (Denso, 2011).

2.5.1 Anatomi Qr Code



Gambar 2.2 Anatomi *Qr Code*

Beberapa penjelasan anatomi *Qr Code* Menurut (Ariadi, 2011) antara lain :

1. *Finder Pattern* berfungsi untuk identifikasi letak *Qr Code.*
2. *Format Information* berfungsi untuk informasi tentang *error correction level dan mask pattern.*
3. *Data* berfungsi untuk menyimpan data yang dikodekan.
4. Timing Pattern merupakan pola yang berfungsi untuk identifikasi koordinat pusat *Qr Code*, berbentuk modul hitam putih.
5. *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan *Qr Code* terutama distorsi non linier.
6. *Version Information* adalah versi dari sebuah *Qr Code*.
7. *Quiet Zone* merupakan daerah kosong di bagian terluar *QR Code* yang mempermudah mengenali pengenalan *QR* oleh sensor *CCD.*
8. *Qr Code* version adalah versi dari *Qr Code* yang digunakan.

2.5.2 Versi Qr Code

Gambar 2.3 Versi *Qr Code*
(Sumber : qrcode.com)

Qr Code dapat menghasilkan 40 versi yang berbeda dari versi 1 (21 x 21 modul) sampai versi 40 (177 x 177 modul). Tingkatan Versi *Qr Code* 1 dan 2 berbeda 4 modul berlaku sampai dengan versi 40. Setiap versi memiliki konfigurasi atau jumlah modul yang berbeda. Modul ini mengacu pada titik hitam dan putih yang membentuk suatu *QR Code*. Setiap versi *QR Code* memiliki kapasitas maksimum data, jenis karakter dan tingkat koreksi kesalahan. Jika Jumlah data yang ditampung banyak maka modul yang yang akan diperlukan dan menjadikan

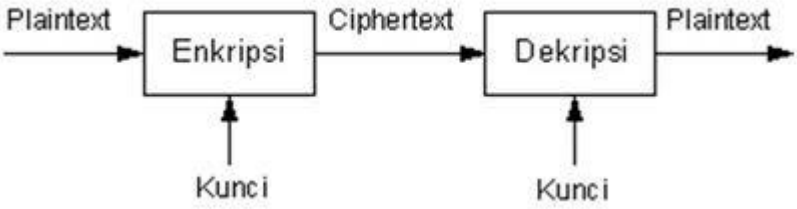
Qr Code menjadi lebih besar (Denso, 2011).

2.6 Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *cryptós* yang artinya *"secret"* (yang tersembunyi) dan *gráphein* yang artinya *"writting"* (tulisan). Jadi, kriptografi berarti "secret writting" (tulisan rahasia). Definisi yang dikemukakan oleh Bruce Schneier (1996), kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan *(Cryptography is the art and science of keeping messages secure).* Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (message). Algoritma kriptografi adalah :

1. Aturan untuk enkripsi *(enciphering)* dan dekripsi *(deciphering).*
2. Fungsi matematika yang digunakan untuk enkripsi dan dekripsi.

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext.* Proses yang dilakukan untuk mengubah plaintext ke dalam ciphertext disebut *encryption* atau *encipherment.* Sedangkan proses untuk mengubah ciphertext kembali ke plaintext disebut *d*
Secara se
dapat digar



Gambar 2.10 Proses Enkripsi/Dekripsi Sederhana

Algoritma kriptografi berkembang terus dan terbagi atas dua bagian yaitu algoritma kriptografi klasik dan modern. Pada kriptografi klasik, kriptografer menggunakan algoritma sederhana, yang memungkinkan cipherteks dapat dipecahkan dengan mudah (melalui penggunaan statistik, terkaan, intuisi, dan sebagainya). Algoritma kriptografi modern dibuat sedemikian kompleks sehingga kriptanalis sangat sulit untuk memecahkan cipherteks tanpa mengetahui kunci. Pengelompokan algoritma juga dilakukan berdasarkan kunci enkripsi – dekripsi yang digunakan, yaitu *symmetric cryptosystem* atau simetris (menggunakan kunci yang sama untuk proses enkripsi – dekripsi) dan *Assymmetric cryptosystem* atau asimetris (menggunakan kunci yang berbeda untuk proses enkripsi – dekripsi).

2.7 Sandi *Algoritma RSA*

RSA di bidang kriptografi adalah sebuah algoritma pada enkripsi *public key*. RSA merupakan algoritma pertama yang cocok untuk *digital signature* seperti halnya ekripsi, dan salah satu yang paling maju dalam bidang kriptografi *public key*. RSA masih digunakan secara luas dalam protokol *electronic commerce*, dan dipercaya dalam mengamankan dengan
ng cukup panjang.
ıat pada tahun 1978.
n dari nama para
Rivest, Adi Shamir,
. RSA adalah salah
ndian yang paling
banyak mengundang kontroversi, selain DES. Sejauh ini belum seorang pun yang berhasil menemukan lubang sekuriti pada DES dan RSA, tetapi tak seorang pun juga yang berhasil memberikan pembuktian ilmiah yang memuaskan dari keamanan kedua teknik sandi ini.

Untuk menyandi informasi dan untuk menerjemahkan pesan tersandi

sebuah algoritma penyandian memerlukan sebuah data biner yang disebut kunci. Tanpa kunci yang cocok orang tidak bisa mendapatkan kembali pesan asli dari pesan tersandi. Pada DES digunakan kunci yang sama untuk menyandi (enkripsi) maupun untuk menterjemahan (dekripsi), sedangkan RSA menggunakan dua kunci yang berbeda. Isitilahnya, DES disebut sistem sandi simetris sementara RSA disebut sistem sandi asimetris. Kedua sistem ini memiliki keuntungan dan kerugiannya sendiri. Sistem sandi simetris cenderung jauh lebih cepat sehingga lebih disukai oleh sementara kalangan industri. Kejelekannya, pihak-pihak yang ingin berkomunikasi secara privat harus punya akses ke sebuah kunci DES bersama. Walaupun biasanya pihak-pihak yang terkait sudah saling percaya, skema ini memungkinkan satu pihak untuk memalsukan pernyataan dari pihak lainnya. RSA yang menggunakan algoritma asimetrik mempunyai dua kunci yang berbeda, disebut pasangan kunci (key pair) untuk proses enkripsi dan dekripsi. Kunci-kunci yang ada pada pasangan kunci mempunyai hubungan secara matematis, tetapi tidak dapat dilihat secara komputasi untuk mendeduksi kunci yang satu ke pasangannya. Algoritma ini disebut kunci publik, karena kunci enkripsi dapat disebarkan. Orang-orang dapat menggunakan kunci publik ini, tapi hanya orang yang mempunyai kunci privat sajalah yang bisa mendekripsi data tersebut.

i. Cara kerja sandi algoritma RSA

Tingkat keamanan algoritma penyandian RSA sangat bergantung pada ukuran kunci sandi tersebut (dalam bit), karena makin besar ukuran kunci, maka makin besar juga kemungkinan kombinasi kunci yang bisa dijebol dengan metode mengencek kombinasi satu persatu kunci atau lebih dikenal dengan istilah brute force attack. Jika dibuat suatu sandi RSA dengan panjang 256 bit, maka metode brute force attack akan menjadi tidak ekonomis dan sia-sia dimana para hacker pun tidak mau/sanggup untuk menjebol sandi tersebut.

ii. Proses Pembuatan Kunci

Dalam membuat suatu sandi, RSA mempunyai cara kerja dalam membuat kunci publik dan kunci privat adalah sebagai berikut:

1. Pilih dua bilangan prima p dan q secara acak , $p \neq q$. Bilangan ini harus bilangan prima.
2. Hitung $N = pq$. Bilangan N disebut *parameter sekuriti.*
3. Hitung $\varphi = (p\text{-}1)(q\text{-}1)$.
4. Pilih bilangan bulat (*integer*) antara satu dan φ ($1 < e < \varphi$) yang tidak mempunyai faktor pembagi dari φ.
5. Hitung d hingga $d\,e \equiv 1 \pmod{\varphi}$.

Keterangan :

a. Langkah 3 dan 4 dapat dihasilkan dengan cara algoritma Euclidean
b. Langkah 4 dapat dihasilkan dengan menemukan integer x sehingga $d = (x(p\text{-}1)(q\text{-}1) + 1)/e$ menghasilkan bilangan bulat, kemudian menggunakan nilai dari $d \pmod{(p\text{-}1)(q\text{-}1)}$.

Setelah melalu cara ini, maka kita akan mendapatkan kunci publik dan kunci privat. Kunci publik terdiri dari dua elemen, yaitu :

a. N, merupakan modulus yang digunakan
b. e, eksponen publik atau eksponen enkripsi.

dan kunci privat, yang terdiri dari:

a. *N*, merupakan modulus yang digunakan, sama seperti pada kunci publik.
b. *d*, eksponen pribadi atau eksponen deskripsi, yang harus dijaga kerahasiaanya.

Nilai *p* dan *q* sebaiknya dibuang atau dijaga kerahasiaannya, karena terdapat *N* dimana p dan q adalah faktor pembagi dari *N*. Walaupun bentuk ini memperbolehkan dekripsi secara cepat dan signing menggunakan *Chinese Remainder Theorem* (CRT), hal ini mejadi lebih tidak aman karena bentuk ini memperbolehkan *side channel attacks*. *Side channel attacks* adalah sebuah serangan yang berdasarkan informasi yang dikumpulkan dari implementasi fisik (atau kelemahan secara fisik) dari sebuah sistem kriptografi, dibanding dengan kelemahan teoritis dari algoritmanya sendiri. Sebagai contohnya, faktor-faktor kurun waktu dari informasi, konsumsi tenaga, bahkan suara yang ditimbulkan dapat membantu mempermudah informasi yang bisa diambil untuk menjebol sistem tersebut.

iii. Proses Enkripsi Pesan

Misalkan pada suatu kasus si A ingin mengirim pesan m kepada si B. A mengubah m menjadi angka $n < N$, menggunakan protokol yang sebelumnya telah disepakati dan dikenal sebagai *padding scheme*. *Padding scheme* harus dibangun secara hati-hati sehingga tidak ada nilai dari *m* yang menyebabkan masalah keamanan. Contohnya, jika kita ambil contoh sederhana dari penampilan ASCII dari *m* dan menggabungkan bit-bit secara bersama-sama akan menghasilkan *n*, kemudian pessan yang berisi ASCII tunggal karakter NUL (nilai numeris 0) akan menghasilkan $n = 0$, yang akan menghasilkan *ciphertext* 0 apapun itu nilai dari *e* dan *N* yang digunakan.

Maka A mempunyai nilai *n* dan mengetahui *N* dan *e*, yang telah diumumkan oleh B. A kemudian menghitung *ciphertext c* yang terkait pada *n*:

$$c = n^e \mod N$$

Perhitungan tersebut dapat diselesaikan dengan menggunakan metode *exponentation by squaring*, yaitu sebuah algoritma yang dipakai untuk komputasi terhadap sejumlah nilai integer yang besar dengan cepat. Kemudian A mengirimkan nilai *C* kepada B.

iv. Proses Dekripsi Pesan

B sudah menerima *C* dari A, dan mengetahui kunci privat yang digunakan B. B kemudian mengembalikan nilai *n* dari *C* dengan langkah-langkah sebagai berikut:

$$n = c^d \mod N$$

Perhitungan diatas akan menghasilkan *n*, dengan begitu B dapat mengembalikan pesan semula *m*. Prosedur dekripsi bekerja karena.

$$c^d \equiv (n^e)^d \equiv n^{ed} \pmod N$$

Kemudian, karena $ed \equiv 1 \pmod{p-1}$ dan $ed \equiv 1 \pmod{q-1}$, hasil dari *Fermat's little theorem*.

$$n^{ed} \equiv n \pmod p$$

dan

$$n^{ed} \equiv n \pmod q$$

Karena *p* dan *q* merupakan bilangan prima yang berbeda, mengaplikasikan *Chinese*

remainder theorem akan menghasilkan dua macam kongruen.

$$n^{ed} \equiv n \pmod{pq}$$

serta

$$c^{d} \equiv n \pmod{N}$$

V. Contoh Penghitungan RSA

Sekarang kita mencoba suatu contoh untuk mengenal lebih dalam sistem kerja enkripisi RSA. Misalnya kita mau mengenkripsi kata *"SECRET"* dengan RSA, lalu kita dekripsi kembali ke dalam *plain text*.

Karena *p* dan *q* berjumlah minimal 100 digit atau lebih, nilai *d* dan *e* bisa berjumlah sama dengan 100 digit dan nilai *N* akan berjumlah 200 digit. Untuk itu di contoh pemakaian berikut, kita akan memakai angka-angka yang kecil agar mudah dalam penghitungan. Cara pengerjaannya adalah:

1. Kita pilih $p = 3$ dan $q = 5$
2. Hitung $N = pq = 3*5 = 15$
3. Nilai *e* harus merupakan bilangan prima yang lebih besar dan relatif dekat dengan (p-1)(q-1) = (2)(4) = 8, sehingga kita pilih e = 11. Angka 11 adalah bilangan prima terdekat dan lebih besar daripada 8.
4. Nilai *d* harus dipilih sehingga,

$$\frac{(ed-1)}{(p-1)(q-1)}$$

adalah sebuah integer. Lalu nilai
(11 *d* - 1) / [(2)(4)] = (11d - 1) / 8
juga merupakan integer. Setelah melalui proses penghitungan, salah satu nilai yang mungkin adalah $d = 3$.

5. Lalu kita masukkan kata yang akan dienkripsi, "SECRET". Kita akan mengkonversi string ini ke representasi desimal menggunakan nilai karakter ASCII, yang akan menghasilkan nilai ASCII 83 69 67 82 69 84
6. Pengirim akan mengenkripsi setiap digit angka pada saat yang bersamaan menggunakan nilai kunci publik (*e*, *n*) = (11,15). Lalu setiap karakter *ciphertext* akan masuk ke persamaan $C_i = M_i^{11} \bmod 15$. Yang akan menghasilkan nilai digit masukan adalah 0x836967826984 yang akan dikirim sebagai 0x2c696d286924.
7. Penerima akan mendekripsi setiap digit angka menggunakan nilai kunci privat (*d*, *n*) = (3, 15). Lalu, setiap karakter plaintext akan masuk persamaan $M_i = C_i^3 \bmod 15$. String masukan yang bernilai 0x2c696d286924, akan dikonversi kembail menjadi 0x836967826984, dan akhirnya angka-angka tersebut akan diubah kembali menjadi bentuk string *plaintext* yang bernilai "SECRET".

Dari contoh di atas kita dapat menangkap suatu kelemahan dari pemakaian *p* dan *q* yang bernilai kecil yaitu bisa kita lihat di digit ke-4, ke-6 dan ke-9 tidak berubah saat dienkripsi, dan nilai 2 dan 8 dienkripsi menjadi 8

dan 2, yang berarti dienkripsi menjadi kebalikannya. Tapi kesimpulan yang bisa diambil dari contoh yang sederhana ini adalah RSA dapat digunakan dalam penyandian dalam pengiriman informasi.

Kunci RSA yang mempunyai ukuran 512 dan 768 bit dianggap masih lemah dan mudah dijebol. Ukuran kunci yang dianjurkan adalah 1024 bit. Ukuran 2048 dan 3072 bit merupakan suatu ukuran yang lebih baik.

vi. Keamanan RSA

Keamanan dari sistem kriptografi RSA adalah didasari oleh dua problem matematika:

1. Problem dalam faktorisasi bilangan berjumlah banyak.
2. Problem RSA, yaitu mencari modulo akar en dari sebuah bilangan komposit (yang faktor-faktornya tidak diketahui.

Proses dekripsi penuh dari sebuah ciphertext RSA dianggap sesuatu hal yang tidak mudah karena kedua problem ini diasumsikan sulit. Belum ada algoritma yang mampu untuk menyelesaikannya. Problem RSA didefinisikan sebagai tugas untuk mencari suatu akar modulo e^n (*e* pangkat ke *n*) dari bilangan komposit *N*. Mengembalikan suatu nilai *m* dimana $m^e=c$ mod n, (*e, n*) adalah kunci publik RSA dan *c* adalah *ciphertext* RSA.

Metode pendekatan yang diyakini dapat menyelesaikan problem RSA saat ini adalah memfaktor dari modulus n. Dengan kemampuan untuk mengembalikan faktor yang merupakan bilangan prima, sebuah serangan dapat menghitung eksponen rahasia dari *d* dan dari kunci publik (*e, n*), lalu mendekripsi *c* menggunakan prosedur standar. Untuk menyelesaikannya, penyerang (bisa penyadap, penguping, dll.) memfaktor nilai *n* menjadi *p* dan *q*, lalu menghitung (p-1)(q-1) yang dapat menghasilkan nilai *d* dan *e*.

Penyerangan yang paling umum pada RSA ialah pada penanganan masalah faktorisasi pada bilangan yang sangat besar. Apabila terdapat faktorisasi metode yang baru dan cepat telah dikembangkan, maka ada kemungkinan untuk membongkar RSA.

Pada tahun 2005, bilangan faktorisasi terbesar yang digunakan secara umum ialah sepanjang 663 bit, menggunakan metode distribusi mutakhir. Kunci RSA pada umumnya sepanjang 1024—2048 bit. Beberapa pakar meyakini bahwa kunci 1024-bit ada kemungkinan dipecahkan pada waktu dekat (hal ini masih dalam perdebatan), tetapi tidak ada seorangpun yang berpendapat kunci 2048-bit akan pecah pada masa depan yang terprediksi. Jika *N* sepanjang 256-bit atau lebih pendek, maka kunci RSA akan dapat ditemukan dalam beberapa jam hanya dengan menggunakan PC, dengan menggunakan perangkat lunak yang tersedia. Jika *N* sepanjang 512-bit atau lebih pendek, *N* akan dapat difaktorisasi dalam hitungan ratusan jam seperti pada tahun 1999 dengan menggunakan ratusan komputer. Secara teori, perangkat keras bernama TWIRL dan penjelasan dari Shamir dan Tromer pada tahun 2003 mengundang berbagai pertanyaan akan keamanan dari kunci 1024- bit. Saat ini disarankan bahwa *N* setidaknya sepanjang 2048-bit.

vii. Membuat RSA Sukar Dijebol

Jika nilai *N* berjumlah kecil, maka nilai faktor *p* dan *q* akan mudah diterka oleh para *hacker* . Maka untuk membuat nilai *N* sukar untuk dijebol oleh para *hacker* kita perlu nilai faktor *p* dan *q* yang besar. Misalkan, dibandingkan kita memilih nilai 5 dan 11, lebih baik kita pilih bilangan prima yang besar, seperti 673 dan 24971, yang akan menghasilkan nilai *d* = 16805483 dan nilai *e* = 16779840.

Tetapi jika dihitung dengan suatu perangkat lunak ataupun suatu program yang kita buat yang dapat menghitung faktor-faktor dari suatu nilai. Angka- angka di atas dapat dengan mudahnya didapatkan faktor-faktornya. Sehingga hal ini menyimpulkan bahwa kita membutuhkan nilai *p* dan *q* yang sangat besar.

viii. Ancaman yang Mungkin Menyerang RSA

Sistem pengenkripsian RSA mempunyai kemungkinan-kemungkinan kelemahan yang bisa diserang oleh para *eavesdropper* (penyadap, penguping), berikut adalah kelemahan-kelemahan dalam RSA yang sebaiknya dihindari:

1. Nilai *n* terlalu kecil, sehingga mudah untuk difaktorisasi
2. Jumlah nilai eksponen e^n yang terlalu kecil
3. Ukuran kunci yang terlalu kecil, sehingga sandi dapat dijebol dengan *brute force attack*
4. Nilai *d* terlalu kecil
5. Pengunaan nilai modulus yang familiar, hal ini memudahkan para *hacker* untuk menjebol sandi yang ada.

ix. Pertimbangan Teknis dalam Enkripsi RSA

Jika kita berniat untuk mengenkripsi suatu data untuk dikirim ke suatu tujuan, ada beberapa faktor yang sebaiknya diperhatikan agar data yang kita kirim tidak mudah dijebol di tengah jalan.

1. Dalam pembuatan kunci, sebaiknya memilih nilai *p* dan *q* yang jumlahnya tidak saling berdekatan dan besar, karena jika nilai *N* kecil, faktor dari *N* akan sangat mudah didapat.
2. Penggunakan pembangkit bilangan acak yang kuat untuk kunci simetrik yang digunakan, karena *eavesdropper* dapat melakukan *bypass* terhadap RSA dengan menebak kunci simetrik yang digunakan.
3. Sebagaimana halnya *cipher* , bagaimana *public key* RSA didistribusi menjadi hal penting dalam keamanan. Distribusi kunci harus aman dari *man- in-the-middle attack* (penghadang-ditengah-jalan).
4. Memastikan bahwa operasi dekripsi menggunakan waktu yang konstan untuk setiap *ciphertext* yang diproses.
5. Menggunakan *padding scheme* yang relatif terbukti aman seperti *Optimal Asymmetric Encryption Padding*.

III. METODOLOGI PENELITIAN

Tahapan Penelitian

Dalam pengerjaan Tugas Akhir ini diperlukan langkah-langkah kegiatan penelitian untuk mendapatkan hasil yang

maksimal. Untuk itu penulis merencanakan suatu langkah-langkah untuk dapat memaksimalkan dalam pengerjaan Tugas Akhir ini. Langkah-langkah tersebut adalah sebagai berikut :

3.1 Analisis dan Perancangan

3.4.1 Analisis

Masalah yang terjadi pada penjualan makanan dan minuman dirumah makan saat ini masih mengandalkan penjualan secara langsung atau dengan cara menunggu konsumen/pelanggan datang ke tempat rumah makan tersebut hal ini tentu kurang efisien dan peluang untuk mendapatkan keuntungan masih tergolong rendah, belum juga adanya persaingan dari rumah makan lainnya. Di era digital sekarang ini banyak orang tidak mau membuang-buang waktunya dan lebih memilih hal-hal yang efisien meskipun hanya sekedar membeli suatu makanan, karena terkadang orang lebih suka memesan makanan untuk diantar lalu menikmatinya dirumah atau ditempat kerjanya. Hal ini mendorong banyak rumah makan yang memakai jasa kurir untuk mempromosikan dan mengantarkan makanan kepada konsumen, selain jasa kurir berbagai aplikasi-aplikasi pendukung pemesanan secara online pun banyak bermunculan, namun masih banyak aplikasi yang kurang baik dalam segi keamanan data sehingga memungkinkan adanya kecurangan atau pemalsuan data sebuah pesanan oleh orang lain yang dapat merugikan kedua belah pihak. Dengan menambahkan sandi *Algoritma RSA* pada authentifikasi *QR Code* diharapkan dapat mengatasi adanya kecurangan pihak lain dan pemalsuan pesananan yang terjadi pada aplikasi-aplikasi sebelumnya.

3.4.2 Perancangan Sistem

Prosedur yang akan digunakan untuk pelabelan dokumen adalah :

1. Setiap konsumen pemesan diharuskan mempunyai aplikasi delivery order, untuk dapat memilih makanan dan minuman serta rumah makan penyedianya, prosedur pemesanan yaitu konsumen mendownload aplikasi, pilih menu makanan atau minuman yang ingin dipesan, kemudian akan diproses oleh sistem yang nantinya berupa authentifikasi *QR Code* yang terenkripsi.
2. Format data *QR Code* sebelum akan di sandikan oleh *algoritma RSA* berupa, nama pemesan, alamat pemesan, menu pesanan dan kode pesanan.
3. Dalam proses *delivery order*, kurir akan diberikan aplikasi pemindai khusus atau aplikasi scanner khusus yang digunakan untuk menscan kode *QR Code* pemesan.

Pada sistem pelabelan dokumen ini, sandi *algoritma RSA* yang digunakan untuk menyandikan data di dalam *QR Code* adalah sandi *algoritma RSA* dengan 160 karakter.

3.4.3 Flowchart Diagram

Flowchart atau diagram aliran adalah langkah-langkah prosedur sistem yang digambarkan secara grafik.

Flowchart dapat memberi solusi untuk menyelesaikan masalah dalam proses atau algoritma program dalam sistem.

Berikut adalah flowchart sistem yang akan dibuat pada penelitian ini :

1. Flowchart Proses Enkripsi Pembuatan Label menggunakan *Algoritma RSA*
2. Flowchart Scan *QR Code*

Flowchart pada gambar 3.4 adalah flowchart saat melakukan scan qr code pada authentifikasi pesanan konsumen oleh kurir. Generate dari *QR Code* dilakukan dengan menggunakan library Zxing. Pada proses scanning *Qr Code* yang otomatis terhubung dengan database sistem. Jika data yang discan sesuai maka system akan mengidentifikasi identitas Konsumen pemesan. Namun jika data tidak valid maka proses akan langsung terhenti tanpa mengetahui identitas konsumen pemesan

3. Flowchart Algoritma Pembangkit Kunci

Algoritma RSA memiliki dua kunci yang berbeda untuk proses enkripsi dan dekripsi. Dalam menentukan dua bilangan prima sebagai kunci adalah bilangan prima yang besar, karena pemfaktoran bilangan dari dua bilangan prima yang besar sangat sulit, sehingga keamanan pesan lebih terjamin. Pasangan kunci adalah elemen penting dari algoritma RSA.

Berikut ini langkah- langkah dalam membangkitkan dua kunci algoritma RSA.

1. Pilih dua bilangan prima sembarang, p dan q.
2. Hitung $n = p.q$
3. Hitung $\Phi_{(n)} = (p - 1)(q - 1)$.
4. Pilih kunci publik e, yang relatif prima terhadap $\Phi_{(n)}$.
5. Bangkitkan kunci pribadi dengan menggunakan $e.d \equiv 1 \ mod \ \Phi_{(n)}$ atau $e*d \ mod \ \Phi_{(n)} = 1$

Hasil dari algoritma tersebut akan menghasilkan dua kunci, yaitu kunci public (e,n) dan kunci private (d,n).

Cntoh Perhitungan Pembangkit Kunci Algoritma RSA.

Misalkan B akan membangkitkan kunci publik dan kunci pribadi miliknya. B memilih $p = 7$ dan $q = 13$ (keduanya prima). Selanjutnya B menghitung.

$n = 7 \times 13 = 91$

Dan

$\Phi_{(n)} = (7-1)\ (13-1) = 72$

B memilih $e = 5$ karena 5 relatif prima terhadap 72. B mengumumkan nilai e dan n.

Selanjutnya B menghitung nilai dengan algoritma Euclid yang diperluas menjadi

$72 = 14.5 + 2$ $n = 1, a_1 = 5, q_1 = 14$

$5 = 2.2 + 1$ $n = 2, a_2 = 2, q_2 = 2$

$2 = 2.1$

$n = 3, a_3 = 1, q_3 = 2$

$t_2 =$

$t_0 - q_1.t_1 = 0 - 14(10) = -14 = 58$

$t_3 = t_1 - q_2.t_2 = 1 - 2.(-14) = 29.$

Karena $e.d \equiv 1 \bmod \Phi_{(n)}$ dapat ditulis menjadi $e^{-1} \ mod \ \Phi_{(n)} = d$, maka didapat $5^{-1} \ mod \ 72 = 29$. Sehingga diperoleh $d = 29$. Ini adalah kunci pribadi untuk mendekripsikan pesan dan harus dirahasiakan oleh B. Dari perhitungan tersebut didapat kunci publik dan kunci pribadi berturut- turut adalah

Kunci public = (e = 5, n = 91)

Dan

Kunci privat = (d = 29, n = 91)

4. Flowchart Proses Enkripsi Algoritma RSA

Langkah-langkah dalam melakukan proses enkripsi adalah sebagai berikut:

1. Langkah pertama menentukan kunci public penerima pesannya, kunci public disini telah dibuat yaitu nilai e dan n.
2. Kedua Plainteks atau kode pemesanan konsumen dibuat menjadi blok-blok m_1, m_2, m_3, m_4, sehingga block m_i nantinya akan dijadikan pesan chipertext.
3. Ketiga Proses enkripsi menggunakan kunci public (e, n) dari plaintext menjadi chipertext dengan rumus sebagai berikut : $c_i = m_i^e \bmod n$
4. Sehingga langkah terakhir plaintext yang telah dienkripsi menjadi pesan chipertext akan digenerate menjadi *QR Code* yang nantinya akan menjadi kode delivery order konstumer atau pemesan makanan.

Contoh Perhitungan :

Misalkan A akan mengirim pesan ke B. Pesan (Plainteks) yang akan dikirim adalah

m = YAVI

atau dalam sistem desimal pengkodean ASCII adalah

89658673

A memecah m menjadi blok yang lebih kecil, misalkan membagi menjadi 5 blok yang berukuran 2 digit

$m_1 = 89$
$m_2 = 65$
$m_3 = 8$
$m_4 = 73$

Nilai-nilai m_i ini masih terletak di selang [0,91 – 1] agar transformasi menjadi satu-ke-satu. A mengetahui kunci publik B adalah e = 5 dan n = 91. A dapat mengenkripsikan setiap blok plainteks sebagai berikut :

$c_1 = 89^5 \ mod \ 91 = 59$
$c_2 = 65^5 \ mod \ 91 = 39$
$c_3 = 86^5 \ mod \ 91 = 60$
$c_4 = 73^5 \ mod \ 91 = 47$

Jadi chiperteks yang dihasilkan adalah

c = 59396047

5. Flowchart Proses Dekripsi Algoritma RSA

Langkah-langkah dalam melakukan proses dekripsi adalah sebagai berikut:

1. Pertama *QR Code* konsumen pemesan delivery order akan discan oleh aplikasi pemindai khusus atau aplikasi scanning *QR Code.*
2. Kedua aplikasi akan menscan *QR Code* dengan menggunakan nilai dekripsi yang telah ditentukan dengan nilai d dan nilai n.
3. Ketiga *QR Code* yang berisikan chipertext akan discan menggunakan kunci private (*d*) dengan rumus dekripsi sebagai berikut : $m_i = c_i^d \bmod n$.
4. Keempat sehingga setelah discan atau didekripsikan dari kode chipertext akan menjadi nilai plaintext kembali.
5. Langkah terakhir dari nilai plaintext akan dikonversikan menggunakan kode ASCII yang nantinya akan menjadi segala pesanan konsumen itu sendiri.

Contoh Perhitungannya : B akan mendekripsi pesan dengan menggunakan kunci pribadi (d = 29, *n* = 91). Blok – blok chiperteks didekripsikan dengan cara :

$c_1 = 59^{29} \ mod \ 91 = 89$
$c_2 = 53^{29} \ mod \ 91 = 65$
$c_3 = 78^{29} \ mod \ 91 = 86$
$c_4 = 47^{29} \ mod \ 91 = 73$

Akhirnya diperoleh plainteks semula yaitu *m* = 89658673 yang dalam sistem karakter pengkodean ASCII *m* = YAVI.

3.4.4 Use Case Diagram

Use case merupakan gambaran dari sebuah sistem dari sudut pandang pengguna. Diagram use case digunakan untuk menggambarkan fungsi-fungsi dari aspek perilaku sebuah sistem tersebut.

Pada sistem ini aktor dibagi menjadi 4 bagian yaitu konsumen pemesan makanan, admin server, rumah makan penyedia, dan kurir pengantar. User merupakan pengguna pada mobile device yaitu konsumen dan kurir. Sedangkan admin adalah petugas administrator yang memproses setiap pesanan konsumen yang nantinya akan diantarkan oleh kurir atau diproses *delivery order*. Dan terakhir adalah rumah makan sebagai penyedia barang yang dipesan, sehingga ketika rumah makan mendapatkan notifikasi dari admin makan disitu terdapat pesanan dari konsumen. Diagram use case sistem dapat dilihat pada gambar 3.8

Penjelasan dari use diatas adalah sebagai berikut :

1. Terdapat empat aktor yaitu konsumen, admin server, rumah makan dan kurir.
2. Terdapat 8 use case dimana ada : pemesanan *(delivery order)*, pemberian kode authentifikasi berupa *QR Code*, memberikan daftar pesanan, konfirmasi pesanan siap antar , konfirmasi kurir untuk mengantarkan pesanan, mengantarkan pesanan konsumen, scan authentifikasi dan pengambilan tagihan, kemudian simpan ke database.

3.5 Implementasi

Implementasi aplikasi *delivery order* atau pemesanan makanan secara online

nantinya diharapkan dapat memudahkan konsumen dalam pemesanan dan memberikan dampak baik bagi pengusaha rumah makan, dengan memanfaatkan kode batang *QR Code* yang telah dienkripsi sebagai authentifikasinya, sehingga kurir nantinya dalam proses pengiriman barangnya hanya diberikan aplikasi pemindai atau aplikasi scanner *QR Code* khusus untuk menscanning authentifikasi setiap pemesan. Sehingga diharapkan dapat mengurangi pemalsuan pesanan yang mengakibatkan kerugian kedua belah pihak. Pada implementasinya, untuk mengolah gambar penanda *QR Code* digunakan sebuah library open source bernama *Zxing.* Sistem generator *QR Code* dan data dienkripsi menggunakan open source Php (*php hypertext preprocessor*), dan *Smartphone Android* dengan bahasa pemrogaman java.

Pada sistem pelabelan dokumen dibutuhan beberapa aplikasi pendukung seperti : *Android Studio, Qr Code generator, sublime text.* Pada tahap uji coba akan diimplementasikan pada authentifikasi *delivery order* konsumen yang sudah memesan makanan kepada admin/server. Data yang akan dienkripsi untuk menjadi authentifikasi berupa kode *QR Code* adalah Nama pemesan, Lokasi/posisi pemesan, Menu barang pesanan, dan Kode pesanan. Dengan adanya aplikasi *delivery order* makanan menggunakan authentifikasi berupa *QR Code* yang telah terenkripsi didaerah kabupaten jember diharapkan dapat memudah konsumen dalam memesan dan menikmati makanan diseluruh rumah makan dan tentunya diharapkan dapat menambah keuntungan dari segi usaha yang dilakukan rumah makan itu sendiri, dikarenakan prosesnya yang menggunakan penyandian khusus untuk menjaga keamanan dalam pemesanan dan menjaga kepercayaan para pelanggannya.

3.6 Pengujian

3.6.1 Proses pembuatan isi label *QR Code* pesanan konsumen yang akan dienkripsi menggunakan *algoritma RSA.*

Proses ini menjelaskan pembuatan label dan isi dari *QR Code* yang nantinya akan dienkripsi menggunakan *algoritma RSA*, adapun isi dari *QR Code* disini yaitu nama pemesan, alamat pemesan, menu pesanan dan kode id pemesan. Adapun cara kerja pembuatan label *QR Code* seperti dibawah ini :

1. Nama pemesan : YAVI
2. Alamat Pemesan : KARIMATA
3. Menu Pesanan : 1 Buah Mie Kober Pedas Level 10 pada menu mie kober disini diberi kode “T1FA1H”

T1 : Menunjukkan Tempat Rumah Makan (Mie Kober Karimata)

FA : Jenis Makanan (Mie Kober)

1 : Jumlah Pesanan (Satu Bungkus)

H : Level Makanan (High)

4. Kode id Pemesan : “1J001”

1J : Lokasi Pemesan Makanan (Karimata)

001 : Id Nomor Urut Order

Sehingga didapatkan beberapa data yaitu nama pemesan "YAVI", alamat pemesan "JEMBER", menu pesanan "T1FA1H", dan kode id pemesanan "1J001". Yang nantinya data berupa nama, alamat pemesan, menu pesanan, dan kode id pemesanan yang akan dienkripsi menggunakan sandi *algoritma RSA.*

3.6.2 Proses generate kode batang *QR Code* dan proses enkripsi menggunakan sandi *algoritma RSA.*

Pada proses pengimputan diatas sebelumnya didapatkan data untuk proses pemesanan makanan siap antar (*delivery order)* yaitu nama pemesan "YAVI", alamat pemesan "KARIMATA", menu pesanan "T1FA1H" dan kode id pemesanan "1J001". Dari data yang diperoleh sebelumnya disini jika belum dienkripsi menggunakan sandi *algoritma RSA* maka bentuk kode batang *QR Code* nya seperti gambar dibawah ini

Gambar diatas ialah kode batang *QR Code* pesanan konsumen yang belum dienkripsi menggunakan *algoritma RSA*, sehingga untuk menjaga keamanan data dari setiap konsumen pemesan makanan siap antar (*delivery order)* maka dari data yang diperoleh sebelumnya ialah nama pemesan "YAVI", alamat pemesan "KARIMATA", menu pesanan "T1FA1H", dan kode id pemesan "1J001". Data yang dienkripsi disini ialah YAVI, KARIMATA, T1FA1H, dan 1J001. Sehingga data akan dikonversikan kedalam ASCII terlebih dahulu adapun prosesnya seperti pada tabel 3.1 berikut ini :

Karakter	Nilai ANSI ASCII
YAVI	89658673
KARIMATA	7565827377658465
T1FA1H	844970654972
1J001	4974484849

Tabel 3.2 Proses konversi karakter kedalam ASCII

Data diatas adalah data pemesan yang telah dikonfersikan kedalam ASCII yang nantinya data tersebut akan dienkripsi menggunakan sandi *algoritma RSA,* dengan contoh perhitungan enkripsinya, namun sebelum menghitung enkripsi terlebih dahulu kita bangkitkan kunci public dan kunci privatnya terlebih dahulu, seperti contoh dibawah ini :

Kunci public :

p = 7 n = p*q

$\Phi_{(n)}$ = (p-1)(p-1)

q = 13 n = 7*13

$\Phi_{(n)}$ = (7-1)(13-1)

e = 5 n = 91

$\Phi_{(n)}$ = 72

Pada kunci public diperoleh nilai p = 7, q = 13 dan e = 5 dan untuk kunci sandi *algoritma RSA* harus menggunakan bilangan prima. Sehingga kunci bilangan privatnya atau kunci d nya adalah $e.d \equiv 1 \bmod \Phi_{(n)}$ atau dapat dituliskan e^{-1} *mod* $\Phi_{(n)} = d$ jadi kunci privatnya atau nilai d adalah 5^{-1} *mod* 72 = 29. Yang dapat diketahui *d = 29.* Adapun perhitungan untuk proses enkripsinya seperti dibawah ini :

Rumus Enkripsi yaitu $c_i = m_i^e \bmod n$

1. YAVI = 89658673
   a. Y = 89^5 mod 91 = 59
   b. A = 65^5 mod 91 = 39
   c. V = 86^5 mod 91 = 60
   d. I = 73^5 mod 91 = 47

2. KARIMATA = 7565827377658465
   a. K = 75^5 mod 91 = 17
   b. A = 65^5 mod 91 = 39
   c. R = 82^5 mod 91 = 10
   d. I = 73^5 mod 91 = 47
   e. M = 77^5 mod 91 = 77
   f. A = 65^5 mod 91 = 39
   g. T = 84^5 mod 91 = 28
   h. A = 65^5 mod 91 = 39

3. T1FA1H = 844970654972
   a. T = 84^5 mod 91 = 28
   b. 1 = 49^5 mod 91 = 56
   c. F = 70^5 mod 91 = 70
   d. A = 65^5 mod 91 = 39
   e. 1 = 49^5 mod 91 = 56
   f. H = 72^5 mod 91 = 11

4. 1J001 = 4974484849
   a. 1 = 49^5 mod 91 = 56
   b. J = 74^5 mod 91 = 16
   c. 0 = 48^5 mod 91 = 55
   d. 0 = 48^5 mod 91 = 55
   e. 1 = 49^5 mod 91 = 56

Pada perhitungan diatas adalah perhitungan untuk proses enkripsi dari nama pemesan dan lokasi pemesan perhitungan tersebut dilakukan sampai proses enkripsi menu pesanan konsumen dan kode id konsumen sehingga didapatkan kode enkripsi sebagai pada tabel 3.2 dibawah ini :

Karakter	Nilai ANSI ASCII	Enkripsi
YAVI	89658673	59396047
KARIMATA	7565827377658465	1739104777392839
T1FA1H	844970654972	285670395611
1J001	4974484849	5616555556

Tabel 3.3 Proses Enkripsi Isi Kode Batang *QR Code*

Ketika proses enkripsi mulai dari nama, lokasi, menu pesanan dan kode id pesanan telah selesai maka proses selanjutnya adalah menggenerate

semua kode hash atau kode enkripsi tersebut ke dalam kode batang *QR Code* konsumen atau pemesan makanan siap antar (*delivery order)* karakter yang dipakai pada proses enkripsi disini dibatasi dengan jumlah maksimum karakter yaitu 50 karakter, adapun pada contoh karakter diatas berjumlah 23 karakter sehingga didapatkan kode batang *QR Code* seperti gambar dibawah ini :

Adapun diatas adalah kode batang *QR Code* yang telah terenkripsi, sehingga dalam proses scanning data nantinya dibutuhkan aplikasi scanning atau pemindai *QR Code* khusus yang telah disetting atau diberi sandi *algoritma RSA* agar dapat mengetahui isi dari setiap pemesanan makanan dari setiap konsumen. Yang nantinya aplikasi ini diharapkan dapat mempermudah dalam melakukan proses pemesanan maupun kurir yang mengantarkan makanan (*delivery order*) dan mengurangi adanya pemalsuan, kecurangan dan salah antar kepada konsumen.

3.7 Analisa Hasil

3.7.1 Analisa hasil dan pengujian Scanning Kode Batang *QR Code* Konsumen *delivery order*

Pada pengujian scanning ini dilakukan dengan menggunakan aplikasi scanner khusus yang telah diberikan sandi *algoritma RSA* sehingga jika kode batang *QR Code* yang telah terenkripsi discan akan dapat diketahui isi keaslian dari data tersebut karena dalam proses ini aplikasi scanner akan mendekripsikan kode hash atau kode enkripsi yang ada pada kode batang *QR Code* tersebut dengan contoh jika kode batang *QR Code* discan dengan tidak menggunakan aplikasi pemindai atau aplikasi scanner khusus akan tampil seperti gambar dibawah ini :

Seperti contoh gambar diatas jika aplikasi scanning tidak menggunakan sandi *algoritma RSA* maka akan memunculkan kode hash atau kode enkripsi dari nama, lokasi, menu pesanan dan id pesanan, sehingga hanya memunculkan angka-angka enkripsi seperti contoh gambar diatas, namun jika menggunakan aplikasi pemindai khusus yang telah diberikan sandi *algoritma RSA* nantinya maka scanning kode batang *QR Code* akan mendekripsikan dari setiap kode hash atau kode enkripsi yang ada pada batang *QR Code* itu sendiri sehingga dapat diketahui keasliannya seperti contoh gambar dibawah ini :

Pada gambar scanning diatas adalah contoh aplikasi scanning kode batang *QR Code* yang telah menggunakan sandi *algoritma RSA* sehingga kode batang yang awalnya terenkripsi akan didekripsikan oleh aplikasi scanning khusus untuk dapat mengetahui nama, lokasi, menu pesanan dan id konsumen.

Yang nantinya diharapkan dalam pengaplikasiannya tidak akan terdapat lagi pemalsuan ataupun kecurangan konsumen dalam proses *delivery order* dikarenakan setiap konsumen yang memesan akan mendapatkan kode batang *QR Code* yang tidak sama dan telah dienkripsi oleh sandi *algoritma RSA* tersebut, yang memungkin bagi konsumen/user tidak mendapatkan kerugian karena

kesalahan pengiriman dan bagi pihak rumah makan/penyedia jasa makanan tidak akan mendapatkan kerugian karena adanya kesalahan ataupun pemalsuan pengiriman.

IV. Pembahasan

Pada bab ini akan dijelaskan tentang proses pengimplementasian integrasi sistem antara Delivery Order (*Pemesanan Makanan Secara Online)* yang berlabel *QR Code* yang telah terenkripsi *algortima RSA* dengan aplikasi scan *QR Code algoritma RSA* berbasis androi, sesuai perancangan sisetem yang telah dibahas pada bab 3 serta melakukan pengujian sistem yang telah dibangun.

4.1 Proses Sandi *Algoritma RSA*

Pada proses sandi *Algoritma RSA* ini, akan dilakukan beberapa ujicoba enkripsi pada nama pemesan, alamat pemesan, menu pesanan, kode pesanan yang nantinya akan dienkripsi. Adapun prosesnya seperti gambar 4.1 dibawah ini :

Penjelasan pada proses enkripsi *algoritma RSA* ini dilakukan oleh sistem, dimana nantinya konsumen atau pihak pemesan melalui delivery order yang telah melakukan login dan memesan makanan dengan mengisikan sesuai format yang ditentukan, lalu pesanan konsumen akan diproses oleh sistem, sehingga jika pesanan konsumen tersedia maka konsumen/pemesan (*delivery order*) akan mendapatkan authentifikasi berupa *QR Code* yang telah terenkripsi oleh sistem

Pada proses enkripsi, pesan yang dienkripsi adalah id pembeli, tanggal pembeli &nama driver. Pada proses enkripsi kunci yang digunakan adalah kunci public dan kunci privat, dengan kunci public yaitu p = 7, q = 13, n = 91, $\Phi_{(n)}$= 72, e = 5 dan kunci privatnya adalah d = 29, maka proses enkripsinya dengan cara id pembeli, tanggal pembeli dan nama driver dijadikan atau dikonversikan kedalam kode ASCII lalu kemudian dienkripsi menggunakan rumus $c_i = m_i^e \bmod n$sehingga rumus perhitungannya seperti dibawah ini :

1. Id pembeli disini bersifat automatis dimisalkan id pembeli yaitu 12.
2. Tanggal pembeli : 12-10-2018
3. Nama driver : doni

Konversi kode ASCII dari id pembeli, tanggal pembeli dan nama driver seperti dibawah ini :

1. IdPembeli “12” Konversi kode ASCIInya adalah 49,50
2. Tanggal Pembeli“12-10-20” Konversi Kode ASCIInya adalah49,50,45,49,48,45,50,48,49, 56
3. Nama Driver “doni” Konversi kode ASCIInya adalah 100,111,110,105.

Jika id pembeli, tanggal pembeli dan nama driver telah dikonversikan kedalam kode ASCII maka proses enkripsi dapat dilakukan, artinya dalam proses enkripsi kode harus dikonversikan kedalam bentuk kode ASCII terlebih dahulu, rumus yang digunakan dalam proses enkripsi adalah $c_i = m_i^e \bmod n$sehingga didapatkan kode chipertext atau kode yang telah dienkripsi dari kode ASCII “id pembeli, tanggal pembeli dan nama driver” seperti berikut : “142,74,267,379,74,257,56,70,74,257,70,257,379,267,152,71,384,365”kemudian dari chipertext tersebut akan langsung digenerate menjadi kode batang *QR Code* yang nantinya itu menjadi kode pemesanan untuk konsumen yang akan discan oleh

kurir atau driver pengantar pesanan menggunakan aplikasi scanner khusus.

Dari percobaan diatas dapat disimpulkan bahwadata kode batang Qr Code pemesanan delivery order konsumen adalah benar sehingga aplikasi scaning tes berhasil, namun jika data kode batang Qr Code palsu atau dimanipulasi maka sistem tidak akan bisa memproses sehingga status pembayaran konsumen delivery order tidak akan berganti terbayar, pada proses scanning kurir atau driver pengantar pesanan aplikasi scanning yang di gunakan akan otomatis melakukan dekripsi kode batang *QR Code* konsumen delivery order sehingga data konsumen berupa *QR Code* tidak akan bisa mudah di manipulasi.

4.2 Proses Dekripsi *Algoritma RSA*

Pada proses dekripsi algoritma RSA disini, data yang telah dienkripsi berupa kode hash delivery order berbentuk *QR Code*. Selanjutnya akan didekripsi kembali dengan menggunakan aplikasi scanner khusus yang telah terintegrasi dengan sistem sehingga nantinya akan muncul kode awal dan pesanan dapat diproses artinya pesanan telah terbayar, seperti pada gambar 4.3 berikut ini :Penjelasan pada proses deskripsi ini, dilakukan oleh sistem. Dimana nantinya konsumen yang telah memesan (*delivery order*) akan mendapatkan authentifikasi berupa kode *QR Code* yang telah terenkripsi, kemudian akan discan menggunakan aplikasi android scanner khusus. Sehingga proses deskripsi ini dimulai dengan menscan authentifikasi pesanan konsumen berupa kode *QR Code* yang output dari hasil scan tersebut adalah tanda validasi bahwa makanan benar telah terbayar dan isi kod *QR Code* akan dapat dilihat pada menu history karena telah didekripsikan.

4.3 Contoh Kasus Kecurangan Manipulasi *QR Code*

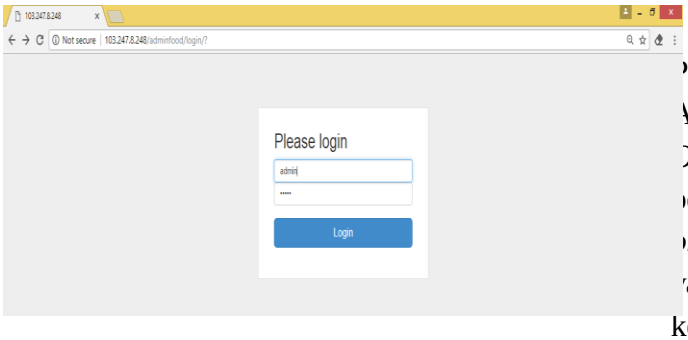
Contoh kasus yang dimanipulasi oleh orang lain. Misalkan ada authentifikasi berupa *QR Code* pesanan makanan dengan data *QR Code* palsu. Maka jika terdapat masalah pada kode *QR Code* yang bukan dari pihak aplikasi *delivery order*, sistem akan dapat mengetahui sehingga, misalkan data *QR Code* yang dibuat itu palsu ketika dilakukan proses scanning menggunakan *Algoritma RSA* output yang dikeluarkan akan berbeda pada perangkat android.

Dan apabila data tersebut tidak terenkripsi, maka hasil output data yang akan dikeluarkan tidak berupa validasi data yang telah terbayar atau berupa data tidak terbayar dan sistem tidak akan mengetahui transaksi apa pada saat kita melihatnya dihistory, contoh gambar seperti dibawah ini :

Gambar 4.6 Data Dekripsi Dengan Data Palsu/Manipulasi

4.4 Proses Pengimputan Data, Pemesanan &*Delivery Order* Pada Website

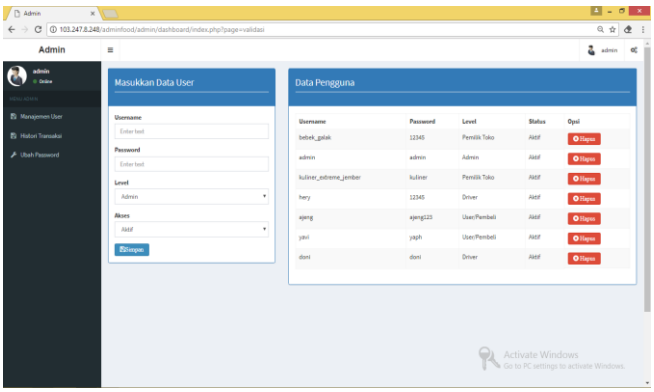
Pada proses desain login, admin akan memasukkan user dan password yang telah ada pada sistem atau website Delivery Order, dapat dilihat pada gambar 4.7 berikut ini :



Gambar 4.7Proses Login Admin Website

Pada desain sistem login admin, admin memasukkan user dan password yang telah ada pada sistem. Dan kemudian melanjutkan proses penginputan data setting pertama bisa dilihat pada gambar 4.8.

Pada web admin pengimputan data diatas pada menu management user digunakan untuk membuat atau membuka akun baru seperti pelapak (rumah makan), driver ataupun user (pemesan) namun user dapat membuat akun sendiri menggunakan aplikasi android *delivery order* pada ponselnya masing-masing maka akan muncul data yang telah tersimpan pada menu management user seperti gambar 4.9 dibawah ini :

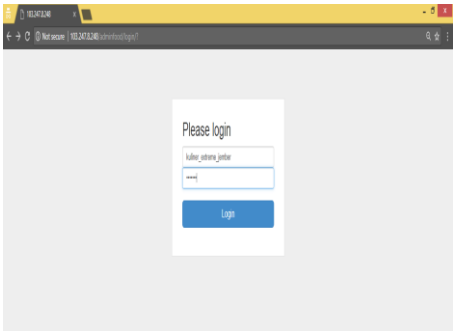


Gambar 4.9Data Pengguna Yang Tersimpan

Pada data pengguna diatas berisikan emilik lapak atau Pemilik toko, Driver, dmin, dan User Pembeli selaku (*Delivery rder*) sehingga nanti dalam proses emesanan makanan atau proses *delivery rder* dapat diketahui rumah makan, driver ang akan melakukan proses transaksi, kemudian proses transaksi tersebut akan tersimpan dalam menu Histroy Transaksi seperti gambar 4.10 dibawah ini :

Setelah admin melakukan perjanjian kerjasama maka pelapak atau rumah makan penyedia pesanan dapat melakukan pemesaran melalui akun yang telah dibuat melalui admin pada web *delivery order*, seperti gambar dibawah ini, gambar dibawah ini adalah contoh login pelapak atau rumah makan :

Pada menu login pelapak user dan password yang diinputkan adalah user dan password yang telah dibuat oleh admin dan telah melakukan kerjasama dengan admin, sehingga admin dapat melakukan pemasaran makanannya atau menu apa yang disediakan, seperti pada gambar4.12 berikutini:



Gambar 4.12Web Pemilik Toko

Pada gambar 4.12 diatas terdapat beberapa menu yaitu menu ***master lokasi*** dimana menu ini untuk menentukan letak lokasi dari pelapak atau rumah makan yang menjalin kerja sama dengan admin atau dengan web *delivery order* tersebut, yang kedua adalah menu ***master makanan*** menu ini memuat segala macam barang dagangan pelapak atau rumah makan sehingga menu ini berisikan nama produk, harga, detail produk beserta foto produk, produk disini yaitu berupa makanan siap saji yang akan di *delivery oder*, yang nantinya semua transaksi tersebut akan dimuat pada menu transaksi, jika terjadi transaksi maka sistem akan memberikan notifikasi kepada pelapak atau rumah makan yang nantinya rumah makan akan segara memproses transaksi tersebut dengan menyediakan makanan pesanan konsumen (*delivery order*), ketika makanan telah siap pelapak akan memberikan status proses, kemudian secara otomatis sistem akan memberikan authentifikasi kepada driver, adapun proses langkah-langkah pemesanan pada aplikasi delivery order di android smartphone adalah sebagai berikut :

Langkah Pertama adalah download aplikasi *delivery order* diibaratkan aplikasi berada diplaystore ketika telah terdownload makan install aplikasi lalu jalankan aplikasinya, sehingga nantinya tampilan login usernya akan seperti berikut :

Gambar 4.13 Desain Tambilan LoginUser Android

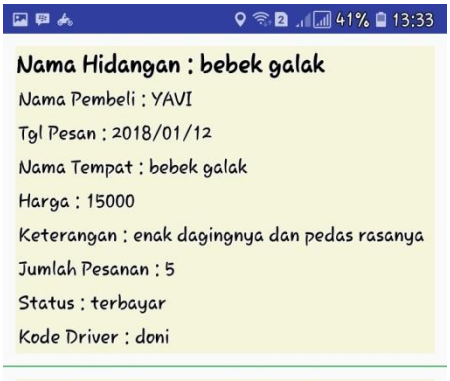
Gambar diatas adalah tampilan awal atau login dari aplikasi *Delivery Order* dismarthphone, aplikasi *delivery order* memiliki beberapa menu yaitu menu *Daftar hidangan*, *History pemesanan* dan *Keluar.*Yang pertama ialah menu daftar hidangan, menu ini digunakan untuk melihat hidangan yang tersedia dirumah makan yang telah sekaligus memilih hidangan yang ingin dipesan atau ingin di *delivery order* tentunya hidangan-hidangan tersebut sesuai dengan rumah makan yang telah menjalin kerjasama dengan pihak aplikasi. Kedua history pemesanan, pada menu history pemesanan inilah transaksi jual beli akan dicatat dan diketahui proses statusnya seperti contoh makanan masih dalam proses pemasakan atau makanan dalam proses pengiriman kurir yang nantinya menu ini akan dapat mencetak kode *QR Code* sesuai nama, alamat, kode, dan menu pemesanyang tentunya juga kode tersebut nantinya akan dienkripsi dengan Algoritma RSAsehingga data pemesanan akan terjaga keamannnya adapun contoh menu tampilan *Delivery Order* user diandroid sebagai berikut :

Sehingga Langkah Kedua yang kita lakukan jika ingin memesan makan menggunakan aplikasi tersebut ialah dengan memilih menu Daftar Hidangan adapun gambar jelasnya seperti dibawah ini :

Gambar 4.15 Tampilan Menu Pemesanan di Android

Pada tampilan diatas itu adalah tampilan menu hidangan terdapat dua pilihan yaitu menu hidangan es sunder bolong dan menu hidangan bebek galak,sehingga jika konsumen ingin membeli atau memesan secara *delivery order* maka konsumen tinggal klik menu yang ingin dipesan seperti contoh pada gambar sampingnya ialah konsumen memesan menu bebek galak dengan jumlah 5 bungku, lalu tinggal pilih menu simpan, maka secara otomatis pesanan

Pada proses akhir yaitu ketika kode batang QR Code yang telah terenkripsi tersebut discan oleh kurir jika kode tersebut asli maka status transaksi akan berubah menjadi "terbayar" dan proses transaksi selesai, namun apabila dalam proses scanning kode batang QR Code bukan yang asli maka status transaksi tidak akan berubah menjadi terbayar sehingga status transaksinya tetap yaitu proses, disini perbedaan antara sistem dilevery order yang menggunakan keamanan data yang memiliki kelebihan agar data atau pesanan tidak dengan mudah terjadi pemalsuan order atau order fiktif, adapun contoh transaksi delivery order yang benar dan sampai proses transaksi selesai, seperti gambar dibawah ini :



Gambar 4.22 Bukti Data Konsumen Asli dan Transaksi Selesai

Pada proses akhir yaitu ketika kode batang QR Code yang telah terenkripsi tersebut discan oleh kurir jika kode tersebut asli maka status transaksi akan berubah menjadi "terbayar" dan proses transaksi selesai, namun apabila dalam proses scanning kode batang QR Code bukan yang asli maka status transaksi tidak akan berubah menjadi terbayar sehingga status transaksinya tetap yaitu proses, disini perbedaan antara sistem dilevery order yang menggunakan keamanan data yang memiliki kelebihan agar data atau pesanan tidak dengan mudah terjadi pemalsuan order atau order fiktif, adapun contoh transaksi delivery order yang

4.5 Perbandingan Sistem

Integrasi pada sistem pemesanan makanan (*delivery order)* dikembangkan untuk mempermudah pemesanan makanan dan memberikan solusi cepat pada proses *delivery order* dikarenakan makanan dapat di cek ada atau tidaknya. Data pengembangan sistem dan perbedaan sistem lama dengan sistem baru dapat dilihat pada tabel 4.1

4.6 Pengujian

A. Pengujian Black Box

Pengujian black box merupakan metode pengujian yang berfokus pada kebutuhan fungsional dari aplikasi.pengujian black box dilakukan dengan focus pada hasil keluaran yang diharapkan

dari sistem yang diuji, apakah dapat berjalan sesuai yang diharapkan atau tidak. Tabel pengujian black box dapat dilihat pada tabel 4.2.

V. ESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan uraian permasalahan dan pembahasan pada bab sebelumnya mengenai *"Pengembangan Aplikasi Delivery Order Menggunakan Algoritma RSA Pada Authentifikasi QR Code Berbasis Android",* Maka dapat diambil kesimpulan bahwa :

1. Implementasi *QR Code* yang telah terenkripsi menggunakan *Algoritma RSA* pada authentifikasi pemesanan makanan (*delivery order*) dapat mempermudah transaksi dan dapat menjaga keamanan data.
2. Sistem atau Aplikasi yang digunakan dapat mempermudah dan mempercepat transaksi dikarenan sistem yang digunakan juga terhubung dengan pelapak atau penyedia makanan (rumah makan)
3. Sistem atau Aplikasi yang dikembangkan dapat menguji adanya data yang dimisalkan palsu dan dapat mengetahui proses transaksi sehingga dapat mengurangi driver terlanjur datang ke tempat namun tempat makan tidak menyediakan makanan yang dipesan.

5.2 Saran

Pada penelitian ini masih sangat jauh dari nilai sempurna, penulis menyarankan bagi peneliti selanjutnya agar :

1. Peneliti selanjutnya diharapkan menggunakan penyandian *QR Code* dengan metode atau algoritma terbaru.
2. Sistem ini masih memiliki kekurangan pada akurasi GPS yang masih rendah, sehingga disarankan untuk peneliti selanjutnya melakukan pengembangan sistem dengan pengambilan titik koordinat menggunakan alat bantu berupa GPS *Tracker.*
3. Pada proses pengambilan titik koordinat pengembang diharapkan dapat lebih mendetailkan setiap garis titik koordinat yang diambil pada lokasi tanah pemohon.

DAFTAR PUSTAKA

Devha, Canda P. 2013. *PengamananPesanRahasiaMenggunakanAlgoritmaKriptografiRivest Shank Adleman (RSA),* UniversitasPendidikan Indonesia.

Asmono. 2013.*PemanfaatKomponen Layer DalamSistemOperasi Android,* Universitas Indonesia.

Ariadi.(2011).*AnalisisdanPerancanganKodeMatriksDuaDimensiQuickResponse(QR)Code.*Skripsi.UniversitasSumateraUtara.

Ashford, Robin. 2010. *QR Code and academic libraries reaching mobile users.(Online)*http://crln.acrl.org/content/71/10/526.full

Bruen,Aiden.A&Forcinito,Mario.A. 2011.*Cryptography,InformationTheory,andError-Correction: AHandbook forthe21stCentury*. JohnWiley&Sons :Canada.

EdyWinarno.2009.*PenggunaanXML Database XindicepadaAplikasiKriptografimenggunakan Data XML untukKeamananDistribusi Data*, JurnalTeknologiInformasi DINAMIK Volume XIV, No.2.

Bunafit Nugroho. 2004. *PHP dan MySQL dengan editor Dreamweaver MX*.ANDIYogyakarta, Yogyakarta.

Denso Wave Incorporated. 2013. Answers to your question about the QR Code. (Online) http://www.qrcode.com/en/

Denso ADC.2011. *QR Code Essentials*.http://www.nacs.org /LinkClick.aspx?fileticket=D1FpVAvvJuo%3D&tabid=1426&mid=4802.

Munir,Rinaldi.2006.*Kriptografi,Algoritma RSA &ElGamal*.DapartemenTeknikInformatika, ITB:Bandung.

Savitri, Ayunda W. 2013. *Android kian mengepakkan sayapnya di Indonesia*.Sumber :Okezone.com.

JurnalSarjanaTeknikInformatika e-ISSN: 2338-5197 Volume 1 Nomor 1, *Pemanfaatan Google Maps Api Untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web*.Juni 2013.

Wildan, Habibi. 2011. *Undergraduate Thesis Google Maps*. ITS : Surabaya.

Yuhana, LailiUmi. 2010.*Pemanfaatan Google Maps UntukPemetaan&Pencarian Data PerguruanTinggiNegeri Di Indonesi*, ITS : Surabaya.

Abdul, Kadir. 2003. *PengenalanSistemInformasi*. ANDI Yogyakarta, Yogyakarta.

RifkiSadikin. 2012. *KriptografiuntukKeamananJaringan*. EdisiPertama. Penerbit ANDI,Yogyakarta.

Shodiq, Amri. 2008. *Pemrograman Google Maps API*.ITB : Bandung