Pengembangan Sistem Keamanan untuk E-Commerce

I Gusti Ngurah Indra Saputra, Gusti Made Arya Sasmita, A. A. K. Agung Cahyawan W.
Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Udayana
Bukit Jimbaran, Bali, Indonesia
michael_bryan_97@yahoo.com, aryasasmita@it.unud.ac.id, agung.cahyawan@unud.ac.id

Abstrak

E-commerce adalah kegiatan transaksi barang atau jasa secara jarak jauh antara dua buah perusahaan (business to business) atau antara perusahaan dengan pelanggan (business to consumer). E-commerce telah mempermudah transaksi secara jarak jauh, namun ada resiko keamanan ketika melakukan transaksi pada e-commerce. Sistem keamanan e-commerce secara umum menggunakan bantuan protokol keamanan lainnya seperti SSL (Secure Socket Layer) sehingga sangat bergantung dengan protokol keamanan tersebut. Masalah sistem keamanan e-commerce yang ada diatasi dengan pengembangan sistem keamanan yang menggunakan algoritma keamanan langsung di dalam halaman web. Sistem dirancang mengamankan data transaksi menggunakan algoritma RC6, kunci enkripsi RC6 diamankan dengan algoritma RSA, dan data hasil enkripsi RC6 di-encode dengan Base64. Sistem keamanan yang dihasilkan memblokir data transaksi saat pengguna mengklik tombol submit transaksi, kemudian data transaksi dienkripsi menggunakan algoritma yang diterapkan sebelum dikirim ke server sehingga data transaksi sudah aman tanpa bantuan protokol keamanan lainnya misalnya SSL (Secure Socket Layer).

Kata kunci: *e-commerce*, sistem keamanan, *RC6*, *Base64, RSA*

Abstract

E-commerce is a remote goods or services transaction activity between two company (business to business) or between company with customers (business to consumer). E-commerce was simplifying the transaction process between producers and consumers, but there is a risk of security issue if doing transactions on e-commerce. A common e-commerce security system using another protocol security e.g. SSL (Secure Socket Layer) so these e-commerce security system is very dependent with these security protocol. The e-commerce security problem was fixed by developed security system that using security system directly on the web page. The system is designed to securing transaction data using RC6 encryption, RC6 encryption key is secured by RSA encryption, and encrypted data is encoded by Base64. The resulting security system is blocking transaction data when users clicking submit transaction button, then these transaction data encrypted using applied algorithm before send it into server so transaction data was secured without aid of another security protocol e.g. SSL (Secure Socket Layer).

Keyword: *e-commerce*, security system, *RC6*, *Base64*, *RSA*

1. Pendahuluan

Para pelaku bisnis telah memanfaatkan kegiatan transaksi barang atau jasa secara jarak jauh tanpa harus bertatapan muka secara langsung dengan para pelaku bisnis atau para konsumen lainnya. Teknologi yang memungkinkan untuk melakukan kegiatan transaksi barang dan jasa tersebut dikenal dengan istilah *E-commerce*. *E-commerce* adalah teknologi *Internet* yang berfokus pada kegiatan transaksi barang atau jasa secara jarak jauh tanpa harus bertemu secara langsung. Para pelaku bisnis menggunakan teknologi *e-commerce* karena teknologi *e-commerce* telah meningkatkan reliabilitas pelaku bisnis dalam kegiatan transaksi, tidak hanya karena kemudahannya, tetapi juga karena teknologi *e-commerce* sudah mulai terjangkau oleh semua pengguna *Internet* sehingga memungkinkan para pelaku bisnis untuk memperluas jaringan pemasarannya. Sistem *e-commerce* harus mengikuti aturan dari beberapa

infrastruktur, yaitu infrastruktur sistem distribusi barang (*flow of good*), infrastruktur sistem pembayaran (*flow of money*), dan infrastruktur sistem informasi yang diterapkan (*flow of information*) untuk menjaga reliabilitas dalam melakukan transaksi barang atau jasa secara jarak jauh [1]. Aturan infrastruktur tersebut mencakup semua kebutuhan dari sistem *e-commerce*, terutama untuk menjaga kepercayaan dari konsumen. Salah satu cara untuk memenuhi kelancaran dari pemenuhan infrastruktur tersebut yaitu dengan cara memperhatikan aspek keamanan (*security*) yang harus diberikan kepada para pelanggan.

Sistem keamanan *e-commerce* adalah sistem yang berfungsi untuk menjaga keamanan dan kenyamanan dalam proses transaksi pada *e-commerce*. Sistem keamanan pada sistem *e-commerce* umumnya mengamankan sistem *e-commerce* dengan bantuan dari protokol keamanan eksternal yang harus di-*install* ke dalam *web server* sehingga membuat keamanan dari sistem *e-commerce* tersebut sangat bergantung dari protokol keamanannya. Kekurangan dari sistem keamanan *e-commerce* yang ada mendasari pengembangan sistem keamanan *e-commerce* pada penelitian ini. Pengembangan sistem keamanan pada *e-commerce* yang dirancang mengamankan data transaksi langsung melalui *script* pada halaman *web* dan tanpa menggunakan protokol keamanan manapun. Sistem keamanan yang dikembangkan menggunakan sistem keamanan berbasis *script* algoritma yaitu *encoding* berbasis *Base64*, algoritma enkripsi *RC6* dan algoritma enkripsi *RSA*. *Script* dari algoritma keamanan tersebut mengamankan proses transaksi dalam *e-commerce* langsung dari dalam halaman *web* sehingga data transaksi telah diamankan tanpa menggunakan protokol keamanan manapun.

Penelitian dalam pengembangan sistem keamanan untuk *e-commerce* menggunakan beberapa *state of the art* yang menjadi acuan dan perbandingan yaitu penelitian dari Mateus Mas Belalawe yang membahas penelitian tentang studi kasus pada sistem keamanan *e-commerce* terhadap situs www.buahonline.com [2], Andre M. R. Wajong dan Carolina Rizki Putri yang membahas penelitian tentang keamanan *e-commerce* [3], dan Samsul Huda dkk. membahas penelitian tentang implementasi keamanan *e-commerce* yang menggunakan *schnorr digital signature* [4]. Ketiga penelitian tersebut menggunakan sistem keamanan *e-commerce* yang mengandalkan protokol keamanan eksternal yaitu *SSL* (*Secure Socket Layer*), *TLS* (*Transport Layer Security*), dan *PGP* (*Pretty Good Privacy*). Protokol keamanan eksternal tersebut membutuhkan proses instalasi pada sistem *server*-nya sehingga sistem keamanan dari sistem *e-commerce* yang digunakan sangat bergantung pada protokol keamanan yang terpasang. Sistem keamanan *e-commerce* yang dikembangkan pada penelitian ini diharapkan mampu mengamankan data transaksi tanpa protokol keamanan yang lainnya seperti *SSL*.

2. Metodologi Penelitian

Metodologi penelitian membahas metode-metode yang dilakukan untuk melakukan penelitian dalam pengembangan sistem keamanan untuk *e-commerce*. Pengembangan sistem keamanan untuk *e-commerce* menggunakan beberapa metode dalam penelitian ini yaitu tahap pengerjaan dan penjelasan diagram arsitektur sistem keamanan. Subbab berikut memuat metodologi penelitian yang digunakan untuk melakukan penelitian sistem keamanan yang dilengkapi gambar dan tabel pendukung.

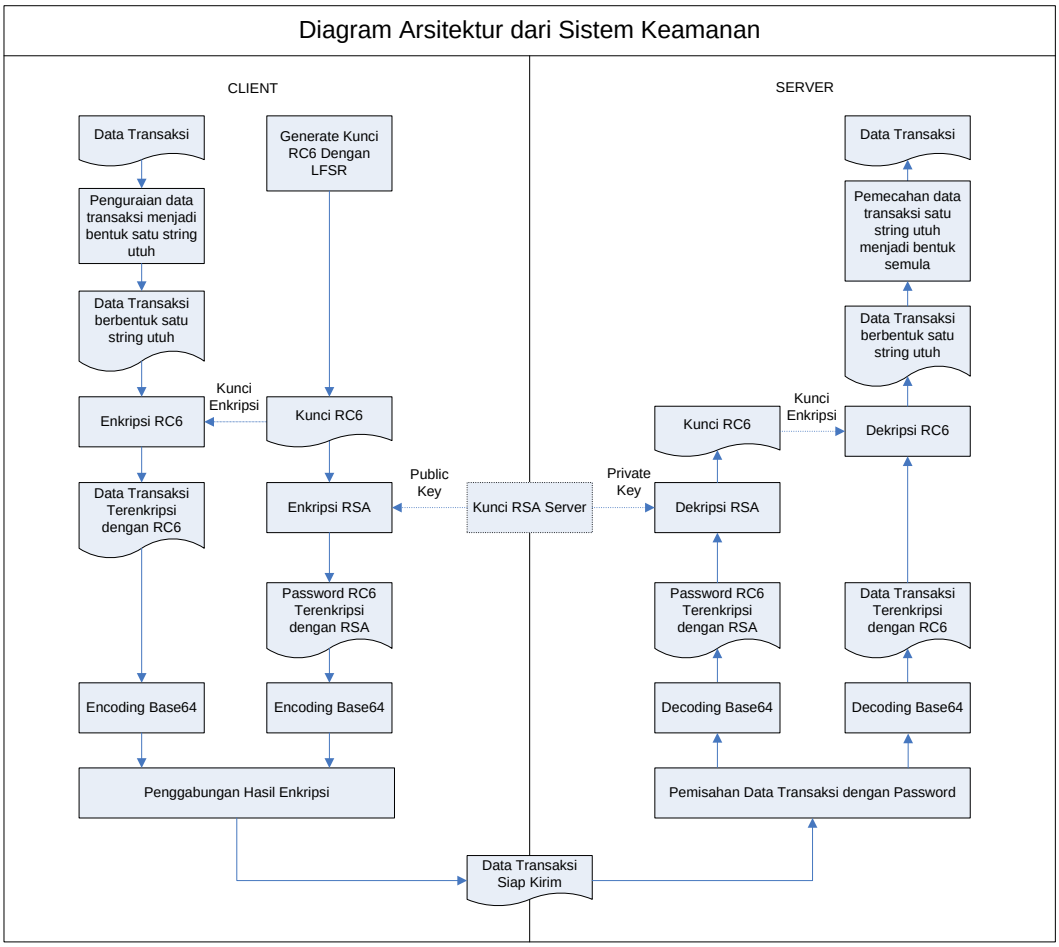
2.1. Tahap Pengerjaan

Pengembangan dari sistem keamanan *e-commerce* menggunakan tahap pengerjaan sebagai berikut:

1. Analisa sistem, yaitu melakukan analisa yang lebih mendalam terhadap sistem transaksi *e-commerce* secara umum.
2. Menentukan pengembangan metode dari sistem keamanan *e-commerce*.
3. Melakukan perancangan sistem dengan menggunakan perangkat pemodelan sistem untuk menggambarkan setiap proses yang akan ditangani, aliran data, dan proses pengamanan data.
4. Rancangan sistem diaplikasikan ke dalam halaman *web* baik untuk sisi *client* maupun sisi *server*.
5. Pengujian sistem keamanan dan sistem *e-commerce* secara menyeluruh dari setiap fasilitas atau menu yang ada pada sistem.

2.2. Diagram Arsitektur dari Sistem Keamanan

Sistem keamanan pada *e-commerce* menggunakan sistem keamanan yang bekerja sepenuhnya menggunakan *script* pada pemrograman berbasis *web* yang telah diatur sedemikian rupa dari sisi *client* dan sisi *server*. Sistem keamanan pada bagian *server* menggunakan bahasa pemrograman *PHP*, sedangan sistem keamanan pada bagian *client* menggunakan bahasa pemrograman *JavaScript*. Sistem keamanan *e-commerce* memiliki arsitektur program yang ditunjukkan oleh Gambar 1.



Gambar 1. Diagram arsitektur dari sistem keamanan untuk e-*commerce*

Gambar 1 menunjukkan cara kerja dari sistem keamanan *e-commerce* dengan cara dibagi menjadi dua sisi, yaitu pada sisi *server* dan sisi *client*. Pengembangan sistem keamanan *e-commerce* berfokus pada keamanan data transaksi yang dikirimkan dari *client* karena data transaksi tersebut memuat banyak informasi sensitif yang dikirimkan dari *client* misalnya data kartu kredit, data *username* dan *password*, serta informasi pribadi dari *client* lainnya misalnya nomor telepon. Cara kerja dari sistem keamanan *e-commerce* pada sisi *client* dibahas pada pembahasan berikut:

1. Pelanggan memasukkan data transaksi ke dalam *form* yang diikuti dengan menekan tombol *submit.*
2. Sistem keamanan memblokir percobaan transaksi dan memutuskannya setelah mendeteksi percobaan *submit*.
3. Sistem keamanan meng-*generate* kunci *RC6* secara *random* dengan algoritma *LFSR*, kemudian hasil *generate* diubah menjadi karakter *ASCII* agar dapat digunakan pada langkah berikutnya.
4. Sistem keamanan mengurai semua *input* pada data transaksi dengan cara yang digunakan oleh kebanyakan *browser*, yaitu dengan cara mengambil semua *name* dan *value* pada setiap *input* dalam data transaksi, kemudian merangkainya menjadi satu *string* dengan

tanda pemisah "=" (sama dengan). Proses *penggabungan* string dilakukan pada setiap data *input* yang telah dimasukkan. Sekumpulan *string* yang telah digabungkan dari setiap *input* dirangkai kembali menjadi satu *string* utuh dengan tanda pemisah "&" (dan).
5. Sistem keamanan mengenkripsi data transaksi yang telah di-*input* dengan algoritma *RC6* menggunakan kunci *RC6* yang di-*generate* dengan metode *LFSR* sebelumnya.
6. Kunci *RC6* dienkripsi dengan algoritma *RSA* menggunakan kunci publik dari *server*.
7. Data transaksi dan kunci *RC6* yang telah dienkripsi kemudian di-*encode* dengan metode *Base64* kemudian digabungkan dan dikirimkan ke *server*.

Cara kerja dari sistem keamanan *e-commerce* pada sisi *server* dibahas pada pembahasan berikut
1. Data yang telah sampai di *server* dipisahan antara data transaksi terenkripsi dengan kunci terenkripsi berdasarkan tanda sama dengan (=).
2. Kunci enkripsi *RC6* di-*decode* dengan metode *Base64* kemudian didekripsi dengan algoritma dekripsi *RSA* menggunakan kunci privat dari *server*.
3. Data transaksi di-*decode* dengan metode *Base64* kemudian didekripsi dengan algoritma *RC6* menggunakan kunci *RC6* yang telah didekripsi sebelumnya.
4. Data transaksi diuraikan kembali oleh sistem keamanan ke dalam bentuk variabel *input* dengan cara memisahkan satu string utuh ke dalam beberapa bagian berdasarkan tanda dan (&). *String* yang sudah dipisahkan dimasukkan ke dalam variabel *array* $_POST dengan *name* dan *value* yang terdapat pada masing-masing *string* tersebut yang terpisahkan dengan tanda sama dengan (=).

Data transaksi yang telah diolah oleh sistem dekripsi pada *server* dapat digunakan oleh sistem *e-commerce* yang mengadopsinya tanpa melakukan penyesuaian pada sistem *web* secara keseluruhan. Sistem keamanan pada sistem *e-commerce* dibuat menggunakan pemrograman berbasis *web* sehingga sistem keamanan dapat mengamankan data transaksi pada sistem *e-commerce* tanpa menggunakan protokol keamanan lainnya.

3. Kajian Pustaka

Kajian pustaka membahas tentang rangkuman dari semua teori yang digunakan di dalam pengembangan sistem keamanan *e-commerce*. Pengembangan sistem keamanan *e-commerce* menggunakan beberapa teori yang menunjang kegiatan penelitian yaitu *State of the Art*, *e-commerce*, *PHP*, *JavaScript*, *RC6*, *Base64*, dan *RSA*. Subbab berikut memuat tentang semua rangkuman teori yang menjadi acuan dalam pengembangan sistem keamanan *e-commerce* yang dilengkapi dengan gambar dan tabel pendukung.

3.1. State of the Art

Penelitian dari sistem keamanan *e-commerce* menggunakan beberapa *state of the art* yang menjadi acuan dan perbandingan dalam pembuatan jurnal sistem keamanan *e-commerce* yang dikembangkan yaitu sebagai berikut:
1. Mateus Mas Belalawe membahas penelitian tentang studi kasus pada sistem keamanan *e-commerce* terhadap situs www.buahonline.com. Penelitian tersebut membahas sistem keamanan *e-commerce* yang masih menggunakan protokol keamanan yaitu *SSL* (*Secure Socket Layer*), *TLS* (*Transport Layer Services*), *SET* (*Secure Electronic Transaction*), dan *PGP* (*Pretty Good Privacy*) [2].
2. Andre M. R. Wajong dan Carolina Rizki Putri membahas penelitian tentang keamanan *e-commerce*. Penelitian tersebut membahas sistem keamanan *e-commerce* yang masih menggunakan protokol keamanan yaitu *PKI* (*Public Key Infrastructure*), *Digital Signature*, *Certificate Digital*, *SSL* (*Secure Socket Layer*), *TLS* (*Transport Layer Services*), *SET (Secure Electronic Transaction*), dan *PGP* (*Pretty Good Privacy*) [3].
3. Samsul Huda dkk. membahas penelitian tentang implementasi keamanan *E-Commerce* yang menggunakan *schnorr digital signature*. Penelitian tersebut membahas sistem keamanan *e-commerce* yang masih menggunakan protokol keamanan yaitu *SSL* (*Secure Socket Layer*), dan *Digital Signature* [4].

3.2. Pengertian *e-commerce*

E-commerce adalah kegiatan transaksi barang atau jasa yang dilakukan secara jarak jauh (*online*) [1]. Proses dalam sistem pembelanjaan *e-commerce* adalah:
1. Pembuatan *web site* untuk produk dan layanan

2. Pemesanan secara *online*
3. Otomasi akun pelanggan secara aman (melalui nomor rekening atau nomor kartu kredit)
4. Pembayaran secara *online*
5. Penanganan transaksi pasca pembayaran

3.3. Sistem keamanan *e-commerce*

Sistem keamanan pada *e-commerce* mencakup beberapa aspek penting yang dijadikan dasar, yaitu aspek-aspek keamanan, macam-macam ancaman, dan solusi dari kekurangan sistem *e-commerce*. Semua aspek penting pada keamanan *e-commerce* sangat berpengaruh terhadap tingkat keamanan pada sistem keamanan *e-commerce* secara keseluruhan. Subbab berikut memuat tentang materi yang berhubungan dengan sistem keamanan *e-commerce* yang dibahas dalam bentuk subbab.

3.3.1. Aspek-Aspek Keamanan

Proses Kriptografi tidak hanya merahasiakan data transaksi tetapi harus memenuhi aspek lainnya yaitu [5]:

1. *Authentication*, yaitu pengirim pesan harus benar-benar berasal dari pengirim yang bersangkutan.
2. *Integrity*, yaitu isi pesan harus benar-benar utuh dan tidak diubah oleh orang lain.
3. *Nonrepudiation*, yaitu pengirim pesan tidak dapat menyangkal bahwa pesan tersebut dikirim oleh yang bersangkutan.
4. *Authority*, yaitu pesan yang dikirim hanya dapat diubah oleh pihak yang berwenang.

3.3.2. Macam-macam ancaman

Macam-macam ancaman yang terjadi dalam sistem *e-commerce* adalah [6]:

1. *System Penetration*, yaitu seseorang yang tidak berhak dapat mengakses sistem komputer dan dapat melakukan segalanya.
2. *Authorization Violation*, yaitu penyalahgunaan wewenang yang dimiliki oleh seseorang yang berhak.
3. *Planting*, yaitu melakukan penyerangan secara terencana, misalnya memasukkan *Trojan Horse* dan melakukan penyerangan dengan waktu yang telah ditentukan sebelumnya.
4. *Communications Monitoring*, yaitu melakukan monitoring semua informasi rahasia.
5. *Communications Tampering*, yaitu mengubah pesan di tengah jalan oleh penyerang di dalam proses transimisi data dan mengganti sistem server dengan sistem server yang palsu.
6. *Denial of Service* (*DoS*), yaitu menolak layanan terhadap *client* yang berhak.
7. *Repudiation*, yaitu menolak aktivitas transaksi karena suatu hal yang disengaja atau kesalahan teknis.

3.3.3. Solusi dari Kekurangan Sistem *E-Commerce* secara umum

Solusi dari beberapa masalah yang telah ditemukan untuk meningkatkan keamanan sistem *e-commerce* yang digunakan saat ini adalah:

1. Menggunakan sistem otentikasi sederhana berbasis *hashing* yang ditanamkan ke dalam sistem *e-commerce* untuk melakukan otentikasi pengesahan dari pelanggan.
2. Menggunakan sistem enkripsi simetris *RC6* yang diperkuat dengan sistem enkripsi *RSA* dan sistem encoding dari *Base64*.
3. Menggunakan sistem enkripsi simetris *RC6* hanya untuk mengamankan isi data transaksi sedangkan untuk kunci enkripsi *RC6-nya* diamankan dengan menggunakan *RSA* baik dari sisi *server* maupun dari sisi *client*.

3.3. *PHP*

PHP adalah bahasa pemrograman berbasis *web* yang dijalankan pada sisi server (*server-side*). PHP diperkenalkan oleh Rasmus Lerdorf pada tahun 1994 [7]. *PHP* didirikan untuk *Personal Home Page* pada awalnya, kemudian sekarang berganti singkatan menjadi *PHP: Hypertext Preprocessor*.

Kode *PHP* dapat dibangun pada sistem halaman *web* dengan cara membangunnya dengan bahasa *PHP* murni, digabungkan dengan kode *HTML*, atau dikombinasikan dengan berbagai *template engine* dan *web framework*. Kode *PHP* diproses oleh *PHP interpreter* yang

diimplementasikan sebagai bagian modul dari *web server*. *Web server* mengirimkan hasil output kepada *client* dalam bentuk bagian dari halaman *web* yang dihasilkan karena kode *PHP* dapat menghasilkan kode *HTML web page*, gambar, atau data dalam bentuk lainnya.

3.4. *JavaScript*

JavaScript adalah bahasa berbasis *web* yang dijalankan pada sisi *client* (*client side*). *JavaScript* digunakan dalam pembuatan *website* agar lebih dinamis dengan cara memerintahkan *browser* untuk mengeksekusi *script JavaScript* langsung di dalam halaman *HTML* [8]. Hasil eksekusi pada *JavaScript* dapat memanipulasi halaman *web* serta objek yang ada di dalam halaman *web* tersebut. *JavaScript* sangat berbeda dengan *Java* dan memiliki semantik yang sangat berbeda. Sintaks pada *JavaScript* diturunkan dari bahasa *C*, namun semantik dan desainnya dipengaruhi oleh bahasa pemrograman *self* dan bahasa pemrograman *scheme*.

3.5. Algoritma *RC6*

Algoritma *RC6* adalah salah satu dari bentuk standar algoritma *Advanced Encryption Standard* (*AES*). Algoritma *RC6* ditemukan oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin. Algoritma *RC6* merupakan pengembangan dari algoritma *RC5* [9]. Algoritma *RC6* adalah algoritma yang ditulis dalam bentuk parameter *RC6-w/r/b*, dengan ketentuan sebagai berikut:

1. w merupakan ukuran *word* dalam satuan *bit*
2. r merupakan jumlah iterasi selama proses enkripsi
3. b adalah ukuran kunci enkripsi dalam satuan *byte*.

Algoritma *RC6* menggunakan nilai parameter yang telah ditetapkan oleh Standar *AES* yaitu w = 32, r = 20, dan b = antara 16, 24, dan 32 *byte*. Cara kerja algoritma *RC6* adalah dengan membagi blok yang berukuran 128-bit menjadi empat buah blok yang berukuran 32-bit, kemudian melakukan enam operasi dasar sebagai berikut.

1. A + B: Operasi penjumlahan bilangan
2. A – B: Operasi pengurangan bilangan
3. A xor B: Operasi exclusive-OR (XOR)
4. A x B: Operasi perkalian bilangan
5. A <<< B: A digeser ke kiri sebanyak B
6. A >>> B: A digeser ke kanan sebanyak B

Detail dari masing-masing proses pada algoritma *RC6* dibahas pada pembahasan beriktnya.

3.5.1. Pembangkitan Kunci

Pembangkitan kunci diproses menggunakan nilai parameter untuk algoritma *RC6-w/r/b* yaitu w=32, r=20 dan b=32. Nilai parameter yang telah ditetapkan tersebut adalah nilai standar yang telah ditetapkan oleh standar *AES* [9]. Algoritma untuk membangkitkan kunci enkripsi pada *RC6* dijelaskan pada Gambar 2.

```
S[ 0 ] = 0xB7E15163
for i = 1 to 43 do {
        S[i] = S[i-1] + 0x9E3779B9
}

A = B = i = j = 0
for k = 1 to 132 do {
    A = S[i] = (S[i] + A + B) <<< 3
    B = L[j] = (L[j] + A + B) <<< (A + B)
    i = (i + 1) mod 44
    j = (j + 1) mod c
}
```

Gambar 2. Pembangkitan kunci pada RC6
Sumber: Muhammad Zulham, 2014

Gambar 2 menunjukkan *pseudocode* dari algoritma pembangkitan kunci untuk proses enkripsi dan dekripsi pada *RC6*. Penjelasan detail dari algoritma pembangkitan kunci adalah sebagai berikut:

1. Pengguna memasukkan sebuah kata kunci (*password*) sepanjang b *byte*, *password* tidak boleh kosong dan tidak boleh melebihi 255 *byte*.
2. *Password* yang telah di-*input* dan di-*submit* ditempatkan ke dalam *array* L dengan tipe data 32-bit *words*. *Byte* pertama pada password dimasukkan ke dalam *array* L elemen ke-0, *byte* kedua pada *array* L elemen ke-1, dan seterusnya hingga *byte* terakhir diletakkan pada *array* L elemen ke b-1.
3. Algoritma *RC6* membentuk sub-kunci yang dibentuk berdasarkan *array* L yang telah dibuat sebelumnya ke dalam *array* S yang berukuran 43 tipe data. *Array* S elemen ke-0 diberi nilai konstanta 0xB7E15163 dan *array* S yang berikutnya diberi nilai hasil penambahan dari nilai *array* S sebelumnya dengan nilai konstanta 0x9E3779B9.
4. Algoritma *RC6* melakukan penguatan kunci dengan operasi penggabungan nilai heksadesimal dari nilai *array* S dengan nilai *array* L yang diikuti dengan operasi rotasi biner. Proses penguatan kunci dilakukan sebanyak 132 iterasi, dan *array* S dari hasil iterasi penguatan kunci selanjutnya digunakan pada proses enkripsi atau dekripsi pada *RC6*.

3.5.2. Algoritma Enkripsi *RC6*

Proses enkripsi pada algoritma *RC6* melibatkan proses pemecahan blok *register* 128-bit menjadi empat buah blok *register* 32-bit. Proses enkripsi pada algoritma *RC6* kemudian dilanjutkan dengan proses *whitening* awal, iterasi enkripsi sebanyak 20 kali, dan proses *whitening* akhir [9]. Penjelasan dari algoritma enkripsi *RC6* dapat dilihat pada Gambar 3.

```
B = B + S[0];
D = D + S[1];
for i = 1 to 20 do {
    t = (B x (2B + 1)) <<< 5;
    u = (D x (2D + 1)) <<< 5;
    A = ((A xor t) <<< u) + S[2i]
    C = ((C xor u) <<< t) + S[2i + 1];
    (A,B,C,D) = (B,C,D,A);
}
A = A + S[42];
C = C + S[43];
```

Gambar 3. Algoritma enkripsi pada *RC6*
Sumber: Scoot Contini, 1998

Gambar 3 menunjukkan *pseudocode* dari algoritma enkripsi pada *RC6*. Penjelasan detail dari algoritma enkripsi *RC6* adalah sebagai berikut:

1. Algoritma *RC6* menggunakan empat buah blok *register* 32-bit sehingga dapat memproses empat karakter *ASCII* dalam satu blok dan dapat memproses 16 karakter *ASCII* dalam satu iterasi. Algoritma *RC6* harus mengubah keempat karakter ke dalam bentuk heksadesimal sesuai dengan nomor urut karakter pada tabel *ASCII* kemudian mengkonvesinya ke dalam bentuk bilangan heksadesimal.
2. Algoritma *RC6* memulai proses *whitening* awal yaitu menjumlahkan blok B dengan blok sub kunci S elemen ke-0 dan blok D dengan blok sub kunci S elemen ke-1. Blok sub kunci didapat dari hasil algoritma pembangkitan kunci dari kata kunci yang dimasukkan oleh pengguna.
3. Algoritma *RC6* memulai proses iterasi enkripsi sebanyak 20 kali dalam setiap proses enkrpsi yang dilakukan. Setiap iterasi selalu melibatkan dua buah sub-kunci berikutnya, misalnya iterasi 1 menggunakan sub-kunci S elemen ke-2 dan S elemen ke-3, dan seterusnya. Setiap akhir dari masing-masing iterasi, posisi blok yang paling kiri dipindahkan ke posisi paling kanan sehingga urutan blok berubah dari A, B, C, D menjadi B, C, D, A.
4. Algoritma *RC6* mengakhiri proses enkripsi dengan proses *whitening* akhir yaitu menjumlahkan blok A dengan blok sub kunci S elemen ke-42 dan blok C dengan blok sub kunci S elemen ke-43.

3.5.3. Algoritma Dekripsi *RC6*

Proses dekripsi pada algoritma *RC6* melibatkan proses pemecahan blok *register* 128-bit menjadi empat buah blok *register* 32-bit. Proses dekripsi pada algoritma *RC6* kemudian dilanjutkan dengan proses *whitening* awal, iterasi dekripsi sebanyak 20 kali, dan proses

whitening akhir [9]. Proses dekripsi pada algoritma *RC6* merupakan kebalikan dari proses enkripsi pada algoritma *RC6*. Gambar 4 menjelaskan algoritma dekripsi pada *RC6*:

```
C = C - S[43];
A = A - S[42];
for i = 20 downto 1 do {
    (A,B,C,D) = (D,A,B,C);
    u = (D x (2D + 1)) <<< 5;
    t = (B x (2B + 1)) <<< 5;
    C = ((C - S[2i + 1]) >>> t) xor u;
    A = ((A - S[2i ]) >>> u) xor t;
}
D = D - S[1];
B = B - S[0]
```

Gambar 4. Algoritma dekripsi pada *RC6*
Sumber: Scoot Contini, 1998

Gambar 4 menjelaskan tentang *pseudocode* dari algoritma dekripsi pada *RC6*. Penjelasan dari algoritma dekripsi *RC6* adalah sebagai berikut:

1. Algoritma *RC6* hanya bisa menggunakan *password* dan sub-kunci yang sama dengan proses enkripsi sebelumnya.
2. Algoritma RC6 memulai proses *whitening* awal yaitu mengurangi blok C dengan dengan blok sub kunci S elemen ke-43 dan blok C dengan dengan blok sub kunci S elemen ke-42
3. Algoritma RC6 memulai proses iterasi dekripsi sebanyak 20 kali namun urutan iterasinya dilakukan terbalik dari proses enkripsi. Setiap awal dari masing-masing iterasi, posisi blok yang paling kanan dipindahkan ke posisi paling kiri sehingga urutan blok berubah dari A, B, C, D menjadi D, A, B, C.
4. Algoritma RC6 mengakhiri proses dekripsi dengan proses *whitening* akhir yaitu menjumlahkan blok D dengan blok sub kunci S elemen ke-1 dan blok B dengan blok sub kunci S elemen ke-0.

3.6. Algoritma *Base64*

Algoritma *Base64* adalah algoritma *encoding* yang mengkonversi karakter *ASCII* menjadi karakter representasi *radix-64* [7]. Algoritma *encoding Base64* berfungsi untuk mentransmisikan data ke dalam jaringan secara aman dan terbebas dari kesalahan fungsi transmisi yang disebabkan oleh karakter *ASCII* yang memiliki fungsi khusus di dalam operasi tertentu (misalnya karakter dengan kode *ASCII* 13 berarti karakter yeng memberi perintah *enter*).

3.6.1. Susunan Alfabet *Encoding Base64*

Susunan alphabet *encoding* pada *Base64* pada sistem keamanan *e-commerce* menggunakan sistem alfabet *URL and Filename safe*, yaitu susunan alfabet *Base64* yang sudah dimodifikasi khusus agar dapat digunakan pada *web browser* tanpa mengalami kendala teknis yang menghambat proses transmisi data (misalnya *charater escaping*). Susunan alfabet *Base64* dapat dilihat pada jurnal publikasi milik Josefsson S. yang berjudul "The Base16, Base32, and Base64 Data Encodings".

3.6.2. Cara Kerja Algoritma *Encoding Base64*

Proses *encoding base64* mengambil tiga karakter *ASCII* dan mengubahnya menjadi empat karakter representasi *radix-64* dalam setiap iterasi. Tiga karakter yang diambil oleh algoritma *Base64* diubah menjadi bentuk biner, kemudian digabungkan menjadi satu deret *bit* yang utuh, lalu dipecah kembali menjadi empat deret angka *bit* yang masing-masing-masing berjumlah sebanyak enam *bit*. Deret angka biner yang sebanyak enam *bit* tersebut diubah menjadi karakter representasi *radix-64* yang selanjutnya digunakan dalam proses transmisi data [7]. Proses encoding dari algoritma *Base64* dapat dilihat pada Gambar 5.

Karakter *ASCII* (Input)	:A	S	C	
Kode *ASCII*	:65	83	67	
Bentuk Biner Kode *ASCII*	:01000001	01010011	01000011	
Penggabungan dari semua biner	:	010000010101001101000011		
Bentuk Biner Kode *Base64*	:010000	010101	001101	000011
Kode *Base64*	:16	21	13	3
Karakter *Base64* (Output)	:Q	V	N	D

Gambar 5. Konversi *String* ke *Base64*

Gambar 5 adalah proses dari konversi *string* biasa menjadi karakter *Base64*. Karakter dari *string* diubah menjadi angka heksadesimal berdasarkan kode *ASCII*, kemudian dikonversi lagi menjadi angka biner. Semua angka biner hasil konversi dari karakter pada *string* digabungkan menjadi satu untuk proses selanjutnya. Susunan angka biner dikonversi ke dalam bentuk karakter representasi *radix-64* dengan cara diambil setiap enam angka bit, kemudian dikonversi menjadi angka desimal, yang kemudian dikonversi lagi menjadi karakter representasi *radix-64*.

Setiap tiga karakter pada *string ASCII* (*string* biasa) selalu menghasilkan empat karakter *base64*, dan setiap karakter *Base64* selalu berjumlah kelipatan empat. Jika jumlah karakter hasil *output* dari konversi *Base64* tidak berjumlah kelipatan empat, maka hasil output selalu ditambahkan dengan *padding* (Karakter "=" (sama dengan)).

3.7. Algoritma *RSA*

Algoritma *RSA* adalah algoritma enkripsi asimetris yang pertama ditemukan di dunia. Algoritma *RSA* ditemukan oleh Ronald R. Rivest, Adi Shamir dan Leonard Adleman Algoritma pada tahun 1977. *RSA* menggunakan dua buah kunci yaitu kunci publik untuk proses enkripsi dan kunci privat yang digunakan untuk proses dekripsi [10].

3.7.1. Pembangkitan Kunci

Kunci untuk algoritma *RSA* dihasilkan dengan cara sebagai berikut:

1. Tentukan dua bilangan bulat prima untuk p dan q. Bilangan prima p dan q harus dipilih secara acak dan panjang bit harus sama.
2. Hitunglah nilai n = pq. Nilai n digunakan sebagai modulus untuk operasi kunci publik dan kunci privat. Panjang kunci dinyatakan dalam *bit* dan disebut dengan *key length*.
3. Hitunglah φ(n) = φ(p)φ(q) = (p−1)(q−1) = n-(p+q-1), dengan m adalah fungsi *euler totient*.
4. Pilih sebuah *integer* e sehingga 1<e<φ(n) dan gcd(e,φ(n)) = 1.
5. Tentukan d dengan rumus d = (1 + nm)/e.

Nilai hasil e dan n digunakan sebagai kunci publik, sedangkan nilai d dan n digunakan sebagai kunci privat.

3.7.2. Enkripsi dan Dekripsi Algoritma *RSA*

Prinsip pada proses enkripsi dan dekripsi pada sistem *RSA* adalah mengkonversi pesan m menjadi angka *integer*, kemudian menggunakan rumus $c = m^e \pmod n$ untuk proses enkripsi beserta rumus $m = c^d \pmod n$ untuk proses dekripsi, dengan ketentuan c adalah pesan terenkripsi dengan bentuk *integer*, m adalah pesan terdekripsi dengan bentuk *integer*, e adalah nilai kunci publik, d adalah nilai kunci privat, serta n adalah modulus yang digunakan dalam operasi enkripsi dan dekripsi. Contoh dari proses enkripsi dan dekripsi dapat dilihat pada skenario berikut.

1. Pilih dua bilangan prima yang berbeda, misalnya p = 61 dan q = 53
2. Hitunglah nilai n = pq. Jika p = 61 dan q = 53, maka n = 61 x 53 = 3233
3. Hitunglah *totient* φ(n) = (p-1)(q-1). Nilai *totient* φ(n) adalah (61 – 1)(53 – 1) = 60 x 52 = 3120
4. Pilih angka e dengan 1 < e < 3120 dan *coprime* dengan 3120. Angka yang relatif *prime* dengan e adalah angka yang sama-sama memiliki faktor pembagi terbesar adalah 1. Rumus secara matematis adalah gcd(e,m) = 1. Nilai e adalah ^{let}e = 17
5. Cari angka d sehingga e*d = 1 mod (m) atau d = (1+nm)/e. Nilai d adalah 17 x 2753 mod 3120 = 1

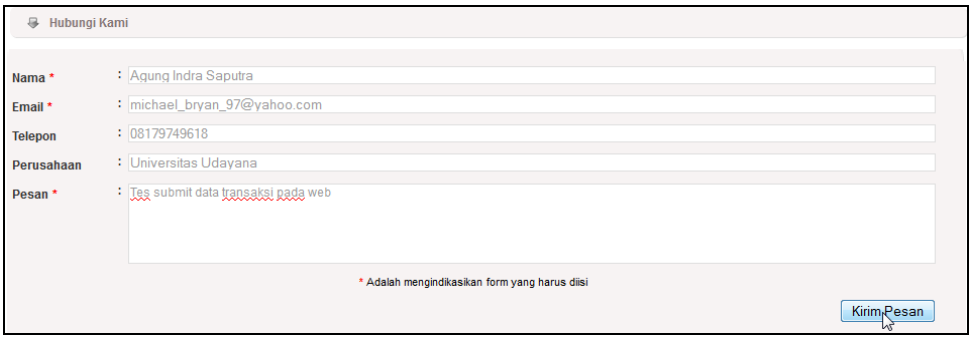
6. Hasil perhitungan sebelumnya akan menghasilkan kunci publik e = 17 dan n = 3233 dan kunci privat d = 2753 dan n = 3233
7. Proses enkripsi menggunakan contoh pesan dengan karakter *ASCII* dengan isi pesan m, fungsi enkripsinya adalah c(m) = m^{17} mod 3233. Contoh enkripsi untuk pesan dengan karakter *ASCII* bernilai 65 adalah c = 65^{17} mod 3233 = 2790
8. Proses dekripsi dari pesan *ciphertext* c adalah m(c) = c^{2753}. Contoh pesan dengan karakter *ciphertext* d adalah m = 2790^{2753} mod 3233 = 65

4. Hasil dan Pembahasan

Hasil dari pembahasan pengembangan sistem keamanan *e-commerce* melibatkan tiga proses pengujian yang dilakukan dari proses penelitian yaitu pengujian *submit form*, pengujian hasil penyadapan, dan pengujian kecepatan sistem. Masing-masing dari proses pengujian dibahas pada pembahasan berikutnya dalam bentuk subbab. Subbab berikut membahas proses dan hasil dari ujicoba pengembangan sistem keamanan pada *e-commerce*.

4.1. Pengujian Saat Melakukan *Submit Form*

Pengujian dilakukan menggunakan *form* pada sistem halaman *web* yang telah diberikan sistem keamanan. Contoh dari pengujian *submit form* dapat dilihat pada Gambar 6.



Gambar 6. *Form* pada halaman *web* yang diujicobakan

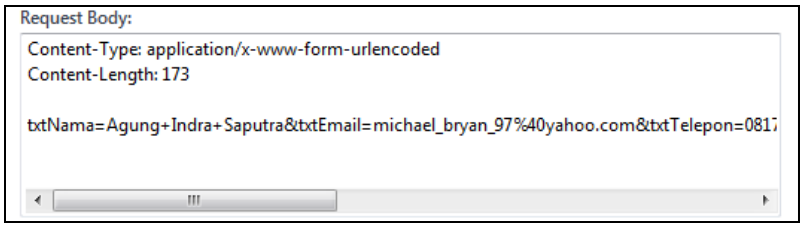
Pengguna memasukkan data transaksi ke dalam *form* seperti biasa kemudian menekan tombol kirim pesan untuk pengiriman ke *server*. Sistem keamanan mengamankan data transaksi pengguna ketika pengguna meng-klik tombol kirim pesan dengan cara sebagai berikut:

1. Pencegatan perintah *submit* secara *default*
2. Generate kunci *RC6* secara *random*
3. Pengambilan setiap nilai *name* dan *value* pada setiap *input*
4. *Escaping* dan perangkaian dari nilai *name* dan *value* setiap *input* menjadi satu *string* utuh
5. Enkripsi rangkaian *string* dengan algoritma *RC6* dengan kunci *RC6* yang sebelumnya telah dibangkitkan secara *random*
6. *Encoding* hasil enkripsi *string* dengan algoritma *Base64*
7. Enkripsi kunci *RC6* dengan algoritma *RSA*
8. *Encoding* hasil enkripsi kunci *RC6* dengan algoritma *Base64*
9. Penggabungan hasil encoding *string* dengan hasil encoding kunci *RC6*
10. Pengiriman hasil pengamanan ke *server*

Data transaksi yang dikirimkan oleh pelanggan dienkripsi dengan algoritma *RC6* menggunakan bahasa pemrograman *JavaScript*, sedangkan data yang telah sampai di *server* didekripsi dengan algoritma *RC6* menggunakan bahasa pemrograman *PHP*.

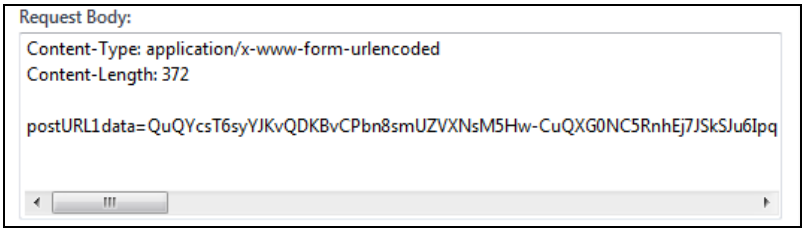
4.2. Hasil Penyadapan

Data transaksi yang dikirimkan disadap oleh *browser* Mozilla Firefox menggunakan fitur *network* pada Mozilla Firefox yang dapat diakses pada menu *tools*, *web developer*, *network*. Proses penyadapan menggunakan fitur dari Mozilla Firefox karena dapat digunakan di dalam *web server* lokal (*localhost*). Hasil penyadapan data transaksi tanpa sistem keamanan *e-commerce* dapat dilihat pada Gambar 7.



Gambar 7. Hasil penyadapan *form* tanpa sistem keamanan menggunakan fitur *network* pada *Mozilla Firefox*

Gambar 7 menunjukkan hasil penyadapan data transaksi tanpa sistem keamanan *e-commerce*. Data transaksi secara umum memiliki informasi pada bagian *request body* yang berisi nilai *name* dan *value* pada masing-masing *input* yang dikirimkan dengan bentuk *plaintext*. *Hacker* dapat melihat isi dari data yang dikirimkan ke dalam *server* jika sistem *e-commerce* tidak menggunakan sistem keamanan. Sistem keamanan *e-commerce* berfungsi untuk mengenkripsi data transaksi yang dikirimkan sehingga hacker tidak dapat melihat isi dari data transaksi setelah disadap. Hasil penyadapan data transaksi dengan sistem keamanan *e-commerce* dapat dilihat pada Gambar 8.

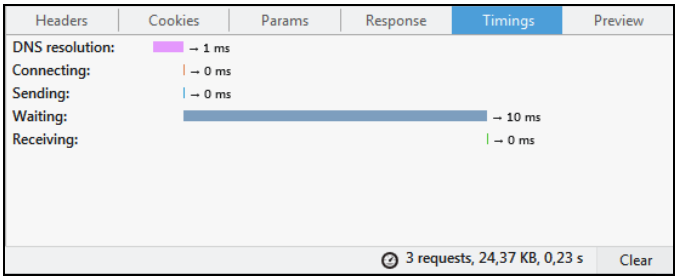


Gambar 8. Hasil penyadapan *form* dengan sistem keamanan menggunakan fitur *network* pada *Mozilla Firefox*

Gambar 8 menunjukkan hasil penyadapan data transaksi dengan sistem keamanan. Sistem keamanan pada *e-commerce* berfungsi untuk menggabungkan semua nilai *name* dan *value* pada semua *input* di dalam *form*, kemudian mengenkripsinya dengan algoritma *RC6* dan mengirimkannya sebagai *value* dari *input name* postURL1data. Sistem keamanan tersebut berfungsi untuk mencegah *hacker* untuk membaca isi pesan yang dikirimkan ke *server* sehingga data yang dikirimkan tetap aman.

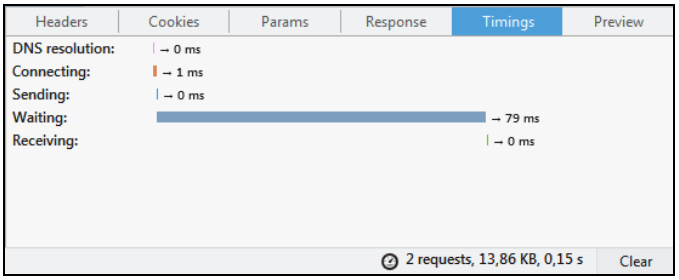
4.3. Uji Kecepatan Sistem

Pengujian kecepatan pada sistem keamanan *e-commerce* membandingkan kecepatan sistem *web* sebelum diberikan sistem keamanan dengan sesudah diberikan sistem keamanan. Kecepatan pada sistem *web* tanpa sistem keamanan dapat dilihat pada Gambar 9.



Gambar 9. Hasil pengujian dari sistem *web* tanpa sistem keamanan

Gambar 9 menunjukkan hasil pengujian kecepatan untuk *submit form* pada halaman *web* tanpa sistem keamanan yaitu selama 0,23 detik dengan jumlah *request* sebanyak 3 *requests*. Lama waktu yang dibutuhkan untuk proses *waiting response* dari *server* adalah 10 ms. Kecepatan pada sistem *web* dengan sistem keamanan yang dihasilkan dapat dilihat pada Gambar 10.



Gambar 10. Hasil pengujian dari sistem *web* dengan sistem keamanan

Gambar 10 menunjukkan hasil pengujian kecepatan untuk submit form pada halaman *web* dengan sistem keamanan yang dihasilkan yaitu selama 0,15 detik dengan jumlah *request* sebanyak 2 *requests*. Perbedaan waktu antara sistem *web* tanpa sistem keamanan dan dengan sistem keamanan sebanyak 0,08 detik. Lama waktu yang dibutuhkan untuk proses *waiting response* dari *server* adalah 79 ms. Hasil penelitian mencatat perbedaan waktu antara *web* tanpa sistem keamanan dengan *web* yang diintegrasikan dengan sistem keamanan adalah 69 ms. Hasil pengujian kecepatan yang diperoleh menunjukkan bahwa terdapat perbedaan yang tidak signifikan dalam masalah kecepatan proses *submit form* pada sistem *web*, sehingga sistem keamanan yang dihasilkan dapat diterapkan ke dalam sistem *web* tanpa mengurangi kecepatan akses pada sistem *web*.

5. Kesimpulan

Sistem keamanan *e-commerce* yang dihasilkan berfungsi saat pengguna menekan tombol *submit* setelah pengguna memasukkan data transaksi. Sistem keamanan mengamankan sistem *e-commerce* dengan cara mengimplementasikan sistem enkripsi simetris *RC6*, sistem enkripsi asimetris *RSA*, serta sistem *encoding Base64.* Data transaksi diamankan menggunakan *script* dari bahasa pemrograman *JavaScript* dan *PHP* sehingga bersifat fleksibel terhadap *web server* dan dapat mengamankan data transaksi tanpa protokol keamanan lainnya. Sistem keamanan *e-commerce* dapat diintegrasikan ke dalam sistem *e-commerce* sehingga tidak terlihat secara kasat mata oleh pengguna sistem *e-commerce* tetapi dapat mengamankan data transaksi yang dikirimkan sehingga data transaksi tidak dapat disadap oleh pihak luar dan dapat digunakan pada sistem *e-commerce* tanpa mengurangi kecepatan *loading* pada *web browser*.

Daftar Pustaka

[1]. Himawan, Hidayatullah. Keamanan Transaksi E-Commerce dengan Menggunakan SMS. Yogyakarta: UPN Veteran Yogyakarta. 2008
[2]. Balawe, Mateus Mas. Tinjauan Keamanan Sistem Transaksi dan Pembayaran Pada E-Commerce Studi Kasus Toko Online www.buahonline.com. Kupang: STIKOM Artha Buana Kupang. 2013
[3]. Andre M. R. Wajong, Putri, Carolina Rizki. Keamanan dalam Sistem E-Commerce. Jakarta: Bina Nusantara University. 2010
[4]. Huda, Samsul, dkk. Implementasi Sistem Pengamanan E-Commerce Menggunakan Schnorr Digital Signature. Surabaya: Politeknik Elektronika Negeri Surabaya. 2014
[5]. R. A., Esti, Kurniati, A. Pemanfaatan Kriptografi dalam Mewujudkan Keamanan Informasi pada E-Voting di Indonesia. Yogyakarta: UPN Veteran Yogyakarta. 2009
[6]. W. Purbo, Onno, Dkk. Mengenal E-Commerce. Jakarta: Elex Media Komputindo. 2001
[7]. Sholeh, Ahmad Timbul, dkk. Mengamankan Skrip pada Bahasa Pemograman PHP dengan Menggunakan Kriptografi Base64. Garut: Sekolah Tinggi Teknologi Garut. 2013
[8]. B, Indra Yatini. Aplikasi Pengolahan Citra Berbasis Web Menggunakan Javascript dan JQuery. Yogyakarta: STMIK AKAKOM Yogyakarta. 2014
[9]. Defni, Rahmayun, Indra. Enkripsi SMS (Short Message Service) pada Telepon Selular Berbasis Android dengan Metode RC6. Padang: Politeknik Negeri Padang. 2014
[10]. Zulham, Muhammad. Perancangan Aplikasi Keamanan Data Email Menggunakan Algoritma Enkripsi RC6 Berbasis Android. Medan: STMIK Potensi Utama. 2014
[11]. Satriawan, I Wayan Dharma. Aplikasi Enkripsi SMS dengan Metode RSA pada Smartphone Berbasis Android. Denpasar: Universitas Udayana. 2014