Implementasi Kriptosystem menggunakan metode Algoritma ECC dengan Fungsi Hash SHA-256 pada sistem ticketing online

Implementation of Crypthosystem using Method Algorithm ECC with Function of Hash SHA-256 in online ticketing system

Sarfina Adilah[1] , R Rumani M [2] , Marisa W. Paryasto[3]

[1,2,3] Prodi S1 Sistem Komputer, Fakultas Teknik Elektro, Telkom University

Bandung, Indonesia

[1] sarfina.adilah@gmail.com, [2] rumani@telkomuniversity.ac.id, [3]marisa.paryasto@telkomuniversity.ac.id

Abstrak

Seiringnya perkembangan zaman, bidang teknologi semakin maju terutama di bidang Komputer. Sudah banyak pengembangan teknologi baru yang bahkan hampir tidak pernah terpikirkan sama sekali. Salah satunya dalam pembelian tiket saat ini pun sudah bisa membelinya secara online. Tetapi siapa yang menduga jika ternyata pembelian tiket secara online rentan pembobolan data terutama di bagian data pemilik User. Adapun solusi untuk mengatasi pembobolan data tersebut sistem seharusnya memakai Kriptografi. Dengan kriptografi pembeli dapat mengirimkan data pembelian secara aman.

Pada penelitian tugas akhir ini akan dirancang suatu sistem yang dapat menjaga keamanan pada sistem pembelian tiket online dengan menggunakan Algoritma Kurva Eliptik dan dengan *Fungsi Security Hash Algorithm* 256 yang dimana dengan cara pengenkripsian dengan public key dan verifikasi dengan fungsi hash.

Pada pengujian dan analisis ini memiliki Respond Time dengan rata-rata 203.6 ms dengan ukuran kunci sebesar 512 bits dan Signature Key size dari SHA-256 sebesar 256 bytes. Jenis serangan yang sudah dianalisa ialah **Spoofing, Sniffing, dan Man in Middle Attack. Spoofing dapat diatasi dengan penggunaan fungsi enkripsi dan dekripsi pada suatu proses**

Kata kunci: *Elliptic Curve Cryptosystem, Security Hash Algorithm 256, security system, encryption,decryption. Signature key.*

Keyword : ***Elliptic Curve Cryptosystem, Security Hash Algorithm 256, security system, encryption,decryption*****.**

Abstract

Nowadays, Technology is very common especially in Computer Science. there manny new program Developement whose unbelieveable. One of them is Buying the tickets with online system today. But, who knows if ticketing online transaction not very safe because there's more attacker for get our ticket without permission. And the solution is, we can use Crypthograph as Secure Key for any transaction, include for online transaction too.

On final task now, will design a system for keeping secure from online transaction with ECC Algorithm which become for encryption and decryption and Hash Function SHA-256 As Digital Signature for verification.

Experiment and Analysis of final task have a some conclusion, Respon Time have average about 203.6 ms with key size 512 bits and Signature Key from SHA-256 about 256 bits. We analysing there's many attack will crush the application, they are Spoofing, Sniffing, and Man in Middle Attack. The Attacks will stop with using encryption, decryption and signature on the process.

Keyword: *Elliptic Curve Cryptosystem, Security Hash Algorithm 256, security system, encryption,decryption. Signature key.*

---

1. Pendahuluan

Dalam pengembangan teknologi yang sangat pesat saat ini. Teknologi dalam komputer sudah menjadi sangat menjadi bidang pengembangan yang sangat pesat. Dalam hal ini ialah dibutuhkannya keamanan data ketika mengirimkan suatu permintaan pada pembelian barang, dalam hal tiket misalnya. Siapapun dapat bisa membeli tiket dengan mudahnya, salah satu contohnya ialah tiket transportasi udara yang dapat dipesan melalui sistem *online* melalui situs jaringan internet ataupun aplikasi.

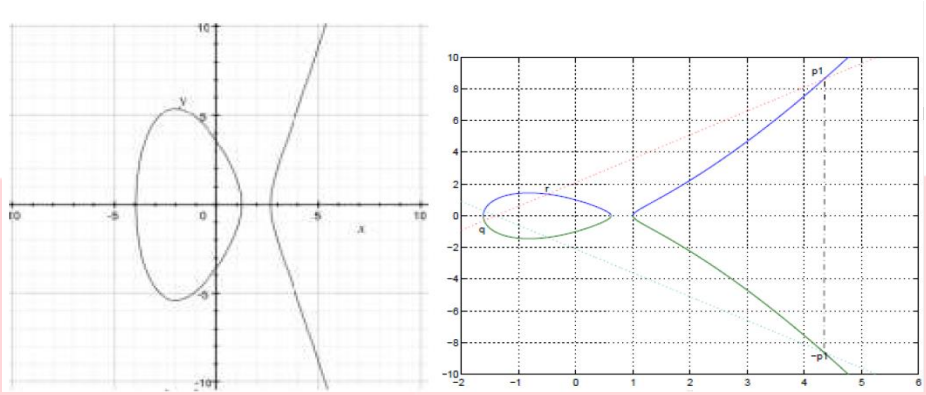
Kriptografi (*cryptography*) berasal dari Bahasa Yunani: "*cryptos*" artinya "*secret*" (rahasia), sedangkan "*graphein*" artinya "*writing*" (tulisan). Jadi kriptografi berarti "*secret writing*" (tulisan rahasia). Jadi dapat disimpulkan Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi [9].

Fungsi *Hash* kriptografi adalah fungsi hash yang memiliki beberapa sifat keamanan tambahan sehingga dapat dipakai untuk tujuan keamanan data.. SHA atau *Security Hash Algorithm* adalah Serangkaian fungsi cryptographic hash yang dirancang oleh National Security Agency (NSA) dan diterbitkan oleh NIST sebagai US Federal Information Processing Standart. Jenis-jenis SHA yaitu SHA-1 dan SHA-2 yang terbagi menjadi SHA-224, SHA-256, SHA-384 dan SHA-512[6].

2. Dasar Teori

2.1. Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC)termasuk sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva eliptik. Karena alasan tersebut, algoritma kriptografi kurva eliptik mempunyai keuntungan jika dibandingkan dengan algoritma kriptografi kunci publik lainnya yaitu dalam hal ukuran panjang kunci yang lebih pendek tetapi memiliki tingkat keamanan yang sama. Penggunaan kurva ellips dicetuskan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985. [13]



Gambar 2.1. Elliptic Curve Cryptography

Adapun untuk membuat Elliptic Curve Cryptosystem diperlukan penjumlahan dua titik pada kurva Elips E(Fp) yang menghasilkan titik ke tiga pada kurva elips. Aturan ini dapat dijelaskan secara geometris sebagai berikut :

$$y^2 = x^3 + ax + b \text{ + c dengan } 4a^3 + 27b^2 \neq 0 \quad (1)$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \\ \frac{3x_1^2 + a}{2y_1} \end{cases} \quad (2)$$

Diberikan E oleh $y^2 = x^3 + bx + c$ dan P1 = (x_1, y_1) dan P2 = (x_2, y_2) P1 + P2 = P3 (x_3, y_3)

Maka diperoleh (x3, y3) sebagai berikut :

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 + x_3) - y_1 \quad (3)$$

Pada operasi perkalian titik, titik P pada kurva eliptik dikalikan dengan k (bilangan scalar) dengan menggunakan persamaan tersebut untuk mendapatkan titik Q pada kurva yang sama. Maka

$$kP = Q$$

adapun operasi perkaliannya ialah sebagai berikut :

a. Operasi pertambahan titik, yaitu menambahkan dua buah titik J dan K untuk mendapatkan titik L. Dengan demikian L = 2J.
b. Operasi penggandaan titik, yaitu menambahkan titik J dengan dirinya sendiri untuk mendapatkan titik L. Dengan demikian L = 2J.

2.2. SHA-256

SHA-256 adalah fungsi hash satu-arah yang dibuat oleh NIST (The National Institute of Standarts and Technology) pada tahun 2002. SHA-256 menghasilkan message digest dengan panjang 256 bit. SHA-256 menggunakan enam fungsi logika, tiap fungsi beroperasi pada 32-bit. Langkah-langkah dalam algoritma ini ialah.

1. Penambahan bit pengganjal dan nilai panjang.

   Pesan biner yang diproses akan ditambahkan dengan angka 1 sampai dengan pesan diikuti K Zeroes sehingga panjang pesan yang didapat ialah 64 bit sebagian kecil dari 512 bits.

   Pesan yang ditambahkan akan diuraikan dengan A^ 512-bit blocks, menjadi, M_0 ,M_1,... M_N yang dimana masing masing akan menjadi 16 blok 32-bit, yang akan dinamakan kembali menjadi M_i^1, $M_i^{1,}$ ...., $M_i^{15.}$

2. Inisialisasi fungsi hash

   Sebelum mengawali fungsi hash komputasi diatur. Adapun tabel dibawah ini merupakan code inisialisasi SHA-256[13]

   a := 0x6a09e667
   b := 0xbb67ae85
   c := 0x3c6ef372
   d := 0xa54ff53a
   e := 0x510e527f
   f := 0x9b05688c
   g := 0x1f83d9ab
   h := 0x5be0cd19

Fungsi dari SHA-256 akan di operasikan dalam huruf 32-bit, yang dimana akan menambah pemanjangan pesan, adapun fungsi logikanya ialah :

$$Ch\ (x,y,z) = (x \wedge y) \oplus (x \wedge z) \quad (4)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (5)$$

$$\sum_0(x) = ROTR^2(x) \oplus ROTR^{12} \oplus ROTR^{22}(x) \quad (6)$$

$$\sum_1(x) = ROTR^6(x) \oplus ROTR^{11} \oplus ROTR^{25}(x) \quad (7)$$

$$\sigma_0 = ROTR^7(x) \oplus ROTR^{18} \oplus SHR^3(x) \quad (8)$$

$$\sigma_1 = ROTR^{17}(x) \oplus ROTR^{19} \oplus SHR^{10}(x) \quad (9)$$

Selain fungsi yang diatas SHA-256 menggunakan nilai konstanta K [0..63] yang dijelaskan pada tabel berikut ini[19].

Tabel 2.1. Nilai konstanta SHA-256

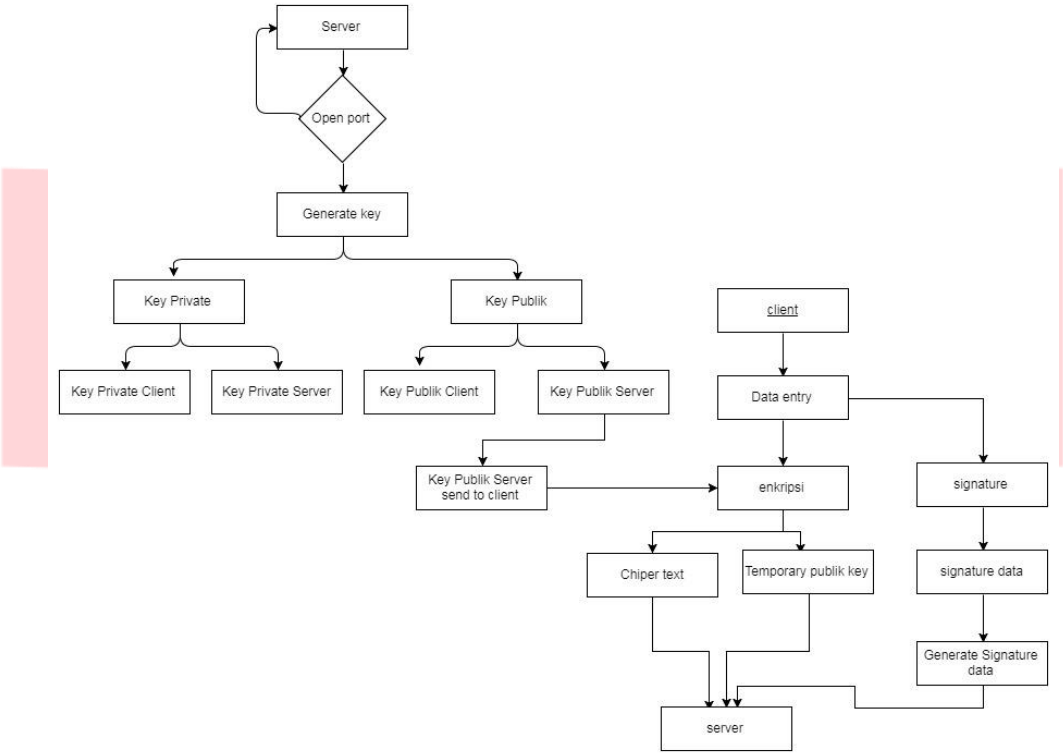
428a2f98	71374491	bScOfbcf	e9b5dba5
3956c25b	59flllfl	923f82a4	ablc5ed5
d807aa98	12835b01	243185be	550c7dc3

72be5d74	80deblfe	9bdc06a7	cl9bfl74
e49b69cl	efbe4786	0fcl9dc6	240calcc
2de92c6f	4a7484aa	5cb0a9dc	76f988da
98365152	a831c66d	b00327c8	bf597fc7
c6e00bf3	d5a79147	706ca6351	14292967
27b70a85	2elb2138	4d2c6dfc	53380dl3
650a7354	766a0abb	81c2c92e	92722c85
a2bfe8al	a81a664b	c24b8b70	c76c51a3
dl92e819	d6990624	f40e3585	106aa070
19a4cll6	Ie376c08	2748774c	34b0bcb5
391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208
90befffa	a4506ceb	bef9a3f7	c67178f2

3. Pembahasan

3.1. Deskripsi Sistem

Algoritma ECC sebagai sertifikat digital (*Digital Certificate)* untuk membuat suatu sistem itu dipercaya oleh user ketika melakukan transaksi di aplikasi tersebut. Sedangkan SHA-256 sebagai tanda tangan digital (*Digital Signature*) yang berfungsi sebagai penvalidasi bahwa data yang diterima oleh server tersebut asli berasal dari user. Dan pada user diberikannya ECC dan SHA-256 pada program tersebut sehingga data yang telah di enkripsi dapat langsung di dekripsi oleh server untuk mengecek keaslian data tersebut.

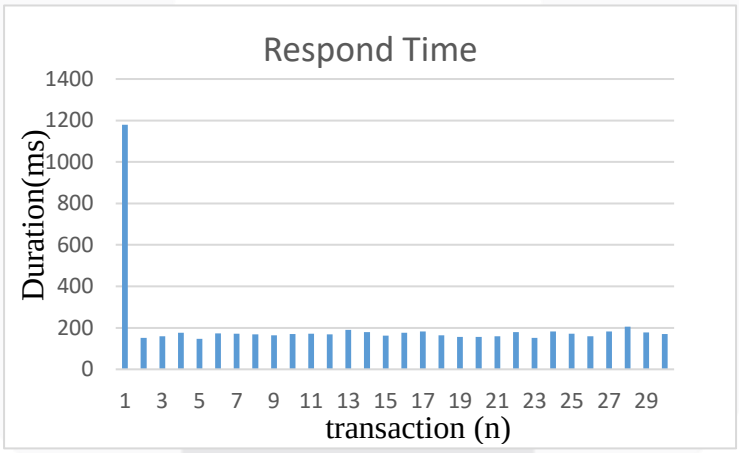


Gambar 3.1. Flowchart Skema Sistem

3.2. Pengujian dan Analisa

3.2.1. Pengujian Time Response

Time Response adalah waktu yang dibutuhkan ketika client mengirimkan permintaan data ke server dan pertukaran kunci yang dilakukan oleh kedua pihak. Pada tugas akhir ini pengukuran Time Response dimulai dari keadaan data yang telah di submit yang memiliki panjang kunci dan dengan waktu yang dibutuhkan.



Gambar 4.1. Diagram Time Response

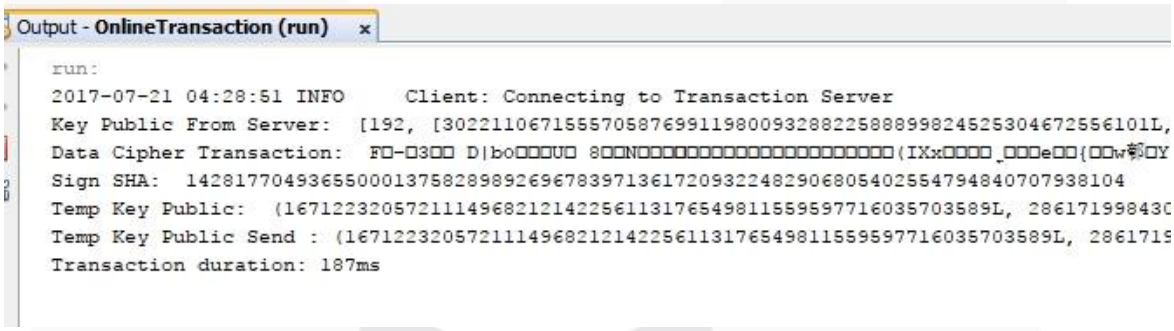
Gambar diatas menunjukkan bahwa jumlah penginputan data yang dilakukan sebanyak 30kali dengan menggunakan kunci ECC dan SHA-256 ebih lambat dikarenakan panjang kunci SHA-256 lebih panjang yaitu 256 bit dengan time respond rata-rata 203.6 milisecond. Sedangkan dengan pengujian MD5 144.13 milisecond. yang dilakukan oleh rekan saya yaitu dengan rata-rata jauh lebih cepat dibanding dengan SHA-256 Meski hanya memiliki kunci terpendek. Belum tentu bisa membuktikan bahwa semakin pendek kuncinya maka semakin cepat waktu enkripsi dan dekripsi. Dan menurut analisa saya, Jika Kunci lebih panjang maka akan lebih secure dikarenakan makin susah untuk melakukan pembukaan kunci tanpa penggunaan kunci yang kita ketahui. Dan SHA-256 ini fungsi hash terbaru, meski sudah ada SHA-512 akan tetapi SHA-256 saat ini sudah banyak digunakan di situs marketing.

3.3. Analisa Tingkat Keamanan.

Setelah melakukan pengujian dengan time respond perbandingan dengan MD-5, tidak luputnya kita mendapatkan analisa dari berbagai jenis serangan yang dapat dilakukan terhadap aplikasi yang dibuat. Adapun jenis-jenis serangan yang dapat dilakukan ialah Spoofing, Sniffing, dan Man in Middle Attack.

3.3.1. Spoofing

Seperti yang kita ketahui Spoofing ialah jenis serangan yang dilakukan oleh attacker atau seseorang yang tidak bertanggung jawab untuk mencoba menelusuri atau menyeludup sebuah jaringan dengan mengirimkan pemberitahuan bahwa paket ini sangat dipercaya untuk melakukan hosting pengisian data dll. Untuk menghindari serangan spoofing ini kita melakukan dengan pembuatan filter yang berfungsi untuk menghadang si attacker untuk melakukan penelusupan dengan cara enkripsi. Adapun untuk kita mengetahui tingkat keamanannya telah ditampilkan dalam gambar dibawah ini, detail lebih lanjut berada di lampiran



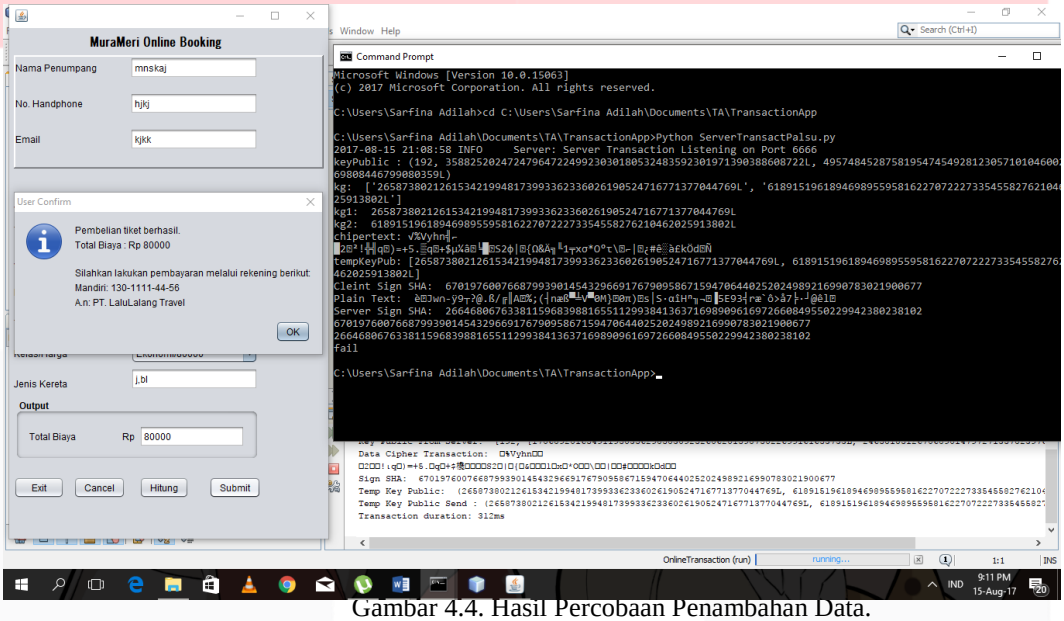
Gambar 3.2. Tampilan Key Cipher Text, Temporary Private Key dan Signature

3.3.2. Sniffiing

Adapun ancaman selanjutnya untuk *client* dan *server* adalah Serangan *Sniffing*. *Sniffing* adalah serangan yang bermaksud menyabotase atau mengambil alih data yang dimiliki kemudian mengubah datanya dan diteruskan kepada data tujuan. Adapun cara-caranya agar tidak terjadi kasus ini ialah dengan membuat *fake server* yang bertujuan untuk melindungi server yang aslinya.

```
print "Sign SHA: ", signSHA
#print "Sign MD5: ", signMD5
print "Temp Key Public: ", kg
kg_send = kg[0],kg[1]
kg_send = ''.join(str(kg_send))
print "Temp Key Public Send :", kg_send
```

Gambar 4.3. Percobaan Penambahan Data.



Gambar 4.4. Hasil Percobaan Penambahan Data.

3.3.3. Man in Middle Attack

Serangan yang terjadi pada tipe ini adalah adanya pencobaan pengambilan atau perubahan data pada saat terjadi pertukaran data antara *client* dengan *server* . Maka dengan adanya Proses Enkripsi dan pertukaran kunci ini. Diharapkan *Attacker* tidak dapat menyerang proses yang sedang terjadi. Dan apabila *attacker* tetap ingin menyerang dengan menyamar, *Server* tetap meminta permintaan *Private key* / *Signature Key* yang dimana itu hanya dimiliki oleh *Server* Tersebut. Sehingga *Attacker* tidak dapat melakukan penyerangan.

4. Kesimpulan dan Saran

4.1. Kesimpulan

Setelah dilakukan beberapa pengujian dan analisa, dapat kita simpulkan beberapa hal yaitu:

1. Penggunaan Algoritma ECC dan SHA-256 untuk penggunaan suatu aplikasi diperlukan untuk penggunaan kunci yang lebih panjang pada proses transaksi online.
2. Pengujian dan Analisis yang dilakukan ialah dengan menguji Respond Time yang merupakan parameter keberhasikan dari pengujian sistem tersebut.
3. Respond time yang telah diujicoba oleh aplikasi dengan melalui Proses Enkripsi, Dekripsi dan Signature Data dengan rata-rata 203.6 ms dalam 30 kali percobaan.

4. Adapun jenis jenis serangan yang sudah dianalisa ialah Spoofing, Sniffing, dan Man in Middle Attack. Spoofing dapat diatasi dengan penggunaan fungsi enkripsi dan dekripsi pada suatu proses. Aplikasi ini dapat mengelabui Sniffing dikarenakan memiliki Fake Server yang dimana sudah membuat fungsi ghost sebagai pelindung dari serangan dan Man In The Middle Attack dapat diatasi dengan penggunaan SHA-256 Sebagai Signature Key dalam proses authentikasi data yang digunakan untuk memeriksa keaslian data tersebut.

DAFTAR PUSTAKA

[1] Budi Raharjo, 2015 ,**Mudah Belajar Python untuk Aplikasi Desktop dan Web**, Bandung :Penerbit Informatika

[2] Forouzan, A. Behrouz., 2008, ***Cryptography and Network Security***, Mexico : The McGraw-Hill Companies, Inc.

[3] Herbert Schildt, 2015, **"The Complete Reference, Seventh Edition (Osborne Complete Reference Series),"** Java Programming.

[4] Komputer, Wahana., 2010, ***The Best Encryption Tools***, Jakarta : Elex Media Komputindo.

[5] Mattes, Daniel., Starkey Chad, 2013, **Verification of Online Transactions**, California : Jumio Inc.,

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

[7] Ponomarev, Oleg., Khurri, Andrey., Gurtov, Andrei, 2010, ***Elliptic Curve Cryptography (ECC) for Host Identity Protocol (HIP),*** Finland : Helnsinki Institute for Information Technology HIT/ Aalto University.

[8] Rodriguez-Henriquez, Fransisco., Saqib, N.A., **Koc**, Cetin Kaya., 2006, ***Cryptographic Algorithms on Reconfigurable Hardware***, United Stated of America : Springer Science+Business Media, LLC.

[9] Yusmantoro, Sandi., Hermansyah, Edy., Efendi, Rusdi., 2014, **Rancang Bangun Aplikasi Pengamanan Keaslian Surat Izin Tempat Usaha menggunakan Algoritma Elgamal Dan Secure Hash Algorithm 256 Studi Kasus : Badan Pelayanan Perizinan Terpadu (BPPT) Kota Bengkulu**, Bengkulu : Universitas Bengkulu.

[10] Warno, 2012, "**Pembelajaran Pemrograman Bahasa Java dan Arti Keyword,**" Jurnal Komputer, vol. 8, no. 1.