Jurnal Ilmu Komputer & Agri-Informatika
Volume 8 Nomor 1 halaman 69 – 78
e ISSN: 2654-9735, p ISSN: 2089-6026

Tersedia secara *online* di:
http://journal.ipb.ac.id/index.

Pengamanan *Internet of Things* untuk Tanda Tangan Digital Menggunakan Algoritme Elgamal Signature Scheme

Security of The Internet of Things (IoT) for Digital Signature Using the Elgamal Signature Scheme Algorithm

SELFI QISTHINA[1], SHELVIE NIDYA NEYMAN[1]*

Abstrak

Internet of Things (IoT) memungkinkan suatu objek menghasilkan data dan bertukar data. Pengaplikasian IoT menggunakan mikrokontroler seperti Arduino belum memiliki fitur untuk menjaga keamanan data di dalamnya. Selain itu, Arduino memiliki kapabilitas komputasi terbatas. Oleh karena itu, perlu diterapkan kriptografi dengan algoritme yang memiliki komputasi rendah pada Arduino untuk menjaga keamanan data. Penjagaan keamanan data terutama pada keaslian asal data, dilakukan dengan menggunakan tanda tangan digital. Penerapan tanda tangan digital dapat dilakukan salah satunya dengan algoritme Elgamal *signature scheme*. Penerapan tanda tangan digital menggunakan algoritme Elgamal *signature scheme* berhasil diterapkan pada perangkat Arduino Uno untuk melakukan tanda tangan digital dan verifikasi. Kinerja algoritme Elgamal signa*ture scheme* dilihat dari analisis waktu eksekusi dan analisis keamanan algoritme. Waktu eksekusi proses tanda tangan digital membutuhkan waktu lebih lama dibandingkan dengan waktu eksekusi proses verifikasi. Algoritme Elgamal *signature scheme* membutuhkan waktu dua kali lebih lama karena banyaknya perhitungan sistematis pada perangkat Arduino Uno. Proses verifikasi terbukti gagal jika ada perubahan data dan pasangan tanda tangan digital.

Kata Kunci**:** Arduino, Elgamal *signature scheme, internet of things,* kriptografi, tanda tangan digital.

Abstract

Internet of Things (IoT) allows an object to generate data and exchange data. The application of IoT using microcontrollers such as Arduino still has no data security features in it. In addition, Arduino has limited computing capabilities. Therefore, cryptography with low computing capabilities needs to be applied on Arduino for data security. The authenticity of the origin of data on IoT can be maintained by applying digital signatures. The application of digital signatures can be done with the Elgamal signature scheme algorithm. The application of digital signatures using the Elgamal signature scheme algorithm is successfully applied to the Arduino Uno device to do signatures and verification. The performance of the algorithm is seen from the analysis of execution time and algorithmic security. The execution time of the signature process takes longer than the verification process. The Elgamal signature scheme algorithm takes twice longer because of many systematic calculations on the Arduino Uno device. The verification process has proven to fail if there are changes to the data and signature pairs.

*Keywords**:** Arduino, cryptography, digital signature, Elgamal signature scheme, internet of things.*

PENDAHULUAN

Internet of things (IoT) tengah menjadi perbincangan dalam dunia teknologi pada saat ini. Istilah IoT umumnya mengacu pada sebuah skenario suatu jaringan internet, kemampuan konektivitas dan komputasi berada dalam sebuah objek yang memungkinkan objek tersebut untuk menghasilkan data, bertukar data, dan mengambil data dengan sedikit campur tangan manusia (Rose *et al.* 2015). Pada dasarnya IoT merupakan konstruksi yang saling menghubungkan perangkat umum satu sama lain. Perangkat umum dapat berupa jam tangan, televisi, termostat, mobil, dan lampu. Selain perangkat umum yang disebutkan sebelumnya,

[1]Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor.
*Penulis Korespondensi: Surel: shelvie.neyman@gmail.com

pengaplikasian IoT juga biasa digunakan pada mikrokontroler. Salah satu contoh mikrokontroler yang umum digunkan dalam pengaplikasian IoT adalah Arduino. Pengaplikasian IoT pada mikrokontroler ini dapat digunakan sebagai sarana pertukaran data atau pengiriman data.

Salah satu contoh pengaplikasiannya adalah sistem *location based* perangkat berdaya komputasi rendah dengan Arduino. Sistem ini bekerja dengan mengirimkan data berupa *longitude, latitude,* dan *internet protocol* (IP) *address* dari perangkat mikrokontroler ke suatu *server*. Pengaplikasian dengan sistem tersebut perlu memiliki fitur keamanan di dalamnya. Keamanan yang dimaksud dapat berupa keamanan saat melakukan pengiriman data antar alat elektronik. Jika pada *client server* keamanan proses pengiriman data dilindungi dengan *hypertext transfer protocol secure* (*HTTPS*), sebaliknya pada IoT belum terdapat keamanan saat proses transaksi data. Keamanan proses transaksi data pada IoT merupakan hal yang penting, bukan hanya pada saat data dikirimkan tetapi juga bagaimana data tidak diubah oleh seseorang atau pihak ketiga sehingga data tersebut bersifat asli dan untuk mengetahui keaslian asal data tersebut.

Kriptografi perlu diterapkan pada pengaplikasian IoT menggunakan mikrokontroler untuk menjaga keamanan proses transaksi data dan menjaga keaslian asal suatu data. Penerapan kriptografi pada mikrokontroler juga harus ringan dan dapat berjalan pada mikrokontroler terutama mikrokontroler Arduino. Pada Arduino, kemampuan komputasi bersifat terbatas namun kebanyakan algoritme komputasi keamanan yang ada memiliki komputasi yang tinggi. Oleh karena itu, protokol kriptografi yang diterapkan harus memiliki algoritme yang efisien dan kemampuan komputasi yang rendah. Salah satu protokol kriptografi yang dapat diterapkan pada mikrokontroler untuk mengetahui keaslian asal data adalah tanda tangan digital. Tanda tangan digital dapat mengidentifikasi keaslian data, kebenaran sumber data, dan mencegah pihak yang mengirimkan data melakukan penyangkalan. Menurut Stallings (2011), tanda tangan digital harus mampu melakukan verifikasi pemilik tanda tangan, mampu melakukan autentikasi pemilik pesan, dan dapat diverifikasi oleh pihak ke tiga jika terjadi perselisihan.

Tanda tangan digital dapat dibuat dengan beberapa algoritme, salah satunya adalah algoritme Elgamal *signature scheme*. Banyak penelitian yang telah dilakukan terkait algoritme tanda tangan digital Elgamal. Penelitian Haraty *et al.* (2006) menjelaskan tentang keamanan, efisiensi, dan keandalan Elgamal serta modifikasinya saat menghadapi serangan. Beberapa penelitian lainnya seperti penelitian Jarusombat dan Kittitornkun (2006) melakukan pengembangan tanda tangan digital berbasis lokasi pada perangkat bergerak yang memiliki kapabilitas komputasi rendah dan daya baterai pendek. Pada penelitian ini algoritme yang digunakan adalah algoritme Elgamal *signature scheme* yang diterapkan pada mikrokontroler.

METODE

Ruang Lingkup Penelitian

Lingkup dari penelitian ini, yaitu perangkat yang digunakan berupa mikrokontroller Arduino Uno dan proses tanda tangan digital dilakukan antara Arduino Uno dan PC, serta antara PC dan Arduino Uno. Proses tanda tangan digital ini dilakukan pada mikrokontroler Arduino Uno menggunakan Kabel USB.

Data Penelitian

Data berupa kunci publik yang dihasilkan pada proses autentikasi entitas. Data merupakan bilangan heksadesimal yang berukuran maksimal 32-bit.

Lingkungan Pengembangan

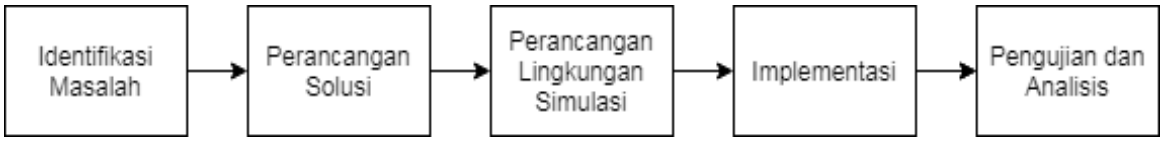
Perangkat keras yang digunakan untuk penelitian ini adalah:

1 PC pribadi dengan spesifikasi sebagai berikut:

a Processor Intel® Core™ i3 i3-3217U 1.80 GHz.
b RAM 4 GB.
c *Hard drive* 500GB.

2 Arduino Uno sebagai mikrokontroler dengan spesifikasi sebagai berikut:
a *Clock speed* 16MHz.
b SRAM 2 KB.
c *Flash memory* 32 KB (0.5 untuk *bootloader*).

Tahapan Penelitian



Gambar 1 Tahapan penelitian.

Identifikasi Masalah

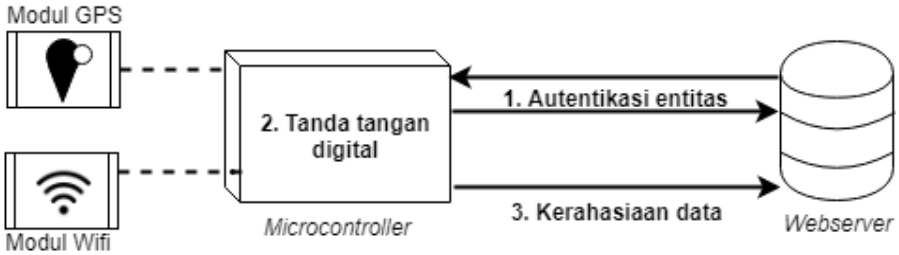
Tahap ini dilakukan untuk mengetahui permasalahan apa yang ada pada sistem berbasis IoT saat ini dan cara mengatasinya. Kegiatan-kegiatan yang dilakukan pada tahap identifikasi masalah adalah sebagai berikut :

1. Mengumpulkan dan memperdalam materi-materi yang berhubungan dengan tanda tangan digital.
2. Mengumpulkan dan memperdalam materi-materi yang berhubungan dengan implementasi Elgamal pada sistem *location based* dengan Arduino.
3. Memperdalam materi-materi yang berhubungan dengan cara kerja dari Arduino.

Dari tahapan ini nantinya berguna untuk memahami sistem yang akan dibuat secara lebih mendalam.

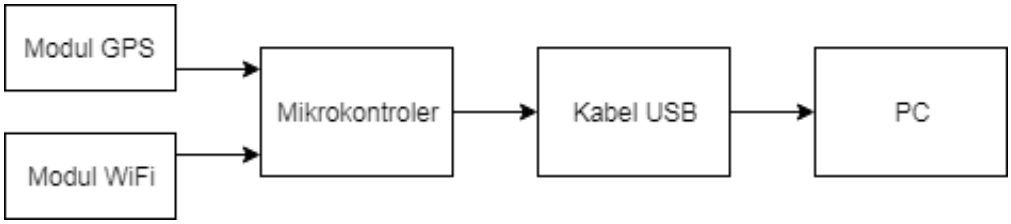
Perancangan Lingkungan Simulasi

Sistem dirancang untuk menyediakan tiga layanan keamanan. Tiga layanan keamanan tersebut adalah autentikasi entitas, tanda tangan digital, dan kerahasiaan data. Lingkungan simulasi pada penelitian ini dibangun pada sistem keamanan IoT yang memiliki tiga tahapan utama seperti pada Gambar 2. Tahapan pertama yang dilakukan adalah Arduino melakukan inisialisasi ke *server* untuk pertukaran kunci atau autentikasi entitas. Tahapan kedua, yaitu pembuatan tanda tangan digital pada data lokasi mikrokontroler untuk autentikasi asal data. Tahapan ketiga, yaitu Arduino melakukan enkripsi pada data sebelum dikirimkan ke *server* untuk menjaga kerahasian data.



Gambar 2 Lingkungan sistem simulasi keseluruhan.

Penelitian ini difokuskan pada tahapan kedua yang merupakan pembangkitan tanda tangan digital untuk menjaga keaslian asal data. Data berupa kunci publik yang dihasilkan dari proses autentikasi entitas. Data tersebut lalu akan dikirimkan ke PC melalui kabel USB seperti pada Gambar 3. Proses pembuatan tanda tangan digital ini akan dilakukan setelah tahapan pertukaran autentikasi entitas telah berhasil.



Gambar 3 Blok diagram lingkungan simulasi.

Implementasi

Proses implementasi penelitian memiliki tiga tahapan lagi di dalamnya, yaitu pembangkitan kunci, pembangkitan tanda tangan digital, dan verifikasi tanda tangan digital. Tahapan implementasi ini akan dilakukan menggunakan algoritme Elgamal *signature scheme*.

a. Pembangkitan Kunci

Tahap ini dilakukan untuk membangkitkan kunci yang ada pada tanda tangan digital. Terdapat dua kunci yang akan dihasilkan pada tahap ini, yaitu kunci publik dan kunci *private*. Hal pertama yang dilakukan adalah sistem akan membangkitkan kunci publik dan parameter Elgamal, yaitu parameter bilangan acak prima (p), bilangan acak (g), dan bilangan acak (k). Kunci *private* akan bisa dibangkitkan oleh sistem setelah mendapatkan kunci publik dengan melakukan perhitungan modulus terhadap kunci publik dan parameter yang sudah dibangkitkan sebelumnya.

b. Pembangkitan Tanda Tangan Digital

Tahapan ini dilakukan setelah pembangkitan kunci telah berhasil dan menghasilkan kunci publik dan kunci *private*. Pada tahap ini pesan yang akan ditandatangani merupakan kunci publik dihasilkan dari proses autentikasi entitas. Sistem akan menandatangani data tersebut menggunakan parameter yang telah dimiliki dan menghasilkan sebuah data baru yang sudah memiliki tanda tangan digital.

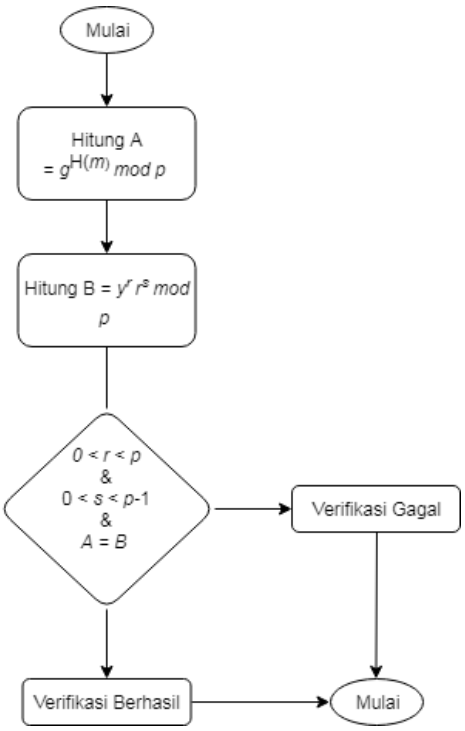
c. Verifikasi Tanda Tangan Digital

Tahapan verifikasi tanda tangan digital dapat dilihat pada Gambar 4. Sistem akan melakukan verifikasi tanda tangan digital dengan cara membandingkan dua buah proses komputasi. Keaslian dari tanda tangan digital tersebut akan terlihat dari hasil perbandingan dua proses komputasi yang dilakukan. Tanda tangan digital bersifat asli atau verifikasi berhasil jika perbandingan dua buah proses menghasilkan nilai yang sama. Verifikasi tanda tangan digital akan gagal jika perbandingan dua buah proses menghasilkan nilai yang berbeda.

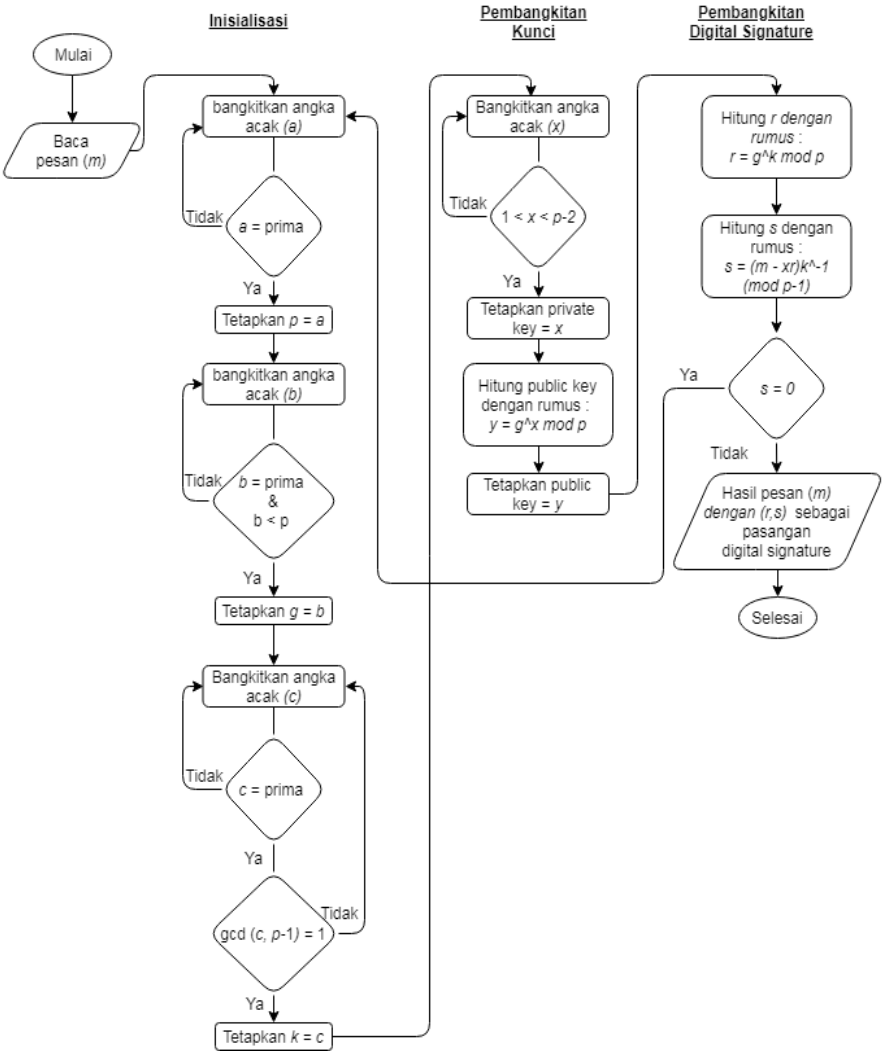
Keamanan tanda tangan digital menggunakan algoritme Elgamal *signature scheme* terletak pada kunci *private* yang dimiliki. Kunci *private* yang telah diketahui oleh pihak ketiga yang tidak berwenang, memungkinan keseluruhan algoritme tanda tangan digital ini dapat diakses oleh semua orang dan tidak memiliki keamanan. Selain keamanannya, komputasi algoritme yang dijalankan juga harus memiliki kapabilitas komputasi yang rendah sehingga dapat berjalan dengan baik pada mikrokontroler. Tahapan implementasi pembangkitan tanda tangan digital yang telah dijelaskan di atas dapat dilihat pada Gambar 5.

Pengujian dan Analisis

Terdapat dua pengujian yang dilakukan pada tahap ini, yaitu pengujian fungsional dan pengujian kinerja. Pengujian fungsional dilakukan untuk melihat sistem dapat berjalan dengan perhitungan komputasi yang rendah. Pengujian kinerja dilakukan untuk melihat ketahanan sistem terhadap kemungkinan serangan yang terjadi.



Gambar 4 Tahapan verfikasi tanda tangan digital.



Gambar 5 Tahapan pembangkitan tanda tangan digital.

HASIL DAN PEMBAHASAN

Implementasi Lingkungan Simulasi

Hasil dari tahapan ini merupakan sebuah sistem yang dapat melakukan pembangkitan tanda tangan digital, serta melakukan verifikasi tanda tangan digital pada sistem dan pada PC. Pada penelitian ini transmisi data dibatasi dengan menggunakan kabel USB seperti yang terlihat pada Gambar 6.

Gambar 6 Implementasi lingkungan simulasi.

Implementasi

a. *Pembangkitan Kunci*

Proses pembangkitan kunci dilakukan oleh pengirim. Kunci yang dibangkitkan berguna untuk memberikan tanda tangan digital pada data. Kunci yang dibangkitan terdapat dua jenis, yaitu kunci *private* dan kunci *public*. Berikut pada Tabel 1 adalah hasil dari proses pembangkitan kunci.

Tabel 1 Hasil pembangkitan kunci

Parameter	Nilai
p	61091
g	60383
k	60719
Kunci *private*	60647
Kunci *public*	35865

Berdasarkan Tabel 1, parameter bilangan acak prima (*p*), bilangan acak (*g*), dan bilangan acak (*k*) dibangkitkan terlebih dahulu sebelum pembangkitan kunci *private* dan kunci *public*. Parameter *p* yang dihasilkan berupa bilangan acak bersifat prima. Parameter *g* dan *k* yang dihasilkan merupakan bilang acak yang memiliki nilai lebih kecil dari parameter *p*. Kunci *public* yang dihasilkan merupakan perhitungan modulus dari parameter *g*, *p* dan kunci *private.*

b. Pembangkitan Tanda Tangan Digital

Proses pembangkitan tanda tangan digital dilakukan setelah mendapatkan kunci *private* dan kunci *public*. Data akan ditanda-tangani dengan pasangan tanda tangan digital (*r*, *s*) yang dihitung melalui rumus perhitungan oleh sistem. Berikut pada Tabel 2 adalah hasil dari pembangkitan tanda tangan digital.

Tabel 2 Hasil pembangkitan tanda tangan digital

Parameter	Nilai
Data (*m*)	2147483647
r	32914
s	25611

Berdasarkan Tabel 2, data (*m*) memiliki panjang bit sebesar 31-bit. Hasil yang diperoleh pada tahap ini berupa pasangan parameter *r* dan *s* yang merupakan tanda tangan digital dari *m*.

c. Verifikasi Tanda Tangan Digital

Verifikasi tanda tangan digital menghasilkan hasil perbandingan dari dua proses komputasi. Berikut pada Tabel 3 adalah hasil verifikasi keabsahan tanda tangan digital. Berdasarkan pada Tabel 3, perbandingan dari hasil dua proses tersebut memiliki nilai yang sama yang artinya verifikasi keabsahan tanda tangan digital berhasil dilakukan.

Tabel 3 Hasil verifikasi tanda tangan digital

Proses komputasi	Hasil
$g^m \ mod \ p$	16717
$kunci \ public^r \ r^s \ mod \ p$	16717

Pengujian dan Evaluasi

Analisis Kinerja Algoritme

Pada tahap ini memaparkan hasil analisis kinerja yang dilakukan pada Arduino Uno dan pada PC. Analisis kinerja yang dilakukan berupa analisis kinerja terhadap waktu eksekusi. Pengukuran kinerja algoritme diukur bedasarkan panjang bilangan acak prima (p) yang digunakan, panjang data (m) yang digunakan, dan lama waktu eksekusi algoritme.

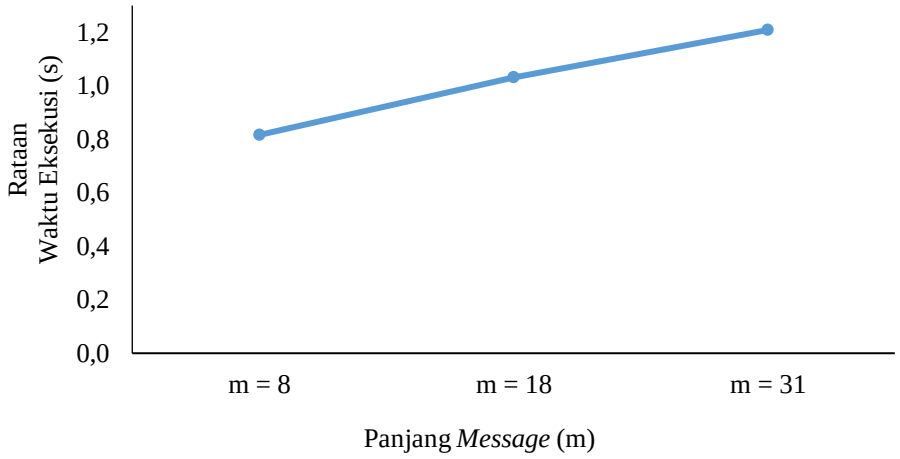
a. Analisis Kinerja Waktu pada Arduino Uno

Pada tahap ini memaparkan hasil analisis kinerja waktu pada Arduino Uno. Analisis kinerja waktu dibagi menjadi dua, yaitu analisis kinerja waktu pada saat pembangkitan tanda tangan digital dan analisis kinerja waktu pada saat verifikasi tanda tangan digital.

a.1. Analisis Kinerja Waktu untuk Pembangkitan Tanda Tangan Digital

Analisis kinerja waktu untuk pembangkitan tanda tangan digital dibagi menjadi dua, yaitu saat panjang p memiki nilai tetap dan panjang m memiliki perubahan nilai. Serta saat panjang m memiliki nilai tetap dan panjang p memiliki perubahan nilai.

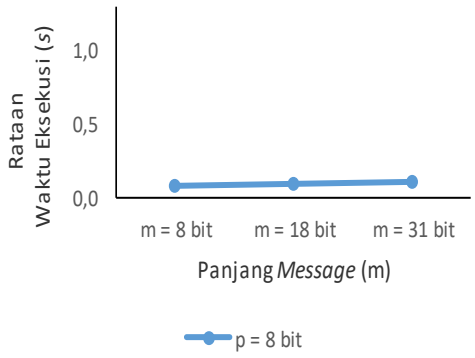
Pada Gambar 7 memaparkan analisis kinerja waktu algoritme dengan panjang p bernilai tetap yaitu 8 bit dan perubahan panjang m dengan nilai, 8 bit, 18 bit, dan 31 bit. Waktu eksekusi yang terlihat pada saat m = 8 bit sebesar 0.1 detik, saat m = 18 bit sebesar 0.11 detik, dan saat m = 31 bit sebesar 0.12 detik. Dari hasil waktu tersebut terilhat selisih perubahan waktu yang tidak besar, yaitu sebesar 0 .01 detik.



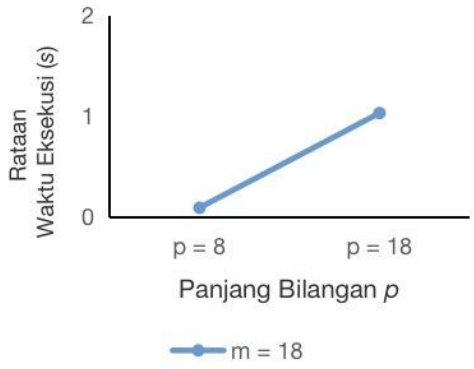
Gambar 7 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan p = 18 bit.

Gambar 8 memaparkan analisis kinerja algoritme dengan panjang *p* bernilai tetap yaitu 18 bit dan memiliki perubahan panjang *m* yang sama. Waktu eksekusi yang terlihat pada saat *m* = 8 bit sebesar 0.8 detik, saat *m* = 18 bit sebesar 1.02 detik, dan saat *m* = 31 bit sebesar 1.20 detik. Perubahan waktu eksekusi yang dihasilkan pada Gambar 5 lebih besar dibandingan dengan perubahan waktu eksekusi pada Gambar 4. Oleh karena itu, berdasarkan Gambar 7 dan 8, perubahan panjang *m* tidak memiliki pengaruh yang besar terhadap perubahan waktu eksekusi algoritme.

Pada Gambar 9 memaparkan analisis kinerja waktu algoritme dengan panjang *m* bernilai tetap yaitu 8 bit dan perubahan panjang *p* dengan nilai, 8 bit dan 18 bit. Waktu eksekusi yang terlihat pada saat *p* = 8 bit sebesar 0.1 detik, dan saat *p* = 18 bit sebesar 0.8 detik.

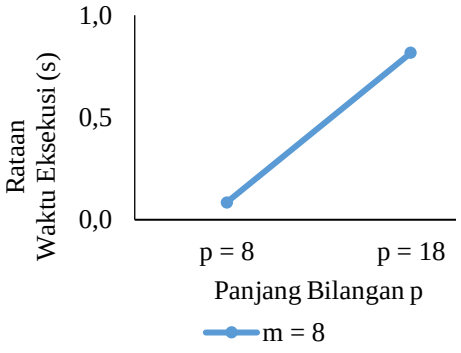


Gambar 8 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan *p* = 8 bit.

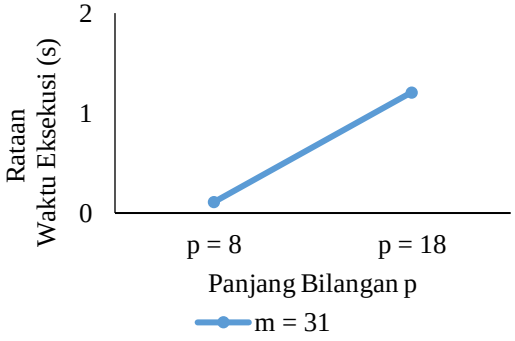


Gambar 9 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan m = 18 bit.

Pada Gambar 10 memaparkan analisis kinerja waktu algoritme dengan panjang *m* bernilai tetap yaitu 18 bit dan memiliki perubahan panjang *p* yang sama. Waktu eksekusi yang terlihat pada saat *p* = 8 bit sebesar 0.1 detik, dan saat *p* = 18 bit sebesar 1.2 detik. Pada Gambar 11 memaparkan analisis kinerja waktu algoritme dengan panjang *m* bernilai tetap yaitu 31 bit dan memiliki perubahan panjang *p* yang sama. Waktu eksekusi yang terlihat pada saat *p* = 8 bit sebesar 0.1 detik, dan saat *p* = 18 bit sebesar 1.3 detik. Hasil pada Gambar 9, 10, dan 11 memiliki selisih perubahan waktu yang besar. Sehingga kinerja waktu algoritme lebih dipengaruhi oleh panjang *p* daripada panjang *m*. Hal ini tersebut diperkuat pada hasil Gambar 9, 10, dan 11 memiliki selisih perubahan waktu yang besar dibandingkan dengan selisih perubahan waktu pada Gambar 7 dan 8.



Gambar 10 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan *m* = 8 bit.



Gambar 11 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan *m* = 31 bit.

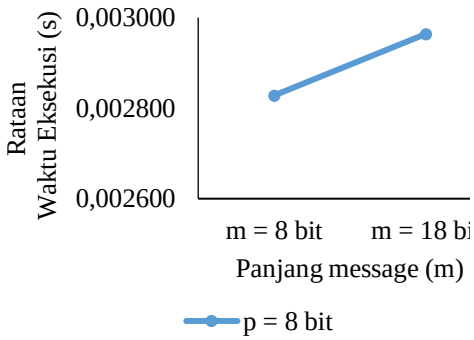
a.2. Analisis Kinerja Waktu untuk Verifikasi Tanda Tangan Digital

Analisis kinerja waktu untuk verifikasi tanda tangan digital dibagi menjadi dua, yaitu saat panjang *p* memiki nilai tetap dan panjang *m* memiliki perubahan nilai. Serta saat panjang *m* memiliki nilai tetap dan panjang *p* memiliki perubahan nilai.

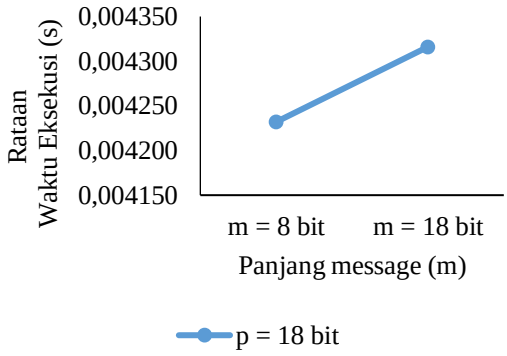
Pada Gambar 12 memaparkan hasil analisis kinerja waktu algoritme dengan panjang *p* bernilai tetap yaitu 8 bit dan perubahan panjang *m* dengan nilai, 8 bit dan 18 bit. Waktu eksekusi yang terlihat pada saat *m* = 8 bit sebesar 0.0028 detik dan saat *m* = 18 bit sebesar 0.0028 detik. Dari hasil waktu tersebut terilhat selisih perubahan waktu yang tidak besar.

Pada Gambar 13 memaparkan hasil analisis kinerja waktu algoritme dengan panjang *p* bernilai tetap yaitu 18 bit dan memiliki perubahan panjang *m* yang sama. Waktu eksekusi yang terlihat pada saat *m* = 8 bit sebesar 0.0042 detik dan saat *m* = 18 bit sebesar 0.0043 detik. Dari hasil waktu tersebut terlihat selisih perubahan waktu yang tidak besar sama seperti pada Gambar 12.

Penelitian ini membuktikan bahwa tidak seperti pada pembangkitan tanda tangan digital, perubahan panjang *p* pada saat verifikasi tanda tangan digital tidak berpengaruh terhadap waktu eksekusi.



Gambar 12 Grafik kinerja algoritme verifikasi tanda tangan digital pada Arduino dengan p = 8 bit.



Gambar 13 Grafik kinerja algoritme verifikasi tanda tangan digital pada Arduino dengan *p* = 18 bit.

Sama seperti halnya pada Arduino, analisis kinerja waktu algoritme juga dilakukan pada PC. Analisis kinerja waktu pada PC menunjukkan hasil dengan pola yang sama seperti pada Arduino.

Analisis Keamanan Algoritme

Analisis keamanan algoritme dilihat dari hasil verifikasi *signature* saat ada perubahan pada *m* dan pada pasangan tanda tangan digital (*r*,*s*). Pada Tabel 4 dan 5, perubahan pada *m* ataupun data pasangan tanda tangan digital (*r*,*s*) menyebabkan perbedaan nilai pada proses komputasi 1 dan proses komputasi 2. Perbedaan nilai yang dihasilkan menujukkan bahwa proses verifikasi telah gagal.

Panjang kunci *private* dihasilkan pada penelitian ini sebesar 18 bit. Jika dilakukan metode *brute force*, maka akan menghasilkan 2^{18} = 262 144 kemungkinan bilangan untuk menebak nilai dari kunci *private* yang dihasilkan. Jika *brute force* dilakukan pada PC penelitian ini membutuhkan waktu sekitar 262 detik atau sekitar empat menit. Idealnya untuk panjang kunci asimetris, minimal 2048 bit seperti pada RSA.

Tabel 4 Data awal dan data sesudah diubah

Parameter	Nilai		
	Data awal	Data pesan diubah	Data (*r*,*s*) diubah
Pesan (*m*)	2011684	2011876	2011684
r	20965	20965	20432
s	481	481	555

Tabel 5 Hasil verifikasi dengan data awal dan data setelah diubah

Proses verifikasi	Hasil		
	Data awal	Data pesan diubah	Data (*r*,*s*) diubah
Proses komputasi 1	29154	27223	27223
Proses komputasi 2	29154	29154	39187

SIMPULAN

Algoritme Elgamal *signature scheme* dapat diterapkan pada perangkat Arduino Uno untuk melakukan tanda tangan digital dan verifikasi tanda tangan digital. Panjang data yang diolah berpengaruh pada waktu eksekusi algoritme. Proses pembangkitan tanda tangan digital memiliki waktu eksekusi yang lebih besar dibandingkan dengan waktu eksekusi verifikasi tanda tangan digital. Waktu eksekusi algoritme pada Arduino Uno memiliki waktu yang lebih besar dua kali lipat dibandingkan dengan waktu eksekusi algoritme pada PC.

Perubahan nilai pada panjang data dan pasangan tanda tangan digital berpengaruh terhadap hasil verifikasi keabsahan tanda tangan digital. Verifikasi tanda tangan digital terbukti gagal jika terjadi perubahan pada panjang data dan pasangan tanda tangan digital.

DAFTAR PUSTAKA

Haraty RA, Elkassar, AN, Shebaro BM. 2006. A comparative study of Elgamal based digital signature algorithms. *Journal of Computational Methods in Sciences and Engineering.* 6:147–156. doi: 10.1109/WAC.2006.375953.

Jarusombat S , Kittitornkun S. 2006. Digital Signature on mobile devices based on location. Di dalam: *2006 International Symposium on Communications and Information Technologies*; 2006 Okt 18-20; Bangkok, Thailand. Bangkok (TH): IEEE. hlm 866-870.

Rose K, Chapin L, Eldridge S. 2015. *The Internet of Things: an Overview*. Geneva (CH): Internet Society.

Stallings W. 2011. *Cryptography and Network Security Principles and Practices*. Ed ke-5. New Jersey (US): Pearson.