PENGAMANAN *INTERNET OF THINGS* (IOT) UNTUK TANDA TANGAN DIGITAL MENGGUNAKAN ALGORITME ELGAMAL SIGNATURE SCHEME

SELFI QISTHINA

DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2019

PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya menyatakan bahwa skripsi berjudul Pengamanan *Internet of Things* (IoT) untuk Tanda Tangan Digital Menggunakan Algoritme Elgamal *Signature Scheme* adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, April 2019

Selfi Qisthina
NIM G64140059

ABSTRAK

SELFI QISTHINA. Pengamanan *Internet of Things* (IoT) untuk Tanda Tangan Digital Menggunakan Algoritme Elgamal *Signature Scheme*. Dibimbing oleh SHELVIE NIDYA NEYMAN.

Internet of Things (IoT) memungkinkan suatu objek menghasilkan data dan bertukar data. Pengaplikasian IoT menggunakan mikrokontroler seperti Arduino masih belum terdapat 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. Keamanan data terutama pada keaslian asal data, dengan melakukan tanda tangan digital. Penerapan tanda tangan digital dapat dilakukan salah satu contohnya 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 *signature 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

SELFI QISTHINA. Security of The Internet of Things (IoT) for Digital Signature Using the Elgamal *signature scheme* Algorithm. Supervised by SHELVIE NIDYA NEYMAN.

Internet of Things (IoT) allows an object to generate data and exchange data. The application of IoT using microcontroller such as Arduino still has no data security in it. In addition, Arduino has limited computing capabilities. Therefore, cryptography with low computing capabilities need 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 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 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.

PENGAMANAN *INTERNET OF THINGS* (IOT) UNTUK TANDA TANGAN DIGITAL MENGGUNAKAN ALGORITME ELGAMAL SIGNATURE SCHEME

SELFI QISTHINA

Skripsi
sebagai salah satu syarat untuk memperoleh gelar
Sarjana Ilmu Komputer
pada
Departemen Ilmu Komputer

DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2019

Penguji:
1 Dr Eng Heru Sukoco, SSi MT
2 Aunza Rahmad Akbar, SKomp MKom

Judul Skripsi: Pengamanan *Internet of Things* (IoT) untuk Tanda Tangan Digital Menggunakan Algoritme Elgamal *Signature Scheme*.
Nama : Selfi Qisthina
NIM : G64140059

Disetujui oleh

Dr Shelvie Nidya Neyman, SKom MSi
Pembimbing

Diketahui oleh

Prof Dr Ir I Egus Baono, MSi MKom
Ketua Departemen

Tanggal Lulus: 09 APR 2019

PRAKATA

Puji dan syukur penulis panjatkan kepada Allah *subhanahu wa ta'ala* atas segala karunia-Nya sehingga karya ilmiah ini berhasil diselesaikan. Tema yang dipilih dalam penelitian yang dilaksanakan sejak bulan Maret 2018 ini dengan judul Pengamanan *Internet of Things* (IoT) untuk Tanda Tangan Digital Menggunakan Algoritme Elgamal *Signature Scheme*.

Terima kasih penulis ucapkan kepada Ibu Dr Shelvie Nidya Neyman, SKom MKom selaku pembimbing, serta Bapak Dr Eng Heru Sukoco, SSi MT dan Bapak Auriza Rahmad Akbar, SKomp MKom yang telah banyak memberi saran. Ungkapan terima kasih juga disampaikan kepada ayah, ibu, serta seluruh keluarga, atas segala doa dan kasih sayangnya.

Semoga karya ilmiah ini bermanfaat.

Bogor, April 2019

Selfi Qisthina

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

PENDAHULUAN

Latar Belakang

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, 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 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 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 keamanan 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 penilitan ini algoritme yang digunakan adalah algoritme Elgamal *signature scheme* yang diterapkan pada mikrokontroler.

Perumusan Masalah

Masalah yang diangkat pada penelitian ini adalah perlunya adanya teknik tanda tangan digital pada perangkat IoT yang memiliki komputasi terbatas karena belum ada penelitian yang membahas tentang pembangkitan tanda tangan digital dan verfikasi tanda tangan digital pada perangkat IoT. Oleh karena itu, perlu diterapkan suatu algoritme tanda tangan digital yang memiliki kapabilitas komputasi rendah.

Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

1 Menerapkan tanda tangan digital pada perangkat IoT, yaitu mikrokontroller Arduino Uno menggunakan algoritme Elgamal *signature scheme*.
2 Mengukur kinerja algoritme tanda tangan digital yang diterapkan pada lingkungan IoT.

Manfaat Penelitian

Dapat memberikan pilihan yang lebih luas dalam pembentukan skema tanda tangan digital untuk perangkat IoT.

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.

TINJAUAN PUSTAKA

Mikrokontroler

Mikrokontroler dapat dikatakan sebagai alat elektronik digital yang memiliki masukan dan keluaran serta dapat dikendalikan dengan suatu program yang dapat ditulis dan dihapus secara khusus (Desnanjaya dan Supartha 2016).

Menurut Suhaeb *et al.* (2017), mikrokontroller yang ada di pasaran yaitu Intel 8048 dan 8051 (MCS51), Motorola 68HC11, Microchip PI, Hitachi H8, dan Atmel AVR. Mikrokontroler Atmel AVR merupakan *hardware* yang dimiliki oleh Arduino.

Arduino merupakan perangkat elektronik yang bersifat *open source* dan Arduino juga merupakan kombinasi dari *hardware*, bahasa pemrograman dan *integrated development environment* (IDE) (Kurniawan 2016). Arduino juga memiliki bahasa pemrograman sendiri, yaitu bahasa pemrogaman yang diadaptasi dari bahasa C (Mochtiarsa dan Supriadi 2016). Terdapat banyak jenis Arduino, beberapa diantaranya yaitu Arduino Uno, Arduino Mega, Arduino Nano dan Arduino Due. Salah satu Arduino yang banyak digunakan di kalangan umum, adalah Arduino Uno.

Tanda Tangan Digital

Tanda tangan digital adalah suatu mekanisme autentikasi yang memungkinkan pengirim pesan dapat menyisipkan sesuatu sebagai tanda tangan dari pesan tersebut (Stallings 2011). Tanda tangan digital memiliki karakteristik dan skema tersendiri. Menurut (Schneier 1996), karakteristik penandaan digital adalah sebagai berikut:

1 Penandaan merupakan bukti yang bersifat autentik.
2 Penandaan dapat dipastikan keasliannya atau tidak dapat dipalsukan.
3 Penandaan tidak dapat dipindah untuk digunakan kembali.
4 Dokumen yang telah ditandai tidak dapat diubah.
5 Penandaan tidak dapat disangkal.

Menurut Menezes *et al.* (1996) tanda tangan digital memiliki skema yang diklasifikasikan dalam dua kelompok, yaitu:

1 Skema tanda tangan digital dengan apendiks, skema ini membutuhkan pesan asli sebagai input untuk algoritme verifikasi. Contoh algoritme yang termasuk dalam skema kelompok ini adalah DSA, Elgamal, dan Schnorr.
2 Skema tanda tangan dengan pemulihan pesan, skema ini tidak membutuhkan pesan asli sebagai input untuk algoritme verifikasi. Dalam hal ini, pesan asli dibangkitkan dari tanda tangan itu sendiri. Contoh dari skema kelompok ini adalah RSA, Rabin, dan Nyberg-Rueppel.

Elgamal Signature Scheme

Algoritme Elgamal diciptakan oleh Taher Elgamal pada tahun 1985. Keamanan algoritme ini terletak penghitungan logaritma diskrit yang sulit. Elgamal signature scheme merupakan suatu algorime yang digunakan untuk proses autentikasi yang terdiri atas tiga proses, yaitu pembangkitan kunci, pembuatan tandatangan el gamal dan verifikasi tanda tangan (Elgamal 1985). Tiga proses autentikasi pada Elgamal signature scheme sebagai berikut.

Paramaeter pada Algoritme Elgamal signature scheme:

- p merupakan bilangan prima.
- g merupakan angka yang dipilih secara acak dengan $a < p$.

1 Pembangkitan kunci
- Tahap pembangkitan kunci akan dilakukan oleh *signer* atau penanda tangan.
- Kunci publik adalah *y* dan kunci *private* adalah *x*.
- Pilih secara acak nilai dari *x* dengan $1 < x < p - 2$.
- Lalu, hitung nilai kunci publik dengan rumus $y = g^x \bmod p$.

2 Pembangkitan tanda tangan
- Pilih angka acak *k* dimana $1 < k < p - 1$ dan gcd(k, $p - 1$) = 1.
- Hitung $r = g^k \pmod p$.
- Hitung $s = (H(m) - xr)k^{-1} \pmod{p-1}$, dengan $H(m)$ adalah pesan yang sudah di-*hash*.
- Jika $s = 0$, mulai dari awal lagi.
- Hasil dari tahapan ini adalah pasangan (r,s) merupakan tanda tangan digital dari m.

3 Verifikasi tanda tangan

Jika $0 < r < p$ dan $0 < s < p - 1$, serta $g^{H(m)} = y^r r^s \bmod$ tanda tangan bersifat asli dan pesan bersifat asli.

METODE

Data Penelitian

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

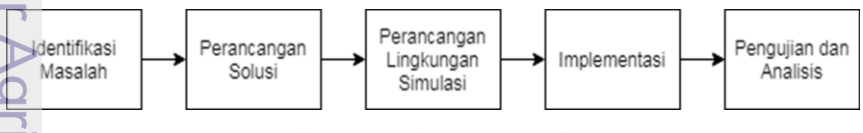
Lingkungan Pengembangan

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 :

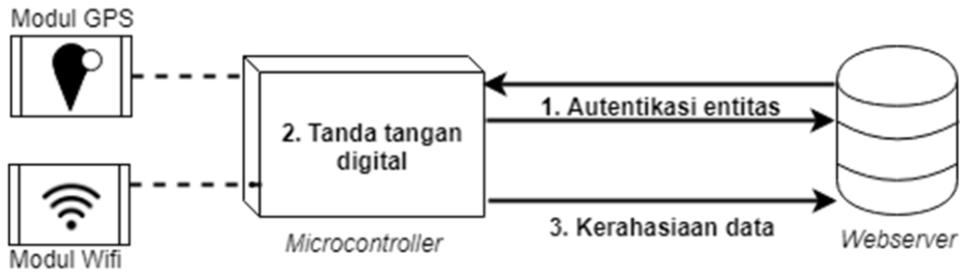
1 Mengumpulkan dan memperdalam materi-materi yang berhubungan dengan tanda tangan digital.
2 Mengumpulkan dan memperdalam materi-materi yang berhubungan dengan implementasi Elgamal pada sistem *location based* dengan Arduino.
3 Memperdalam materi-materi yang berhubungan dengan cara kerja dari Arduino.

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

Perancangan Solusi

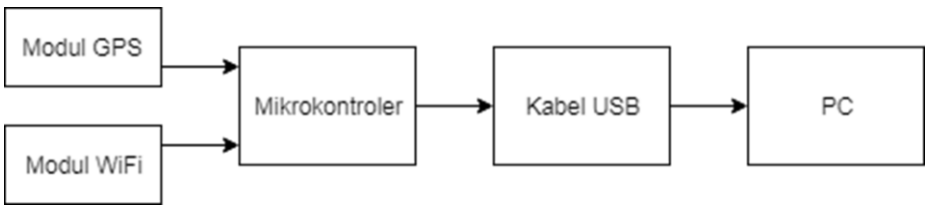
Tahap ini dilakukan untuk mengidentifikasi *hardware* dan *software* yang akan digunakan. *Hardware* dan *software* yang akan digunakan harus dapat bekerja pada lingkungan yang memiliki kapabilitas komputasi rendah.

Perancangan Lingkungan Simulasi



Gambar 2 Lingkungan sistem simulasi keseluruhan

Sistem dirancang untuk menyediakan tiga layanan keamanan. Tiga layanan keamanan tersebut di antaranya, yaitu autentikasi entitas, tanda tangan digital, dan kerahasiaan data.

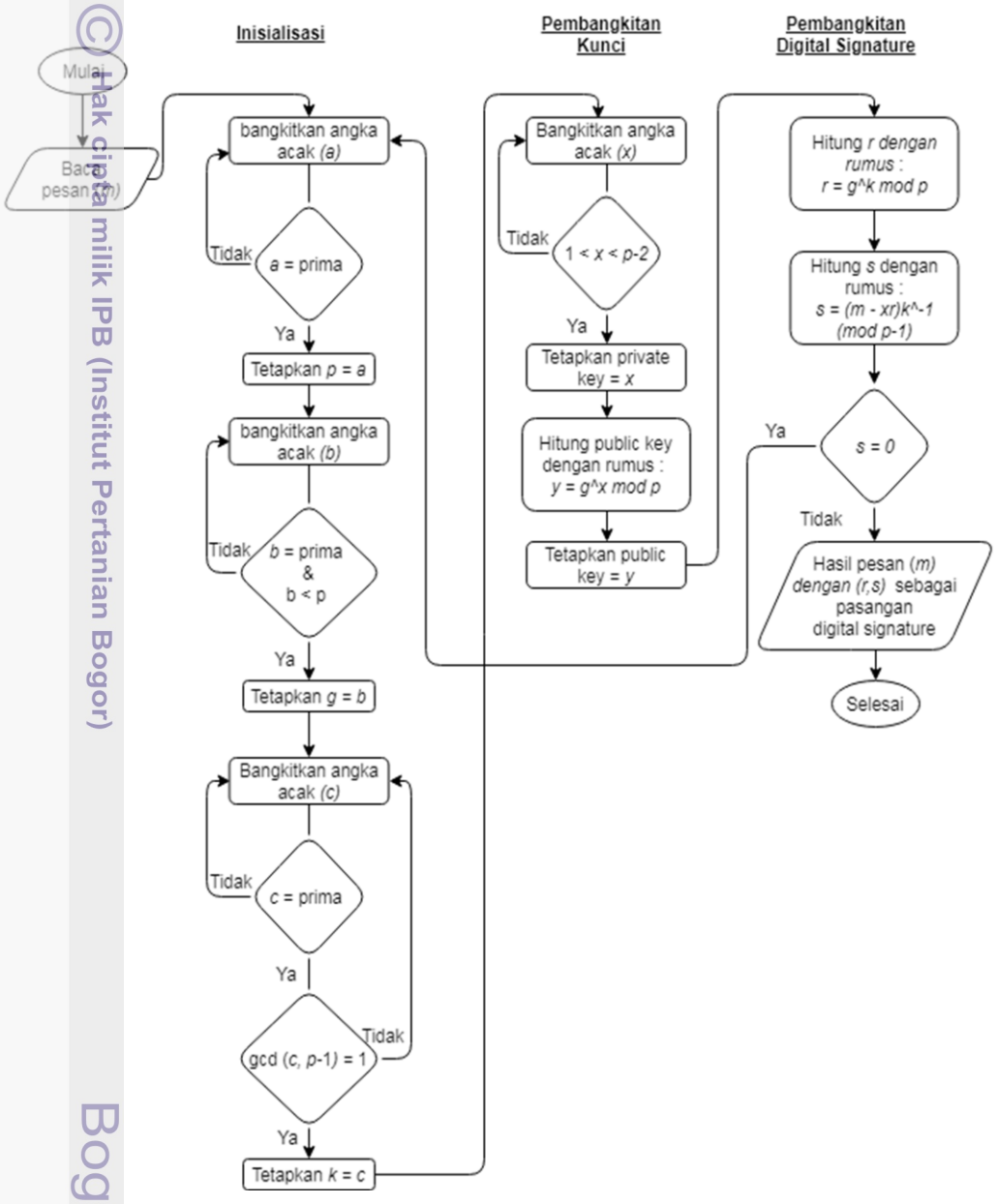


Gambar 3 Blok diagram lingkungan simulasi

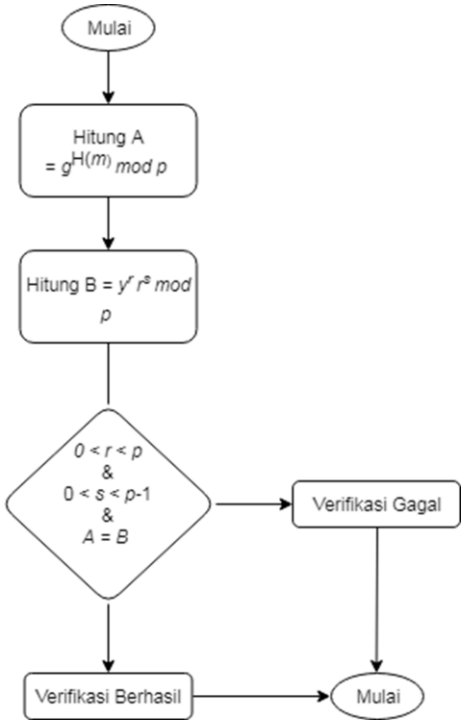
Lingkungan simulasi pada penilitian ini, dibangun pada sistem keamanan IoT yang memiliki tiga tahapan utama seperti pada Gambar 2, yaitu tahapan pertama yang dilakukan adalah Arduino melakukan inisialisasi ke server untuk pertukaran kunci atau autentikasi entitas. Tahapan kedua, yaitu pembuatan tanda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

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.

Implementasi



Gambar 4 Tahapan pembangkitan tanda tangan digital



Gambar 5 Tahapan verfikasi tanda tangan digital

Berdasarkan Gambar 4 dan Gambar 5, 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 *p*, *g*, dan *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

Tahap dilakukan setelah pembangkitan kunci telah berhasil dan menghasilkan kunci publik dan kunci *private*. Pada tahap ini pesan yang akan ditanda-tangani 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**

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.

Pengujian dan Evaluasi

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.

HASIL DAN PEMBAHASAN

Perancangan Solusi

Berdasarkan masalah yang telah disebutkan di atas, keamanan asal serta keutuhan asal data belum tersedia pada pengaplikasian IoT menggunakan Arduino Uno. Oleh karena itu, perangkat keras yang digunakan berupa Arduino Uno yang memiliki kapabilitas komputasi rendah dan banyak digunakan dilingkungan kerja. Berikut akan dijelaskan pada Tabel 1, fungsi dari masing-masing perangkat lunak yang digunakan.

Tabel 1 Perangkat keras penelitian

Perangkat keras	Fungsi
Arduino Uno	Mikrokontroler berkapasitas komputasi rendah yang banyak digunakan di lingkungan kerja
Kabel USB	Berfungsi sebagai penghubung perangkat Arduino Uno dengan PC
PC pribadi	Berfungsi sebagai alat untuk memprogram perangkat Arduino Uno dan melihat hasil penelitian

Perangkat lunak yang digunakan berupa Arduino IDE yang merupakan perangkat lunak yang dikhususkan untuk pemrogaman menggunakan mikrokontroler Arduino Uno. Arduino IDE Berfungsi untuk membuat program

yang akan diunggah ke perangkat Arduino Uno. Perangkat lainnya, yaitu Pycharm Edu yang berfungsi untuk membuat program yang akan digunakan pada server

Implementasi Lingkungan Simulasi

Gambar 6 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.

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 2 adalah hasil dari proses pembangkitan kunci.

Tabel 2 Hasil pembangkitan kunci

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

Berdasarkan Tabel 2, 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 3 adalah hasil dari pembangkitan tanda tangan digital.

Tabel 3 Hasil pembangkitan tanda tangan digital

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

Berdasarkan Tabel 3, 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 poses komputasi. Berikut pada Tabel 5 adalah hasil verifikasi keabsahan tanda tangan digital.

Tabel 4 Hasil verifikasi tanda tangan digital

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

Berdasarkan pada Tabel 4, perbandingan dari hasil dua proses tersebut memiliki nilai yang sama yang artinya verifikasi keabsahan tanda tangan digital berhasil dilakukan.

Pengujian dan Evaluasi

1 Analisis Kinerja Algoritme

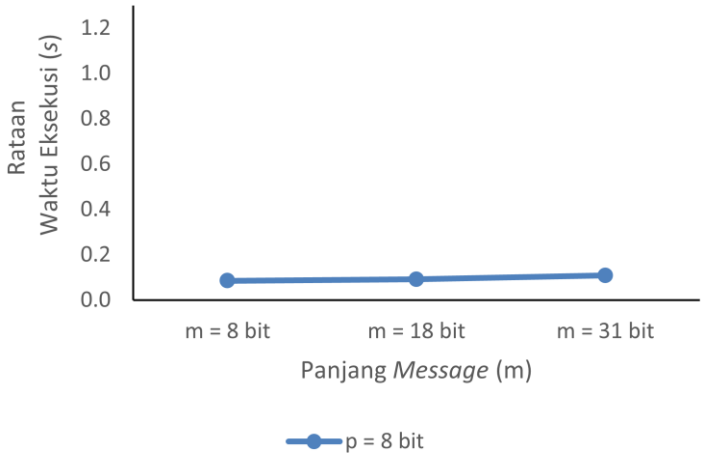
Pada tahap ini memaparkan hasil analisis kinerja yang dilakukan pada Arduino Uno dan pada PC. Analisis kinerja yang dilakukan berupa analisis kinerja terhadap waktu eksekusi. Pengukuran kinerja algoritme diukur bedasarkan panjang bilangan acak prima (*p*) yang digunakan, panjang data (*m*) yang digunakan,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

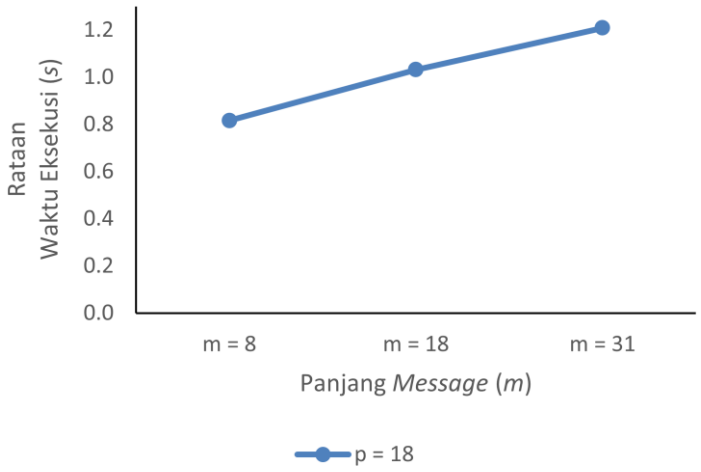
i. Analisis Kinerja Waktu untuk Pembangkitan Tanda Tangan Digital

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.



Gambar 7 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan *p* = 8 bit

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.

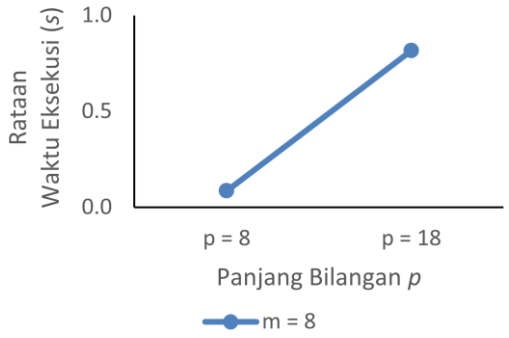


Gambar 8 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan *p* = 8 bit

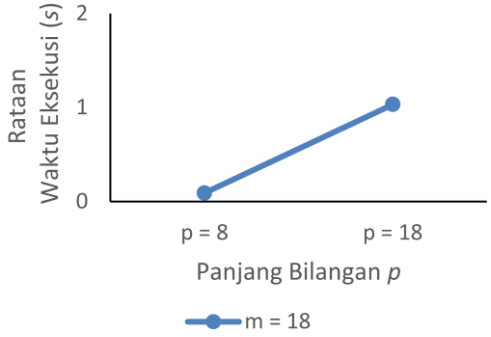
Pada Gambar 8 memaparkan analisis kinerja algoritme dengan panjang *p* bernilai tetap yaitu 18 bit dan memiliki perubahan panjang *m* yang sama. Waktu eksekusi yang terlihat pada saat *m* = 8 bit sebesar 0.8

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.



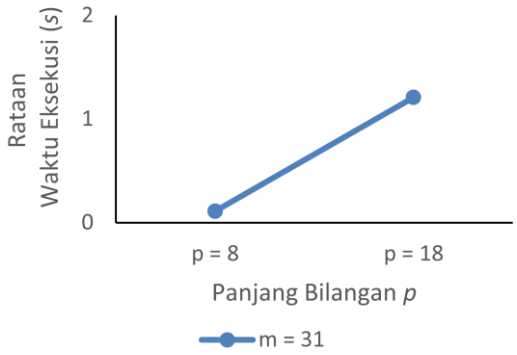
Gambar 9 Grafik kinerja algoritme pembangkitan tanda tangan digital pada Arduino dengan *m*



Gambar 10 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.



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

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.

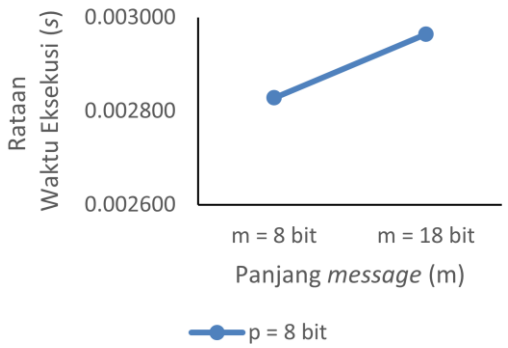
ii. Analisis Kinerja Waktu untuk Verifikasi Tanda Tangan Digital

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.

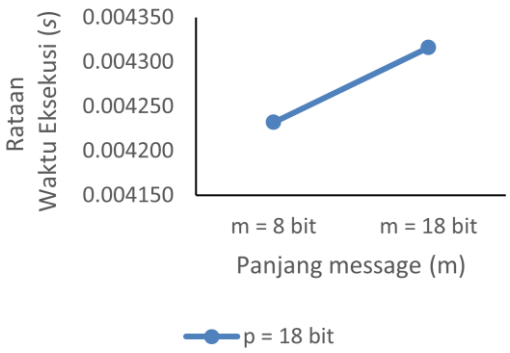
Pada Gambar 12 memaparkan hasil analisis kinerja waktu algoritme dengan panjang p bernilai tetap yaitu 8 bit dan perubahan panjang m dengan nilai, 8 bit dan 18 bit. Waktu eksekusi yang terlihat pada saat m = 8 bit sebesar 0.0028 detik dan saat m = 18 bit sebesar 0.0028 detik. Dari hasil waktu tersebut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



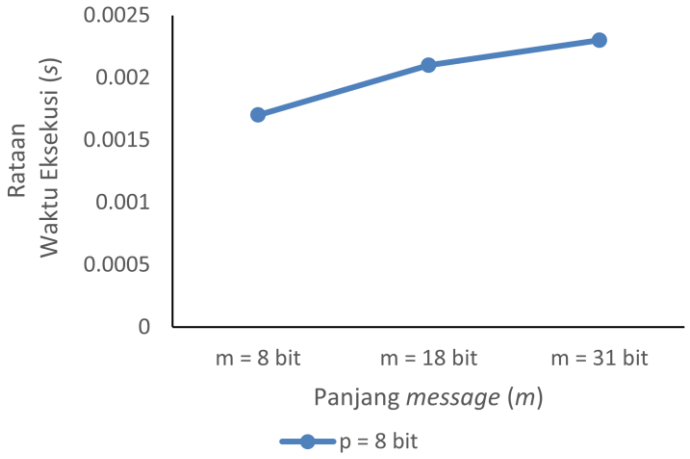
Gambar 13 Grafik kinerja algoritme verifikasi tanda tangan digital pada Arduino dengan p = 18 bit

b. Analisis Kinerja Waktu pada PC

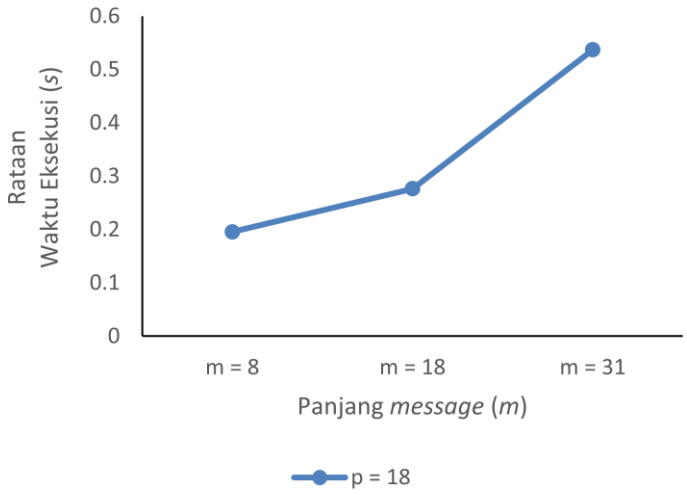
Pada tahap ini memaparkan hasil analisis kinerja waktu pada PC. Sama seperti pada Arduino Uno, analisis kinerja waktu di PC juga dibagi menjadi dua, yaitu analisis kinerja waktu pada saat pembangkitan tanda tangan digital dan analisis kinerja waktu pada saat verifikasi tanda tangan digital.

i. Analisis Kinerja Waktu untuk Pembangkitan Tanda Tangan Digital

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



Gambar 14 Grafik kinerja algoritme pembangkitan tanda tangan digital pada PC dengan p = 8 bit

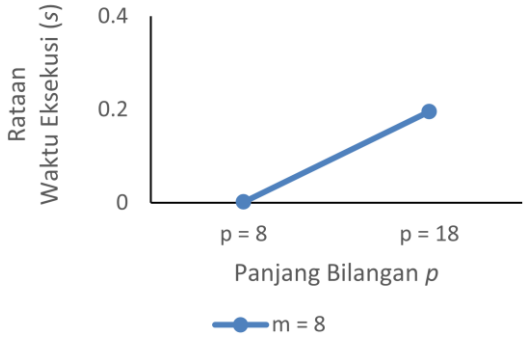


Gambar 15 Grafik kinerja algoritme pembangkitan tanda tangan digital pada PC dengan p = 18 bit

Pada Gambar 14 memaparkan hasil 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 0.0017 detik, saat m = 18 bit sebesar 0.0021 detik, dan saat m = 31 bit sebesar 0.0023 detik. Dari hasil waktu tersebut terilhat selisih perubahan waktu yang tidak besar, yakni 0.0002 detik.

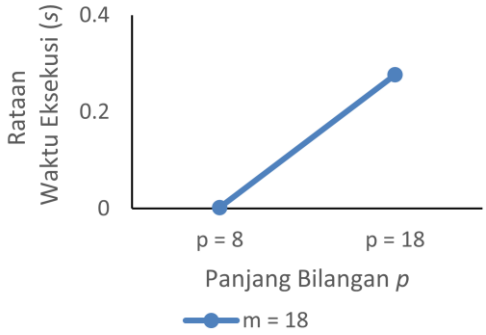
Pada Gambar 15 memaparkan analisis kinerja algoritme dengan panjang p bernilai tetap yaitu 18 bit dan memiliki perubahan panjang m yang sama. Waktu eksekusi yang terlihat pada saat m = 8 bit sebesar 0.19 detik, saat m = 18 bit sebesar 0.27 detik, dan saat m = 31 bit sebesar 0.53 detik. Perubahan waktu eksekusi yang dihasilkan pada Gambar 15 lebih besar dibandingan dengan perubahan waktu eksekusi

pada Gambar 14. Oleh karena itu, berdasarkan Gambar 14 dan 15, perubahan panjang *m* tidak memiliki pengaruh yang besar terhadap perubahan waktu eksekusi algoritme.



Gambar 16 Grafik kinerja algoritme pembangkitan tanda tangan digital pada PC dengan *m* = 8 bit

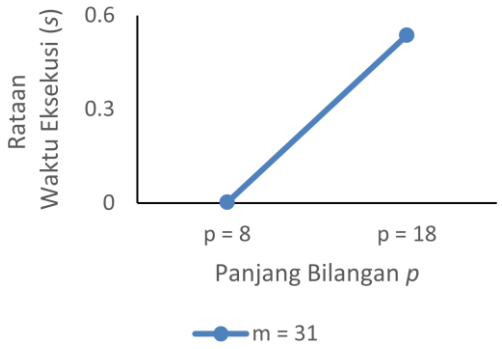
Pada Gambar 16 memaparkan analisis kinerja waktu algoritme dengan panjang *m* bernilai tetap yaitu 8 bit dan perubahan panjang *p* dengan nilai, 8 bit dan 18 bit. Waktu eksekusi yang terlihat pada saat *p* = 8 bit sebesar 0.001 detik, dan saat *p* = 18 bit sebesar 0.19 detik.



Gambar 17 Grafik kinerja algoritme pembangkitan tanda tangan digital pada PC dengan *m* = 18 bit

Pada Gambar 17 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 0.27 detik.

Pada Gambar 18 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 0.002 detik, dan saat *p* = 18 bit sebesar 0.53 detik.



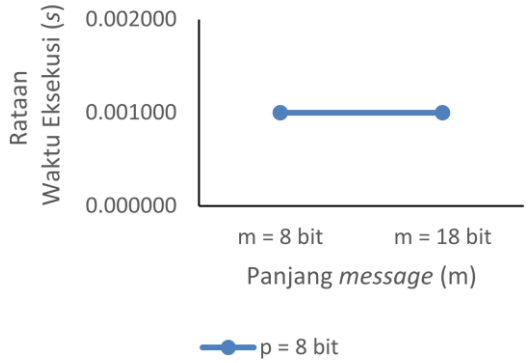
Gambar 18 Grafik kinerja algoritme pembangkitan tanda tangan digital pada PC dengan m = 31 bit

Hasil pada Gambar 16, 17, dan 18 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 16, 17, dan 18 memiliki selisih perubahan waktu yang besar dibandingkan dengan selisih perubahan waktu pada Gambar 14 dan 15.

ii. Analisis Kinerja Waktu untuk Verifikasi Tanda Tangan Digital

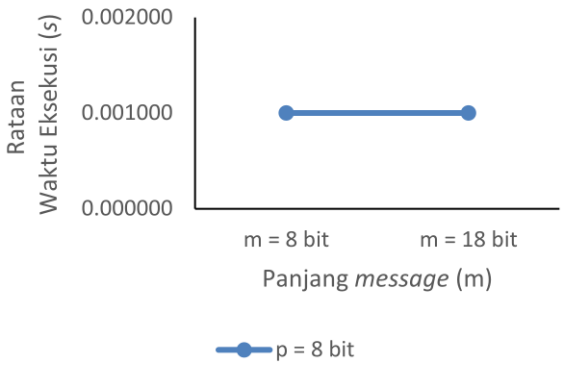
Analisis kinerja waktu untuk verifikasi tanda tangan digital dibagi menjadi dua, yaitu saat panjang p memiki nilai tetap dan panjang m memiliki perubahan nilai. Serta saat panjang pesan memiliki nilai tetap dan panjang p memiliki perubahan nilai.

Pada Gambar 19 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 0.00009 detik dan saat m = 18 bit sebesar 0.00009 detik. Dari hasil waktu tersebut terilhat selisih perubahan waktu yang tidak besar.



Gambar 19 Grafik kinerja algoritme verifikasi tanda tangan digital pada PC dengan p = 8 bit

Pada Gambar 20 memaparkan hasil analisis kinerja waktu algoritme dengan panjang *p* bernilai tetap yaitu 18 bit dan memili ki perubahan panjang *m* yang sama. Waktu eksekusi yang terlihat pada saat *m* = 8 bit sebesar 0.00009 detik dan saat *m* = 18 bit sebesar 0.00009 detik. Dari hasil waktu tersebut terlihat selisih perubahan waktu yang tidak besar sama seperti pada Gambar 19.



Gambar 20 Grafik kinerja algoritme verifikasi tanda tangan digital pada PC dengan *p* = 18 bit

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

2 Analisis Keamanan Algoritme

Analisi keamanan algoritme dilihat dari hasil verifikasi *siganture* saat ada perubahan pada *m* dan pada pasangan tanda tangan digital (*r*,*s*).

Pada Tabel 5 dan 6, 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.

Tabel 5 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 6 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

Panjang kunci *private* dihasilkan pada penilitian ini sebesar 18 bit. Jika dilakukan metode *brute force*, maka akan menghasilkan $2^{18} = 262144$ 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.

SIMPULAN DAN SARAN

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. Pada saat panjang data (*m*) bernilai tetap yaitu 31 bit, waktu eksekusi yang terlihat pada saat *p* = 8 bit sebesar 0.1 detik, dan saat *p* = 18 bit sebesar 1.3 detik. Sedangkan, pada saat panjang bilangan acak prima (*p*) bernilai tetap yaitu 18 bit, 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. Panjang *p* lebih berpengaruh terhadap kinerja algoritme dibandingkan dengan panjang *m*. Semakin panjang nilai *p* maka waktu eksekusi yang dihasilkan semakin lama. 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 *m* dan pasangan tanda tangan digital (*r*, *s*) berpengaruh terhadap hasil verifikasi keabsahan tanda tangan digital. Verifikasi tanda tangan digital terbukti gagal jika ada perubahan pada *m* dan nilai (*r*, *s*).

Saran

Penelitian lebih lanjut dari penelitian ini perlu dilakukan untuk mencapai hasil yang lebih maksimal. Berikut beberapa saran untuk penelitian selanjutnya:

1 Implementasi pada sistem IoT yang ada pada dunia nyata agar tahu kendala yang muncul saat pemasangan sistem di lapangan.
2 Mencoba menerapakan tanda tangan digital pada skema sertifikat digital untuk sistem IoT.

DAFTAR PUSTAKA

Desnanjaya GMN, Supartha KDG. 2016. Rancang bangun alat praktikum miktrokontroler STMIK STIKOM Indonesia. *Jurnal S@CIES*. 6(1):61-68.

Elgamal T. 1985. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms.* Palo Alto (US): Hewlett-Packard Labs.

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.

Kurniawan. 2016. Purwa rupa IoT (internet of things) kendali lampu gedung [skripsi]. Bandar Lampung (ID): Universitas Lampung.

Menezes AJ, Van Oorschot PC, Vanstone SA. 1996. *Handbook of Applied Cryptography*. Florida (US): CRC Press.

Mochtiarsa Y, Supriadi B. 2016. Rancangan kendali lampu menggunakan mikrokontroler ATMega328 berbasis sensor getar. *Jurnal Informatika SIMANTIK*. 1(1):40-41.

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

Schneier B. 1996. *Applied Cryptography : Protocols, Algorithms, and Source Code in C*. Ed ke-2. Hoboken (US): Wiley.

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

Suhaeb S, Djawad AY, Jaya H, Ridwansyah, Sabran, Risal A. 2017. *Mikrokontroler dan Interface*. Makassar (ID): Universitas Negeri Makassar.

RIWAYAT HIDUP

Selfi Qisthina lahir di Jakarta pada tanggal 17 Juni 1996. Ia terlahir sebagai anak terakhir dari Ibu bernama Nurlaelah dan Bapak bernama Kosasih. Ia adalah anak ke-empat dari empat bersaudara. Ia memiliki dua saudara laki-laki bernama Ferdiansyah dan Rizqo Yansyah, serta satu saudara peremuan bernama Riska Nurfajrina.

Ia memulai pendidikan nya di TK Darul Ma'arif dan TK AHDI. Kemudian, ia melanjutkan pendidikan sekolah dasar di MI AHDI, lulus pada tahun 2008. Ia melanjutkan pendidikan SMP di SMP Negeri 86, lulus pada tahun 2011. Kemudian, ia melanjutkan pendidikan SMA di SMA Negeri 97, lulus pada tahun 2014. Pada tahun yang sama ia melanjutkan pendidikannya di Institut Pertanian Bogor dan memilih Program Studi Ilmu Komputer.