Terbit *online* pada laman : http://teknosi.fti.unand.ac.id/

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |

Penerapan Keamanan WSN Berbasis Algoritma RSA 2048 dan SHA-3 pada Pemantauan Suhu

Syariful Ikhwan [a,*], *Risa Farrid Christianti* [b].

[a,b] *Fakultas Teknik Telekomunikasi dan Elektro, Institut Teknologi Telkom Purwokerto, Jl. D.I. Panjaitan No.128, Purwokerto Kidul, Kec. Purwokerto Sel., Kabupaten Banyumas, Jawa Tengah 53147*

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 15 November 2020
Revisi Akhir: 07 Januari 2021
Diterbitkan *Online*: 09 Janauri 2021

KATA KUNCI

RSA 2048,
SHA-3,
Kriptografi
WSN

KORESPONDENSI

E-mail: syariful@ittelkom-pwt.ac.id*

A B S T R A C T

Pemantauan kondisi suatu keadaan dengan menggunakan sensor semakin dibutuhkan untuk mengamati perubahan kondisi dari waktu ke waktu. Data-data yang didapatkan sensor kemudian dikirimkan ke sistem pengumpul yang telah disiapkan melalui saluran jaringan telekomunikasi. Pengirimanan data pada perangkat-perangkat jaringan telekomunikasi yang disebar pada lokasi-lokasi tertentu yang kurang aman diberbagai keadaan memungkinkan data tersebut rentan untuk diambil dan dipalsukan. Sistem pengamanan berupa kriptografi dan *hashing* kemudian digunakan untuk melindungi data agar sampai dengan baik ke penerima. Pada penelitian ini diterapkan sistem keamanan dengan mengimplementasikan algoritma kriptografi asimetrik RSA 2048 bit dan algoritma hashing SHA-3 pada pengiriman paket data yang dikirim. Hal ini dilakukan agar data bisa terjaga keasliannya dan tidak bisa dibaca oleh orang yang tidak berhak jika data tersebut didapatkan. Setelah dilakukan pengujian dengan mengirimkan paket data dari pengirim ke penerima dengan beberapa variasi jarak, didapatkan bahwa ada selisih waktu saat data dikirimkan tanpa menggunakan keamanan dan saat menggunakan keamanan SHA-3 dan RSA 2048 sebesar 70,96603 ms.

1. PENDAHULUAN

Penggunaan perangkat teknologi informasi di semua bidang kehidupan saat ini menimbulkan permasalahan baru yang pada intinya terkait dengan masalah keamanan jaringan. Keamanan jaringan menjadi sangat penting untuk diperhatikan, karena saat data yang dikirimkan, dari satu perangkat ke perangkat lain, dapat diakses oleh orang yang tidak berkepentingan, maka data tersebut bisa jadi akan disalahgunakan, sehingga menyebabkan kerugian. Metode pengamanan terhadap sistem informasi sudah dikembangkan, salah satunya adalah kriptografi. Kriptografi merupakan sistem keamanan yang bertujuan untuk menjaga keaslian dan keutuhan data, dengan memanfaatkan perhitungan matematika dalam bentuk key[1][2][3]. Di dalamnya, terdapat fungsi hash yang memiliki peran untuk memberikan keamanan tambahan dalam verifikasi data pada key.

Wireless Sensor Nework (WSN) didefinisikan sebagai salah satu jaringan wireless terdistribusi yang memanfaatkan teknologi *embedded system*[4] dan seperangkat sensor untuk melakukan proses monitoring, pengiriman data, dan penyajian informasi ke pengguna, melalui media komunikasi internet[5][6]. Tersebarnya node sensor (modul terintegrasi yang terdiri dari sensor dan unit pengirim data) WSN pada wilayah tertentu menyebabkan pemantauan dapat dilakukan tanpa mengunjungi lokasi sensor, karena node sensor telah mengirimkan data-data yang dibutuhkan secara real time. Penelitian Sarika Y. Bonde., Prof. Dr. U. S. Bhadade pada tahun 2017 yang berjudul "*Analysis of Encryption Algorithms (RSA, SRNN, and 2 key pair) for information Security*" meneliti tentang perbandingan waktu yang dibutuhkan dalam proses enkripsi dan dekripsi. Dari perbandingan algoritma RSA, SRNN, dan 2 key pair membandingkan data dengan ukuran file antara 1 KB hingga 150 KB. Dengan menggunakan fungsi library pada JAVA, didapatkan hasil bahwa untuk mengenkripsi file, algoritma RSA dan 2 key pair membutuhkan waktu lebih lama dari pada algoritma SRNN[7]. Sehingga dalam perbandingan kecepatan enkripsi, algoritma SRNN lebih cepat dibandingkan algoritma RSA dan 2 key pair. Sedangkan untuk mendekripsi file, algoritma two key pair dan SRNN membutuhkan waktu lebih lama dibandingkan algoritma RSA, sehingga dalam perbandingan kecepatan dekripsi, algoritma RSA

lebih cepat dibandingkan algoritma 2 key pair dan algoritma SRNN.

Penelitian Walid Elgenaidi, Thomas Newe, Eoin O'Connell, Daniel Toal, Gerrad Dooly, Joseph Coleman pada tahun 2016 yang berjudul *"Memory Storage Administration of Security Encryption Keys for Line Topology in Maritime Wireless Sensor Networks"* membahas terkait pengolahan penyimpanan memori, pembuatan kunci memori, dan penguncian kembali memori menggunakan skema keamanan simetrik dengan dynamic update key, berdasarkan konfigurasi node yang bekerja sebagai pihak ketiga. Dalam perancangannya, penelitian ini menggunakan perangkat Xbee dengan algoritma AES-128, untuk memperkecil kerja kunci penyimpanan pada sensor node memory storage, dimana setiap node harus menyimpan empat kunci, meliputi kunci utama dan kunci cadangan, yang kemudian proses penguncian kembali menjadi sebuah operasi lokal dalam kasus enkripsi atau dekripsi[8]. Teknik ini menyediakan kerahasiaan data berdasarkan enkripsi data file dua kali menggunakan AES-128. Autentikasi menggunakan pembagian kunci individual dengan node utama dan tanpa penolakan, saat pesan dikenali pada Xbee. Penelitian ini diimplementasikan pada platform Libelium Waspmote.

Perbandingan kecepatan enkripsi dan dekripsi algoritma RSA 2048 dibanding NTRU juga lebih baik[9]. Keamanan dalam jaringan sensor nirkabel umumnya didasarkan pada enkripsi simetris dan membutuhkan sistem manajemen kunci untuk membuat dan bertukar kunci rahasia. Kendala umum bagi pendekatan manajemen kunci adalah batas atas jumlah total node dalam jaringan[6]. Berbagai pendekatan kemudian digunakan untuk memaksimalkan cakupan dan konektifitas jaringan[10]. WSN saat ini banyak menggunakan teknologi Zigbee, namun kendala yang dihadapi pada teknologi ini adalah kelemahan pada protokol keamanannya[11], jangkauan yang pendek dan tidak terlalu jauh. Pengembangan teknologi dalam komunikasi perangkat kemudian diusulkan dimana LoRa menjadi salah satu pilihan yang saat ini beroperasi pada frekuensi tidak berlisensi[12].

Berdasarkan penelitian yang dilakukan pada [13] diketahui bahwa RSA 2048 adalah algoritma RSA yang paling baik dalam mengamankan data yang dikirimkan pada jaringan. Beberapa pengujian yang dilakukan mendapatkan hasil bahwa RSA 2048 tetap bisa mengamankan kunci dengan baik. Pada penelitian [14] didapatkan bahwa RSA memiliki keunggulan lebih cepat saat melakukan dekripsi pesan.

Pengamanan data dijaringan komputer membutuhkan sistem yang baik demikian juga saat data telah sampai ke penerima. Data yang telah sampai di penerima harus di verifikasi untuk memastikan bahwa data tersebut adalah data yang benar sebagaimana dikirimkan dari sisi pengirim. Untuk memastikan bahwa data yang diterima adalah data yang benar, maka digunakan fungsi hashing yang akan menghasilkan fungsi yang sama saat di verifikasi pada sisi penerima. Salah satu fungsi hashing yang saat ini banyak digunakan adalah *Secure Hashing 3* (SHA-3)[15][16]. SHA-3 menggunakan algoritma Keccak[17][18]. Kinerja SHA-3 lebih baik daripada pendahulunya yaitu SHA-1 dan SHA-2. SHA 3 memiliki varian yaitu SHA3-224, SHA3-256, SHA3-384, SHA3-512 dan dua fungsi output yang dinamakan dengan SHAKE128 dan SHAKE256 [19] [20].

Penelitian ini melakukan pengiriman paket data dari satu perangkat ke perangkat lain dengan menggunakan kunci enkripsi dan dekripsi kemudian melakukan verifikasi terhadap data yang dikirimkan dan data yang diterima. Algoritma yang digunakan dalam enkripsi dan dekripsi pada penelitian ini adalah RSA 2048 sementara untuk fungsi hash digunakan SHA-3. Pengujian terhadap sistem akan memperlihatkan berapa lama waktu proses enkripsi dan dekripsi yang dilakukan hingga data sampai di sisi penerima dengan baik.

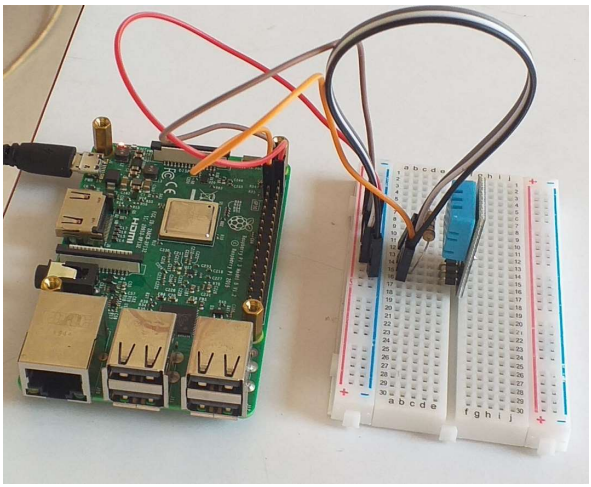
2. METODE

Metode Penelitian yang digunakan pada penelitian ini berupa eksperimentasi yang diterapkan pada perangkat yang sebenarnya. Perangkat yang digunakan berupa Raspberry dan sensor suhu dan kelembaban.

2.1. Peralatan yang digunakan

Pada penelitian ini digunakan Raspberry Pi™ 3 Model B Quad Core 1,2 GHz dengan RAM 1 Gb dan memanfaatkan koneksi WiFi 802.11 frekuensi 2,4 Ghz. Sistem operasi yang digunakan adalah Debian OS. Pemrosesan algoritma pemograman dilakukan pada Python3 dan berbasis IP versi 4.

Sensor yang digunakan sebagai pemicu agar data dikirimkan dari Raspberry Pi™ yang berfungsi sebagai sensor adalah sensor suhu dan kelembaban. Sensor ini menggunakan seri DHT11. Setiap kali ada perubahan pada suhu yang dipantau maka perangkat akan mengirimkan data. Gambar 1 memperlihatkan Raspberry Pi dan sensor yang digunakan pada penelitian.

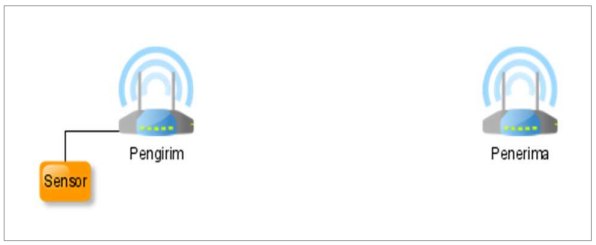


Gambar 1. Perangkat Raspberry Pi™ dan Sensor yang digunakan

2.2. Topologi Penelitian

Topologi yang digunakan dalam penelitian menentukan bagaimana hasil yang didapatkan nantinya setelah pengujian dilakukan. Hal ini penting untuk diketahui agar tidak terjadinya kesalahpahaman dalam menentukan jalur data yang digunakan oleh paket yang dikirimkan melalui jaringan. Topologi komunikasi memanfaatkan gelombang radio sebagai sarana

komunikasi dua arah.



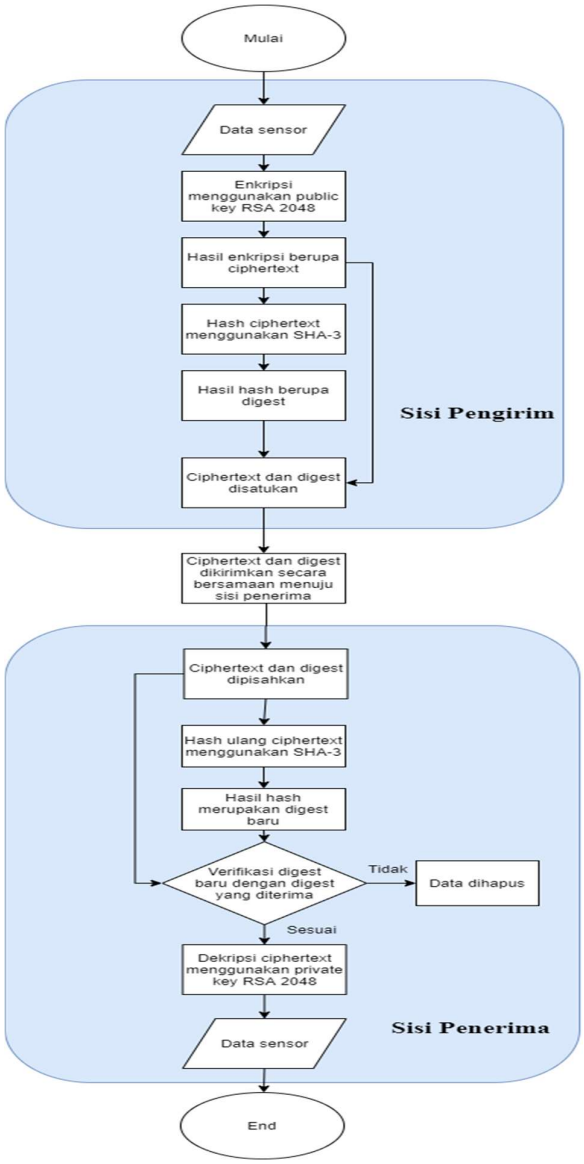
Gambar 2 Topologi pengujian

Gambar 2 memperlihatkan topologi pengujian yang dilakukan saat data diambil dari sensor kemudian dienkripsi dan dikirimkan ke penerima. Perangkat pengirim dan penerima menggunakan Raspberry Pi™ 3 B. Raspberry ini sudah dilengkapi dengan komunikasi wireless pada frekuensi 2,4 GHz. Pengujian pengiriman secara langsung menggunakan *wireless card* yang telah tertanam pada Raspberry tanpa menambahkan antena luar.

Flowchart proses dalam mendapatkan data diperlihatkan pada gambar 3. Proses yang terjadi pada sisi pengirim yaitu, pertama kali sensor mengambil data berupa suhu dan kelembaban. Data tersebut kemudian di enkripsi menggunakan kunci publik RSA 2048 sehingga didapatkan *cipertext*. Hasil enkripsi kemudian di hash menggunakan SHA-3 dan didapatkan hasil berupa *digest*. Proses selanjutnya adalah menggabungkan hasil *digest* dan *ciphertext* yang kemudian digabungkan untuk dikirimkan ke penerima.

Pada sisi penerima, saat file yang dikirimkan dari pengirim telah diterima berupa *ciphertext* dan *digest*, keduanya kemudian dipisahkan kembali. *Ciphertext* kemudian di hash dan dicocokkan dengan *digest* yang diterima sebelumnya apabila hasil *digest* yang di hash tersebut sama maka file *ciphertext* dianggap asli dan dilakukan proses selanjutnya. Apabila hasil *digest* tidak sama dengan yang telah dikirimkan, maka file dianggap telah diubah dan tidak diproses. File *ciphertext* akan dihapus karena dianggap tidak valid. Proses yang dilakukan apabila file telah terverifikasi benar adalah melakukan dekripsi *ciphertext* dengan menggunakan *private key* RSA 2048. Proses ini menghasilkan data sensor sebagaimana ketika didapatkan pada sisi pengirim.

Pengiriman data yang dilakukan pada penelitian ini menggunakan protocol TCP *(Transport Control Protocol).* Protokol TCP yang dipertukarkan pada pengiriman membuat koneksi sebagaimana prosedur dalam pengiriman data TCP yaitu melakukan koneksi *Three-Way Handshake* (SYN, SYN-ACK, ACK) pada saat awal komunikasi dilakukan. Saat perangkat yang berkomunikasi telah terkoneksi, maka proses *Three-Way Handshake* tidak dilakukan lagi pada pengiriman berikutnya. Setelah proses komunikasi untuk tercapainya kesepakatan pengiriman data kedua perangkat, maka dilakukan kesepakatan kunci yang akan digunakan dalam komunikasi. Komunikasi yang dikirim selanjutnya adalah data suhu dan kelembaban yang didapatkan dari sensor. Setelah terkirim dan diterima oleh perangkat penerima, maka informasi ACK yang menandakan bahwa paket telah diterima dikirimkan.



Gambar 3. Flowchart pemrograman

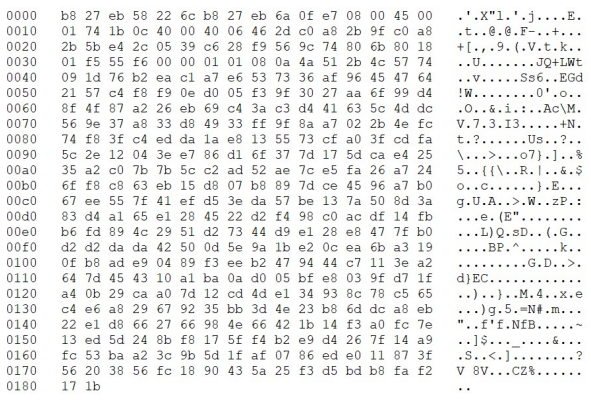
Gambar 4 memperlihatkan paket yang tidak dienkripsi saat pengiriman menggunakan komunikasi Raspberry Pi™ pada penelitian ini. Pengamatan data yang dilakukan sebagaimana di gambar 4 terlihat nilai *plaintext* yang dikirimkan bisa dibaca dengan jelas. Pada contoh terlihat nilai waktu pengambilan data, nilai temperatur yang diamati pada sensor dan kelembabannya. Dapat dipastikan bahwa jika data *plaintext* ini dikirimkan melewati jaringan yang tidak aman, maka data akan mudah dibaca dan diubah oleh pihak ketiga.

```
0000   b8 27 eb 58 22 6c b8 27 eb 6a 0f e7 08 00 45 00   .'.X"l.'.j....E.
0010   00 60 2c 7f 40 00 40 06 35 ce c0 a8 2b 9f c0 a8   .`,.@.@.5...+...
0020   2b 5b e4 2a 05 39 55 a3 ed 91 ca c3 59 04 80 18   +[.*.9U.....Y...
0030   01 f6 97 20 00 00 01 01 08 0a 4a 50 cd a6 57 73   ... ......JP..Ws
0040   ab ae 31 36 30 33 38 35 31 38 30 37 2e 32 31 38   ..1603851807.218
0050   37 30 35 30 3e 54 65 6d 70 3d 32 39 2e 30 2a 20   7050>Temp=29.0*
0060   48 75 6d 69 64 69 74 79 3d 37 38 2e 30 25         Humidity=78.0%
```

Gambar 4. Paket data yang tidak terenkripsi

https://doi.org/10.25077/ TEKNOSI.v6i3.2020.150-157

Pengiriman data yang telah diamankan dengan menggunakan kunci RSA 2048 dan SHA-3 dapat dilihat pada gambar 5. Data awal yang berbentuk plaintext telah berubah menjadi karakter-karakter yang tidak dipahami. Penggunaan simbol dan karakter yang banyak pada file enkripsi akan sangat menyulitkan bagi manusia maupun sistem komputer untuk mengetahui maksud yang terkandung dalam *ciphertext*.



Gambar 5. Paket data yang terenkripsi

Karakter dan simbol yang muncul pada ciphertext terbentuk dari pengacakan yang dilakukan algoritma RSA 2048. Jumlah bit data plainteks yang terbentuk adalah sebanyak 2048 bit.

Pembangkitan kunci publik dan kunci privat RSA adalah sebagai berikut:

a. Pilih dua bilangan prima sembarang yakni p dan q
b. Hitung dan simpan sebuah nilai variabel $n = p.q$, sebaiknya nilai $p \neq q$
c. Hitung nilai $\varphi(n) = (p-1)(q-1)$.
d. Pilih kunci publik e yang relatif prima terhadap $\varphi(n)$, dua bilangan dikatakan relatif prima jika gcd *(greatest common divisor)* kedua bilangan tersebut bernilai 1.
e. Bangkitkan kunci privat d dengan menggunakan persamaan $e.d \equiv 1 \; mod \; \varphi(n)$
   Nilai d didapatkan dengan mengurai persamaan $e.d \equiv 1 \; mod \; \varphi(n)$ dimana $e.d \equiv 1 \; (mod \; \varphi(n))$ dan persamaan ini ekivalen dengan $e.d = 1 + k \; \varphi(n)$ sehingga dapat digunakan persamaan untuk d secara sederhana
   $$d = \frac{1 + k\varphi(n)}{e}$$
   dengan demikian didapatkan nilai kunci publik (e, n) dan kunci privat (d, n).

Proses pembangkitan kunci publik dan kunci privat kemudian dilanjutkan dengan melakukan enkripsi dan dekripsi pesan yang dikirimkan. Proses enkripsi pesan dilakukan dengan tahapan sebagai berikut :

a. Gunakan kunci publik penerima pesan (e, n).
b. Bentuk pesan plainteks menjadi beberapa blok m sehingga menjadi $m_0, m_1, m_2, m_3, \ldots \; m_{n-1}$ sehingga setiap blok akan menyatakan nilai dalam selang $[0, n-1]$.
c. Selanjutnya setiap blok m_i di enkripsi menjadi blok c_i menggunakan persamaan $c_i = {m_i}^e \; mod \; n$.

Pesan yang telah di enkripsi kemudian dikenal dengan *ciphertext* dikirimkan ke sisi penerima. Penerima kemudian melakukan dekripsi dengan menggunakan tahapan berikut :

a. Blok cipher yang telah diterima dari pengirim berupa c_i didekripsikan kembali menjadi m_i dengan menggunakan persamaan $m_i = {c_i}^d \; mod \; n$.
b. Blok-blok m_i kemudian digabungkan kembali menjadi $m_0, m_1, m_2, m_3, \ldots \; m_{n-1}$ sehingga membentuk plainteks.

Sebagai contoh dalam pengiriman pesan pada penelitian ini, pesan yang dikirimkan adalah waktu pengambilan data dari sensor berupa nilai suhu dan kelembaban yang dibaca oleh sensor. Nilai tersebut contohnya adalah 1603851807.2187050> Temp=29.0* Humidity=78.0%. Jika nilai tersebut diubah ke bentuk hexadecimal adalah 31 36 30 33 38 35 31 38 30 37 2e 32 31 38 37 30 35 30 3e 20 54 65 6d 70 3d 32 39 2e 30 2a 20 48 75 6d 69 64 69 74 79 3d 37 38 2e 30 25. Nilai ini kemudian di buat dalam beberapa blok kemudian dienkripsi menggunakan kunci publik sehingga didapatkan ciphertext.

Secara lebih detail, bentuk aplikatif yang digunakan pada penelitian ini berupa pesan 1603851807.2187050> Temp=29.0* Humidity=78.0%, untuk contoh kunci publik yang digunakan misalnya bentuk yang sudah di *generate* oleh sistem adalah

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQE
AmShw3mSLZ4nN3w+mCK6RG6lBaES4aNFJH8NlZoE6cLd
PGUkQCB1Qr161x4CgCKglWmcS9ErbXPXvGHGQqEw3jbfy
lEZgVdrd6nijXvjX9whiOLn242qNhPz46z9N2dgalHG8Cl0R4
KPdqwaNOmT30MAp8ONWJjpQXlevTn1WCSrfR5qdpUiQ8
Rg3LFOm5o1u8jU+7QxPwLufyec2yVfj8uwszO9aeq7I975ELz
qfNHC+0kyBzFBttp24Vo25jU+C0avlpR38icdgW9Xn82Y2r7
WxOn4U31cNutktxYSK0Ix/rR8VhrtII5S8Rg1noTjj+u9HaHhM
iX6BxoTqhXLgGQIDAQAB.

Output cipher teks yang dihasilkan dari penggunaan kunci publik tersebut adalah

OJJazTDFYoHtlHUOjDLziqxCcWUupBKV19zzuWQXSDiAi
2LBpGJCUHKUZXv4xBH+vWrw5Z4cLHipeQnmLayzG+jCs
PHzkQCb+yMBgFpvsWvGulvYEk96UiAb/A22STyXIuhiZ+B
xxZAJgieWBbu23f60vEsp54jW1YJ+A2wlTXa0YkZ/kZwHQ4
IluQomWvzfoFdvjnz2HTpT3fSwdBXrsQLudB7TwYMG4Pym
XYmbOyU6B/hH+CIbiUflFlPpKDjpdmIRdzM8OeDdzcxhwG
DAehStV59BQQue81tT3aG+FzEM7RIVJ8UQ0UupYPLmKgZ
u0dPZ1CuiLyfqCSjej6/WZA==.

Untuk melakukan dekripsi pada sisi penerima digunakan kunci privat yang sebelumnya telah dimiliki oleh penerima yaitu,

MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgE
AAoIBAQCZKHDeZItnic3fD6YIrpEbqUFoRLho0Ukfw2Vmg
Tpwt08ZSRAIHVCvXrXHgKAIqCVaZxL0Sttc9e8YcZCoTDe
Nt/KURmBV2t3qeKNe+Nf3CGI4ufbjao2E/PjrP03Z2BqUcbw
KXRHgo92rBo06ZPfQwCnw41YmOlBeV69OfVYJKt9Hmp2l
SJDxGDcsU6bmjW7yNT7tDE/Au5/J5zbJV+Py7CzM71p6rsj3v
kQvOp80cL7STIHMUG22nbhWjbmNT4LRq+WlHfyJx2Bb1ef
zZjavtbE6fhTfVw262S3FhIrQjH+tHxWGu0gjlLxGDWehOOP
670doeEyJfoHGhOqFcuAZAgMBAAECggEBAIOwyTQAIfm
5/OGik/H9H9o383yTMjk2UvjTyvmTfkty4rlsIDPJ+RmBhy3uh
CxNYCjWXe7d+rZ45MgQxgBf4JijpRonVSgdVBkK9/ZuY0w
O0l3SUN4EPYEu/11PhkIc1eXz0Wrb0cZlP/GRZZfjP04oHSf6
YZXnKDmmjZ8gl/yyoN19t4WVyZzIRT8WcISNntfs2vT9iPgB
GCqEb4yVF2yD1WDN5Vl3RfNNptKALL6xgh7FY9JAtO89B

8Hr9UyLE2QQFbq3KPPi0+45Wp/QFmaFsuKV0IfIIZGQ39qhj
F5wd/r/Q/CiLShNEEz86wQWa9P4aEJKFd20FjKPfyUplc0Cg
YEA3JwZEXHg6NIxOswMyWMyFjsj/VCNxKWNrMsuwac2d
uMCcpdR1OrmC5bLco26Ji7K4VP9hhb8xLvu9vD/AgcZoO4U
hA+VJm4LYaBXwo2y789+QP9iIyFVOZP7WhH4v0mEcBNr
+V5Y8lzSpJj1X3tcSdIULopcn+Fo4IdXgkQ3s08CgYEAsbpDc
gcIUBbcnmKhqZ1L5OdVR0prSAy6Yn/AVtq9W7yz/UBfLZX
xCdWBHNo3UivbiC89mo8RdGrhUGLrWMKwaky3uTImGM
V8xF1ByNfIxe0Fzy+50zckWoP+u/Uj1FvpFveJrIrY+tQufTqm
vVf1nbcGykxw9pEdk3E9+zUQ/BcCgYALftrTjOKSXzmPfciA
yfl3LRfAAG2SvKm4QlIqZyMwaL3c6MOr252nJvFtA7Oaf7+0
pCLG1PXQ7pYSKhavrV0JhL5u7zzvZraTHL7V9XpIgsBqdKR
RflApkut7EwqqNROAn5J1yRPLdUjIqOvIUO5LLnY8wpr2/d
DwOj02HYXlgwKBgEsNucXBqoUH4osQY+zY/F/iQWUlzEB
U/7Dw+iyzmxvOiMP+FzONCol4nFmu3YXZ949/Q6itmlVrfV
MRMwn8zbtWQo/22ZfEbngOWeWTGrpa6gUgxnyn4LZI355I
W6UmWzzTtFQlHpRuXmOMDTYQmyzLShbMSTgW+EJ3p
BEdy9ppAoGBAJG3oYjslNy21es5KXq4ad1YHBx+BO2etXx1
OSlF1/5mfFyvBzJyVdXA8RBXGHVwByCJUmVMcEo2v68i/
xJz+2eUnKbo0ziSn94e7nrMVcmlEtckmfvlCfl/f9Chcsxk9tavxJ
IvnCll/kpyzPFDU4AhyCT3AitUZfuGMAgtRoV4.

Hasil yang didapatkan ketika didekripsi dengan menggunakan kunci private tersebut adalah teks berupa 1603851807.2187050> Temp=29.0* Humidity=78.0%.

Fungsi hashing dilakukan terhadap *ciphertext* yang telah dihasilkan pada proses enkripsi RSA 2048. Nilai *digest* yang dihasilkan saat di *hash* dengan fungsi Keccak-512 bit pada contoh diatas adalah 1b09ae0349677a9b9bba3580800489a8dfa3651dc98ddf160d680ee94dab9d321e833bb8175c606b00a08c8c3b06050ae1b7cc9b123c162faadfb692ab6ef639. Nilai ini juga akan didapatkan oleh penerima saat dilakukan *hashing*. Apabila nilainya tidak sama, maka bisa dipastikan telah terjadi perubahan pada *file* yang dikirimkan.

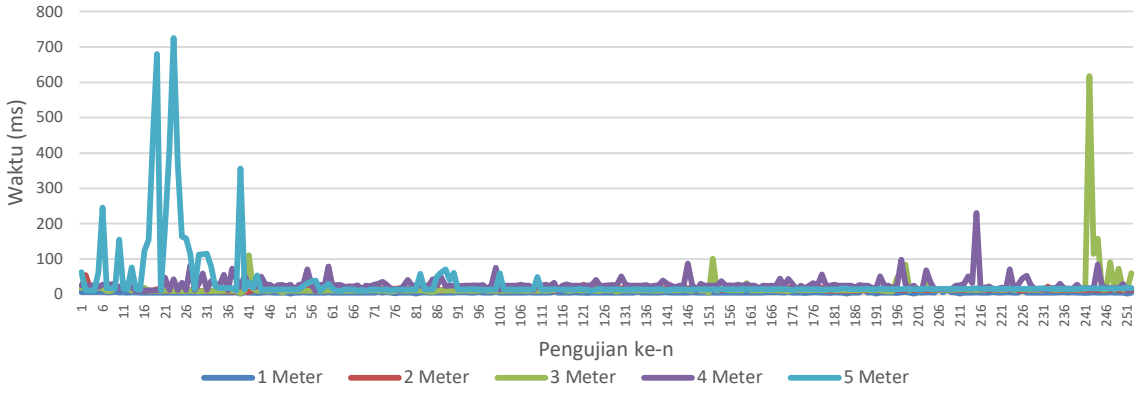
Panjang frame yang dikirimkan oleh pengirim berdasarkan data yang didapat ketika pengujian adalah 386 byte (3088 bit) saat data yang dikirimkan berupa *ciphertext*. Data yang masih berbentuk plainteks memiliki panjang *frame* 110 byte (880 bit). Terdapat selisih panjang frame 276 byte saat data yang dikirimkan berbentuk *ciphertext*.

3. HASIL

Pengujian pengiriman data menggunakan Raspberry pi™ dengan menggunakan enkripsi RSA2048 dan SHA3 dilakukan sebanyak 252 kali pengiriman. Pengiriman sebanyak 252 kali ini dilakukan pada setiap skenario penelitian yaitu pada jarak 1 meter, 2 meter, 3 meter, 4 meter dan 5 meter. Perubahan jarak ini dilakukan untuk mengetahui berapa besar perbedaan saat dikirimkan menggunakan metode enkripsi dan tidak menggunakan enkripsi. Hasil yang didapatkan pada setiap skenario diambil rata-rata nilai. Pengimplementasian enkripsi menggunakan RSA 2048 dan SHA3 ini kemudian dibandingkan dengan saat data dikirimkan tanpa menggunakan proses enkripsi agar dapat diketahui seberapa besar pengaruh yang ditimbulkan oleh sistem yang menerapkan proses enkripsi.

3.1. Hasil Delay tanpa enkripsi

Pola yang terbentuk dari pengiriman pesan akan berbeda saat dikirimkan dengan menggunakan enkripsi dan saat tidak menggunakan enkripsi. Agar terlihat perubahannya, maka pada penelitian ini dilakukan pengiriman pesan saat data yang dikirimkan belum dilakukan enkripsi. Gambar 6 memperlihatkan pesan yang dikirimkan tanpa enkripsi pada sisi pengirim.



Gambar 6. Delay pengiriman saat tidak dienkripsi.

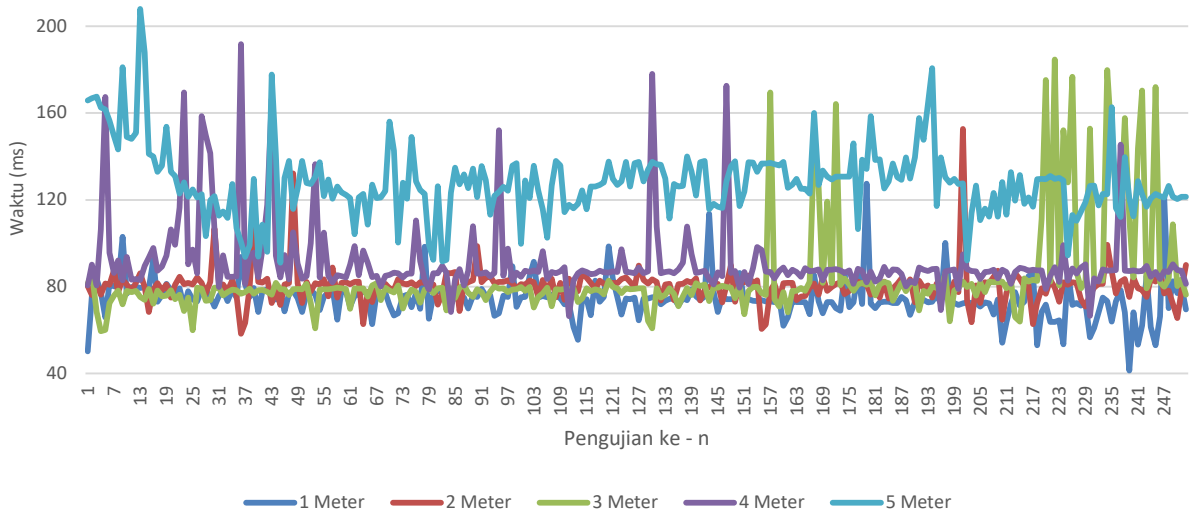
Gambar 6 memperlihatkan grafik hasil delay pengiriman paket data dengan selisih jarak 1 meter pada setiap pengujian saat data tidak di enkripsi. Pengujian 1 meter, 2 meter, 3 meter dan 4 meter terlihat data yang dikirimkan diterima dengan baik dan rata-rata pengiriman berada pada rentang yang tidak terlalu jauh dari jarak lainnya. Pada pengiriman dengan jarak 5 meter, pada awal pengiriman terjadi lonjakan dan ketidakstabilan delay. Delay yang terjadi cukup besar diawal pengiriman dan sangat berbeda dengan yang lainnya namun saat berikutnya hingga pengiriman berakhir, delay yang terjadi kembali stabil.

3.2. Hasil Delay dengan enkripsi

Enkripsi pada pesan yang dikirimkan mempengaruhi delay yang terjadi baik pada saat di proses awal maupun saat di proses di jaringan dan di sisi penerima. Pada sisi pengirima dan penerima terjadi proses pembuatan kunci publik dan kunci privat,

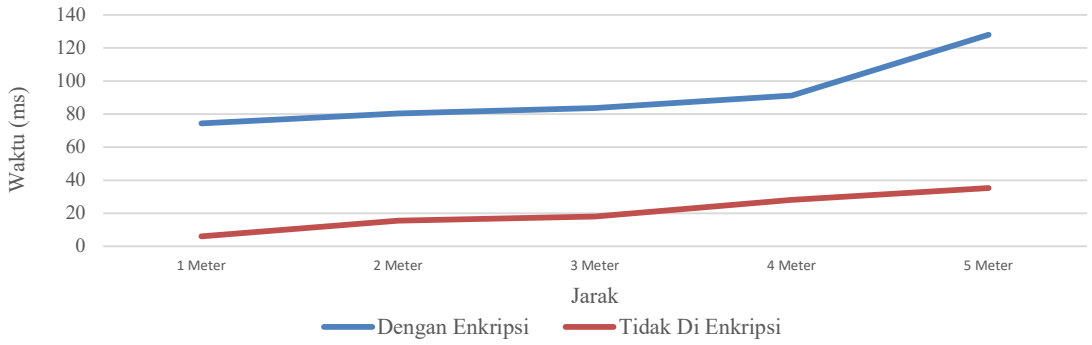
https://doi.org/10.25077/ TEKNOSI.v6i3.2020.150-157

sementara pada sisi jaringan saat dikirimkan terjadi penambahan jumlah *byte* yang diterima dan diteruskan ke *device* berikutnya.



Gambar 7. Delay pengiriman dengan enkripsi

Pada gambar 7 diperlihatkan bahwa saat data dikirimkan dengan menggunakan enkripsi RSA 2048 dan hashing SHA3 terjadi kenaikan saat diberi jarak. Jika diperhatikan, pada pengiriman menggunakan proses enkripsi terlihat delay pengiriman lebih stabil daripada saat tidak dilakukan enkripsi. Pengiriman dengan menggunakan enkripsi tidak terjadi lonjakan delay yang cukup besar walaupun nilainya lebih tinggi dari rata-rata tanpa enkripsi. Awal pengiriman dengan enkripsi terlihat stabil hingga pengiriman berakhir. Anomali yang terjadi sebagaimana di pengiriman paket tanpa enkripsi pada jarak 5 meter tidak terjadi.



Gambar 8. Perbandingan saat dienkripsi dan tidak di enkripsi.

Gambar 8 menampilkan grafik selisih delay pengiriman saat di enkripsi dan tidak di enkripsi. Jika diperhatikan, maka terlihat nilai selisihnya linear antara kedua proses tersebut pada setiap skenario yang dilakukan. Hal ini mengindikasikan bahwa secara pengujian, Langkah yang dilakukan sudah memenuhi prosedur yang dirancang.

Tabel 1. Selisih Delay enkripsi dan tidak di enkripsi

	Delay (ms)				
Perlakuan	1 Meter	2 Meter	3 Meter	4 Meter	5 Meter
Dengan Enkripsi	74,44841	80,41746	83,71944	91,20873	128,0901
Tidak Di Enkripsi	6,043651	15,625	18,01032	28,13849	35,23651
Selisih	68,40476	64,79246	65,70913	63,07024	92,85357

Pada Tabel 1 dapat dilihat dengan lebih jelas bahwa rata-rata delay antara saat dikirimkan pada jarak 1 meter, 2 meter, 3 meter dan 4 meter tidak jauh berbeda yaitu ada pada kisaran 65 ms. Pada jarak 5 meter sedikit lebih naik. Kenaikan pada jarak 5 meter ini jika di perhatikan pada gambar 6, adalah karena adanya delay yang cukup tinggi diawal pengiriman. Jika dirata-ratakan seluruh nilai selisih delay antara saat di enkripsi dan saat tidak di enkripsi adalah sebesar 70,96603 ms. Nilai ini mengindikasikan bahwa terjadi perbedaan nilai delay dengan adanya enkripsi pada sisi pengirim dan dekripsi pada sisi penerima. Nilai penambahan delay yang terjadi adalah sebesar 70,96603 ms.

4. PEMBAHASAN

Perbedaan delay antara saat dikirimkan dengan menggunakan skema algoritma kriptografi memiliki nilai yang lebih besar

dibandingkan saat tidak di enkripsi. Hal ini dapat dipahami karena saat dilakukan enkripsi maka proses pengolahan data membutuhkan waktu yang lebih banyak.

Penambahan jarak antar perangkat yang berkomunikasi juga mempengaruhi delay pengiriman yang terjadi. Pada penelitian ini jarak yang digunakan adalah 1, 2, 3, 4 dan 5 meter terlihat delay yang terjadi semakin besar. Kenaikan rata-rata delay pada pengiriman saat di enkripsi adalah 13,41042 ms sedangkan saat tidak di enkripsi sebesar 7,29821 ms. Nilai rata-rata kenaikan delay ini berarti pada saat di enkripsi naik menjadi dua kali lipat.

4. KESIMPULAN

Pengujian pengiriman data pada perangkat WSN yang menggunakan Raspberry Pi™ sebagai sistemnya mengalami kenaikan waktu pengiriman paket sebesar 70,96603 ms. Peningkatan ini diakibatkan karena proses enkripsi dan dekripsi data pada sisi pengirim dan penerima data. Pengimplementasian pada sistem peringatan dini terkait kejadian-kejadian alam kurang disarankan untuk digunakan karena akan menjadikan data sedikit lebih lambat sampai ditujuan. Namun untuk sistem yang tidak menggunakan kecepatan pengiriman sebagai parameter utama, maka sistem dengan menggunakan RSA 2048 dan SHA-3 ini patut untuk dipertimbangkan karena kelebihan dalam sistem pengamanan kunci enkripsi dan keotentikan datanya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan yang telah mendanai penelitian ini pada Tahun Anggaran 2020 melalui skema Penelitian Dosen Pemula dengan Surat Kontrak Nomor 082/SP2H/LT/DRPM/2020.

DAFTAR PUSTAKA

[1] M. S. Bari and A. T. Siddique, "Study on different Cryptography Algorithm a Critical Review," *Int. J. Adv. Res. Comput. Eng. Technol. Vol. 6, Issue 2, Febr. 2017, ISSN 2278 – 1323 Study*, vol. 6, no. 2, pp. 177–182, 2017.

[2] M. R. Joshi and R. A. Karkade, "Network Security with Cryptography," *Int. J. Comput. Sci. Mob. Comput.*, vol. 41, no. 1, pp. 201–204, 2015.

[3] F. Maqsood, M. Ahmed, M. Mumtaz, and M. Ali, "Cryptography: A Comparative Analysis for Modern Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 442–448, 2017, doi: 10.14569/ijacsa.2017.080659.

[4] H. Prihtiadi and M. Djamal, "The reliability of wireless sensor network on pipeline monitoring system," *J. Math. Fundam. Sci.*, vol. 49, no. 1, pp. 51–56, 2017, doi: 10.5614/j.math.fund.sci.2017.49.1.5.

[5] A. H. Moon, U. Iqbal, and G. M. Bhat, "Implementation of Node Authentication for WSN Using Hash Chains," *Procedia Comput. Sci.*, vol. 89, pp. 90–98, 2016, doi: 10.1016/j.procs.2016.06.013.

[6] F. Gandino, B. Montrucchio, and M. Rebaudengo, "Redundancy in Key Management for WSNs," *Cryptography*, vol. 2, no. 4, p. 40, 2018, doi: 10.3390/cryptography2040040.

[7] S. Y. Bonde and U. S. Bhadade, "Analysis of Encryption Algorithms (RSA, SRNN and 2 Key Pair) for Information Security," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017, pp. 1–5, doi: 10.1109/ICCUBEA.2017.8463720.

[8] W. Elgenaidi, T. Newe, E. O'Connell, D. Toal, G. Dooly, and J. Coleman, "Memory Storage Administration of Security Encryption Keys for Line Topology in Maritime Wireless Sensor Networks," in *2016 10th International Conference on Sensing Technology (ICST)*, 2016, pp. 1–4, doi: 10.1109/ICSensT.2016.7796268.

[9] K. Shim, "A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 577–601, 2016, doi: 10.1109/COMST.2015.2459691.

[10] Y. El Khamlichi, A. Tahiri, A. Abtoy, I. Medina-Bulo, and F. Palomo-Lozano, "A hybrid algorithm for optimal wireless sensor network deployment with the minimum number of sensor nodes," *Algorithms*, vol. 10, no. 3, 2017, doi: 10.3390/a10030080.

[11] W. Razouk, G. V. Crosby, and A. Sekkaki, "New security approach for ZigBee weaknesses," *Procedia Comput. Sci.*, vol. 37, no. June 2016, pp. 376–381, 2014, doi: 10.1016/j.procs.2014.08.056.

[12] S. Bertoldo, L. Carosso, E. Marchetta, M. Paredes, and M. Allegretti, "Feasibility Analysis of a LoRa-Based WSN Using Public Transport," *Appl. Syst. Innov.*, vol. 1, no. 4, p. 49, 2018, doi: 10.3390/asi1040049.

[13] B. K and D. S.S, "An Overview of Cryptanalysis of RSA Public key System," *Int. J. Eng. Technol.*, vol. 9, no. 5, pp. 3575–3579, 2017, doi: 10.21817/ijet/2017/v9i5/170905312.

[14] K. Maletsky, "RSA vs ECC Comparison for Embedded Systems," *Atmel-8951A-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper_072015*, 2015.

[15] N. R. Chandran and E. M. Manuel, "Performance Analysis of Modified SHA-3," *Procedia Technol.*, vol. 24, pp. 904–910, 2016, doi: 10.1016/j.protcy.2016.05.168.

[16] F. Kurniawan, A. Kusyanti, and H. Nurwarsito, "Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi Garuda Training Cost," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 1, no. 9, pp. 803–812, 2017, [Online]. Available: http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/247.

[17] S. Neelima and R. Brindha, "A Low Power FPGA Implementation of SHA3 Design," vol. 28, no. 16, pp. 188–205, 2019.

[18] R. A. Azdy, "Tanda tangan Digital Menggunakan Algoritme Keccak dan RSA," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 5, no. 3, pp. 184–191, 2016, doi: 10.22146/jnteti.v5i3.255.

[19] NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," 2015.

[20] X. Wu and S. Li, "High throughput design and implementation of SHA-3 hash algorithm," *EDSSC 2017 - 13th IEEE Int. Conf. Electron Devices Solid-State Circuits*, vol. 2017-Janua, pp. 1–2, 2017, doi: 10.1109/EDSSC.2017.8126446.

NOMENKLATUR

p	bilangan prima pertama
q	bilangan prima kedua
n	bilangan bulat positif
$\varphi(n)$	variabel penyimpan nilai ke n
k	variabel bilangan bulat dari 1, 2, 3 dan seterusnya
e	bilangan prima yang relatif prima terhadap $\varphi(n)$
mod	singkatan dari modulus

BIODATA PENULIS

Syariful Ikhwan, ST., MT. dilahirkan di Air Bangis, Sumatera Barat tanggal 5 April 1982. Tamat SMA melanjutkan pendidikan di Teknik Elektro Universitas Andalas dan meraih gelar Sarjana Teknik pada tahun 2008. Bekerja sebagai staf jaringan di Lembaga Pengembangan Teknologi Informasi dan Komputer (LPTIK) Universitas Andalas sembari kemudian melanjutkan Studi Magister di Jurusan yang sama saat S1 hingga lulus pada tahun 2014. Saat ini aktif sebagai pengajar di Program Studi D3 Teknik Telekomunikasi, Institut Teknologi Telkom Purwokerto.

Risa Farrid Christianti, S.T., M.T. dilahirkan di Gresik, Jawa Timur, 4 Februari 1978. Tamat SMA pada tahun 1995 kemudian melanjutkan pendidikan Strata 1 di Universitas Katolik Soegijapranata Semarang, pada program studi Teknik Elektro. Setelah lulus pendidikan S1, sempat bekerja di perusahaan otomotif PT. Tossa Shakti di Kaliwungu Kendal selama 7 bulan. Dan tidak lama setelah itu, bekerja sebagai dosen di Universitas Katolik Soegijapranata selama 10 tahun. Melanjutkan pendidikan Strata 2 di Magister Teknik Instrumentasi di Universitas Gadjah Mada Yogyakarta, yang ditempuh selama 2 tahun. Saat ini aktif sebagai pengajar di program studi S1 Teknik Elektro, Fakultas Teknik Telekomunikasi dan Elektro, di Institut Teknologi Telkom Purwokerto.