Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

Implementasi Algoritma Kriptografi RSA untuk Keamanan Transmisi Data pada Sistem Monitoring Energi Listrik Berbasis IoT

Rajawali*, Syamsul Bahri, Kasliono

Fakultas Matematika dan Ilmu Pengetahuan Alam, Prodi Rekayasa Sistem Komputer, Universitas Tanjungpura, Pontianak
Jl. Prof. Dr. Hadari Nawawi, Bansir Laut, Pontianak Tenggara, Kota Pontianak, Kalimantan Barat, Indonesia
Email: [1,*]h1051211029@student.untan.ac.id, [2]syamsul.bahri@siskom.untan.ac.id, [3]kasliono@siskom.untan.ac.id
Email Penulis Korespondensi: h1051211029@student.untan.ac.id
Submitted: **13/08/2025**; Accepted: **18/02/2026**; Published: **04/04/2026**

Abstrak−Keamanan data menjadi isu penting dalam sistem Internet of Things (IoT) yang digunakan untuk memantau konsumsi energi listrik. Penelitian ini bertujuan untuk meningkatkan keamanan transmisi data pada sistem monitoring energi listrik berbasis IoT melalui penerapan algoritma kriptografi Rivest–Shamir–Adleman (RSA). Data dari sensor PZEM-004T dienkripsi menggunakan kunci publik RSA dan diverifikasi menggunakan tanda tangan digital sebelum dikirim ke server. Pengujian dilakukan pada dua kondisi, yaitu tanpa dan dengan enkripsi RSA serta simulasi serangan ARP spoofing menggunakan Ettercap. Hasil menunjukkan bahwa sistem mampu menolak data manipulatif dengan nilai packet loss sebesar 2,08% yang masih tergolong "sangat baik" berdasarkan standar TIPHON, serta throughput sebesar ±9,88 bit/s. Penerapan RSA terbukti efektif menjaga integritas dan keaslian data, sekaligus meningkatkan keandalan sistem monitoring energi listrik berbasis IoT.

Kata Kunci: Kriptografi; RSA; Monitoring Listrik; Packet Loss; ASCII

Abstract−Data security is a crucial issue in Internet of Things (IoT) systems used to monitor electricity consumption. This study aims to enhance the security of data transmission in an IoT-based electricity monitoring system by implementing the Rivest–Shamir–Adleman (RSA) cryptographic algorithm. Data from the PZEM-004T sensor is encrypted using the RSA public key and verified with a digital signature before being transmitted to the server. The system was tested under two conditions: without encryption and with RSA encryption, including a simulated ARP spoofing attack using Ettercap. The results show that the system successfully rejected manipulated data, with a packet loss rate of 2.08%, which is categorized as "very good" based on the TIPHON standard, and achieved a throughput of approximately 9.88 bit/s. The implementation of RSA proved effective in maintaining data integrity and authenticity, thereby improving the reliability of the IoT-based electricity monitoring system.

Keywords: Cryptography; RSA; Electricity Monitoring; Packet Loss; ASCII

1. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) memungkinkan terjadinya komunikasi antar device melalui jaringan internet, sehingga berbagai aktivitas dapat dimonitor secara otomatis dan real-time. IoT telah diterapkan dalam berbagai bidang seperti pertanian, industri, transportasi, hingga manajemen energi [1]. Salah satu penerapan IoT adalah dalam pengelolaan energi listrik pada lingkungan kamar indekos, terutama untuk memantau konsumsi penggunaan energi listrik setiap penghuni [2]. Sistem monitoring energi listrik berbasis IoT dibutuhkan karena mampu mencatat penggunaan daya listrik secara aktual, sehingga pemilik dan penyewa dapat mengetahui konsumsi energi secara transparan dan akurat.

Sistem tagihan manual di kamar indekos kerap menimbulkan ketidakadilan bagi penghuni dengan konsumsi listrik berbeda. Penghuni yang menggunakan peralatan listrik lebih banyak justru membayar sama dengan penghuni yang penggunaan listriknya lebih sedikit. Hal ini dapat menimbulkan keluhan, konflik, dan kesalahpahaman antara penghuni maupun antara penghuni dan pemilik kamar indekos. Oleh karena itu, sistem monitoring energi listrik berbasis IoT menjadi solusi untuk menghadirkan sistem tagihan yang adil dan berbasis data aktual.

Namun, sistem berbasis IoT juga menghadirkan permasalahan baru, khususnya dalam hal keamanan data yang dikirim melalui jaringan. Pengiriman data dari sensor ke server yang dilakukan melalui jaringan lokal sering tidak dilengkapi enkripsi. Hal ini menimbulkan risiko terjadinya serangan seperti Man-in-the-Middle (MitM), yaitu jenis serangan di mana pelaku dapat menyusup ke dalam jalur komunikasi dan memanipulasi data sebelum sampai ke server [3]. Hal ini menjadi penting ketika analis keamanan menyadari bahwa sebanyak 98% lalu lintas antar perangkat IoT tidak terenkripsi [4], sehingga rawan terjadi manipulasi data penggunaan energi listrik.

Keamanan data menjadi aspek penting yang harus diterapkan pada sistem monitoring energi berbasis IoT. Salah satu pendekatan untuk menjamin kerahasiaan dan keaslian data adalah melalui algoritma kriptografi. Salah satu algoritma kriptografi yang banyak digunakan untuk pengamanan data adalah Rivest-Shamir-Adleman (RSA), yaitu algoritma kriptografi kunci publik yang memungkinkan proses enkripsi dilakukan menggunakan kunci publik dan hanya dapat didekripsi dengan kunci privat [5]. Dengan metode ini, data yang dikirimkan dari sensor ke server dapat diamankan sehingga hanya pihak tertentu yang berhak mengakses isi data tersebut. RSA juga mendukung proses verifikasi keaslian data melalui penerapan digital signature [6], yang digunakan untuk memastikan data benar-benar dikirim oleh perangkat asli dan tidak dimodifikasi di tengah jalan

Dalam sistem monitoring energi listrik ini, digunakan sensor PZEM-004T yang dapat mengukur tegangan (volt), arus (ampere), daya aktif (watt), serta energi total (kWh) [7]. Data hasil pengukuran ini dikirimkan oleh

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

mikrokontroler NodeMCU ESP32 ke server lokal menggunakan protokol HTTP. Untuk meningkatkan keamanan data, informasi yang dikirim akan dienkripsi menggunakan algoritma RSA. Data hasil enkripsi dikonversi ke format ASCII dan ditambahkan tanda tangan digital. Di sisi server, data akan divalidasi terlebih dahulu, kemudian didekripsi, dan disimpan ke dalam database hanya jika data tersebut valid.

Beberapa penelitian sebelumnya telah mengembangkan sistem monitoring energi listrik berbasis IoT, seperti Muhamad yang menggunakan sensor PZEM-004T dan aplikasi smartphone, namun belum menyertakan enkripsi data [8]. Selain itu, terdapat penelitian oleh Mbejo yang membahas ancaman keamanan IoT dan menyarankan solusi seperti enkripsi dan autentikasi, tetapi belum diimplementasikan dalam sistem monitoring [9]. Sementara itu, Akbar pada tahun 2024, telah membuktikan efektivitas RSA pada komunikasi sensor via MQTT, namun belum diterapkan dalam sistem energi listrik dan diuji terhadap serangan manipulasi data [10]. Penelitian Siregar juga menerapkan RSA untuk data gaji, tetapi masih dalam sistem internal non-IoT [11]. Adapun Zaatsiyah dan Djuniadi pada tahun 2021 menunjukkan bahwa digital signature berbasis RSA dan MD5 dapat menjamin keaslian data pada dokumen e-invoice melalui simulasi menggunakan perangkat lunak CrypTool [6].

Berdasarkan hasil tinjauan tersebut, dapat disimpulkan bahwa sistem monitoring energi listrik berbasis IoT masih memiliki celah dalam aspek keamanan, khususnya saat pengiriman data melalui jaringan yang rentan terhadap serangan seperti Man-in-the-Middle. Belum banyak penelitian yang mengintegrasikan algoritma kriptografi RSA dan digital signature secara menyeluruh dalam sistem monitoring energi listrik yang menggunakan perangkat keras nyata, serta mengujinya terhadap potensi serangan aktif seperti ARP spoofing.

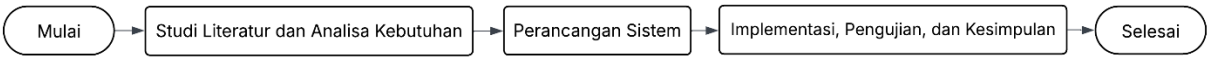
Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem monitoring energi listrik yang aman dan andal di lingkungan kamar indekos dengan menerapkan algoritma Rivest–Shamir–Adleman (RSA) untuk mengenkripsi data dari sensor PZEM-004T sebelum dikirim melalui NodeMCU ESP32 ke server lokal. Sistem dilengkapi dengan tanda tangan digital yang divalidasi sebelum data disimpan ke dalam basis data, serta dirancang untuk menolak data tidak valid dan mengirim ulang data pada siklus berikutnya guna menjaga integritas dan keaslian informasi. Berbeda dari penelitian sebelumnya, penelitian ini tidak hanya mengimplementasikan algoritma RSA pada sistem monitoring energi listrik, tetapi juga menguji ketahanannya terhadap serangan aktif seperti ARP spoofing. Pendekatan ini menunjukkan kontribusi nyata dalam meningkatkan aspek keamanan data pada sistem Internet of Things (IoT), khususnya pada penerapan di lingkungan kamar indekos, serta diharapkan dapat meningkatkan keamanan, keandalan, dan transparansi dalam sistem pencatatan dan penagihan energi listrik.

2. METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan pendekatan yang digunakan dalam merancang dan mengimplementasikan sistem monitoring penggunaan energi listrik yang aman selama proses transmisi data menggunakan algoritma kriptografi RSA. Pendekatan ini bertujuan untuk memastikan bahwa sistem tidak hanya berfungsi secara teknis, tetapi juga mampu menjaga integritas dan keaslian data selama transmisi.

2.1 Tahapan Penelitian

Tahapan penelitian diawali dengan studi literatur dan analisis kebutuhan perangkat keras dan lunak, seperti NodeMCU ESP32, modul PZEM-004T, Hi-Link HLK-PM01, Arduino IDE, Wireshark, dan IPerf3. Tahapan selanjutnya meliputi perancangan sistem, implementasi sistem, pengujian sistem menggunakan simulasi serangan ARP spoofing, dan menarik kesimpulan. Diagram alir proses penelitian disajikan pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

2.1.1 Studi Literatur dan Analisa Kebutuhan

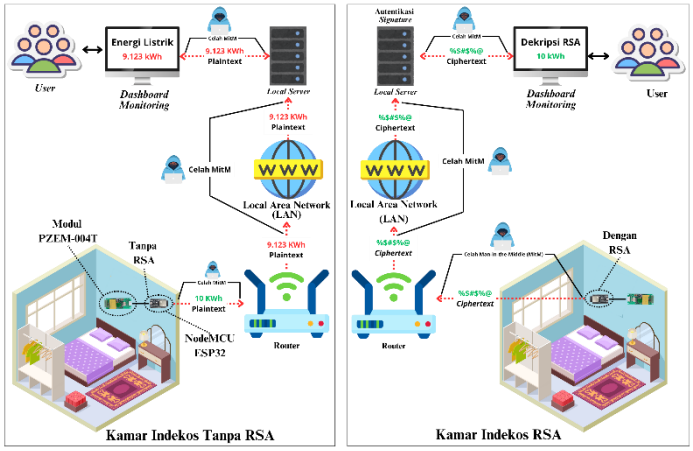
Studi literatur dilakukan untuk memperoleh pemahaman terkait keamanan data pada sistem IoT serta algoritma RSA dan ASCII. Dalam tahap analisis kebutuhan, ditentukan komponen-komponen yang dibutuhkan, yaitu NodeMCU ESP32 yang merupakan mikrokontroler berbasis Wi-Fi dan Bluetooth dengan kemampuan pemrosesan dan konektivitas tinggi yang cocok untuk aplikasi IoT [12], HI-LINK HLK-PM01 yang merupakan modul konversi AC-DC yang digunakan untuk menyuplai daya 5V ke mikrokontroler dari sumber AC 220V [13], Arduino IDE yang merupakan perangkat lunak yang digunakan untuk menulis, mengunggah, dan menjalankan program pada mikrokontroler [14], Wireshark yang merupakan aplikasi open-source untuk menganalisis lalu lintas jaringan [15], dan Iperf3 yang merupakan alat penguji performa jaringan yang digunakan untuk mengukur bandwidth [16]. IPerf3 digunakan untuk mengukur bandwidth maksimum jaringan yang tersedia selama proses pengujian sistem. Selain itu, untuk mengetahui nilai throughput yang dikirim oleh NodeMCU ESP32 ke database, dilakukan perhitungan menggunakan Persamaan (1) [17].

$$\text{Throughput} = \text{Packet data received (bit)} / \text{Total time to send packet (detik)} \quad (1)$$

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

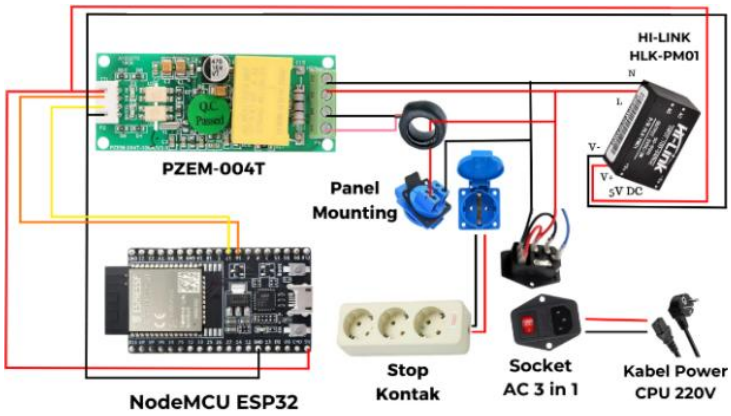
2.1.2 Perancangan Sistem

Sistem dirancang dengan dua skenario yaitu tanpa enkripsi menggunakan algoritma kriptografi RSA, di mana data dikirim langsung dalam bentuk plaintext ke server, dan kedua menggunakan algoritma kriptografi RSA dimana data dienkripsi, dikonversi ke format ASCII, dan disertai digital signature sebelum dikirim. Server melakukan verifikasi dan hanya menyimpan data yang valid. Arsitektur sistem ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Sistem

2.1.2.1 Perancangan Perangkat Keras



Gambar 3. Skema Perangkat Keras

Gambar 3 merupakan skema perangkat keras yang akan diimplementasikan, seluruh elemen perangkat keras seperti NodeMCU ESP32, modul PZEM-004T, dan HI-LINK HLK-PM01 akan diterapkan dan dihubungkan. Adapun penggunaan pin ESP32 dengan modul PZEM-004T dapat dilihat pada Tabel 1.

Tabel 1. Hubungan Pin Perangkat Keras

NodeMCU ESP32	Modul PZEM-004T
5v	VCC
GND	GND
GPIO 16	RX
GPIO 17	TX

2.1.2.2 Algoritma Kriptografi RSA

Algoritma kriptografi RSA menerapkan operasi pada bilangan bulat, karena nilai plainteks yang dienkripsi harus berada dalam rentang [0, 0-1], di mana n adalah hasil kali dari dua bilangan prima (p × q) [18]. Dalam algoritma RSA, terdapat tiga proses utama, yaitu pembangkitan pasangan kunci (publik dan privat), proses enkripsi, dan proses dekripsi [19]. Proses enkripsi dilaksanakan dengan memanfaatkan kunci publik, sedangkan dekripsi dilakukan dengan menggunakan kunci privat demi menjaga kerahasiaan data [20]. Langkah-langkah algoritma RSA mendapatkan kunci publik dan kunci privat adalah sebagai berikut.

1. Pilih dua buah bilangan prima p dan q.
2. Hitung nilai n dengan Persamaan (2).

$$n = p\ x\ q \tag{2}$$

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

3. Hitung nilai Totient euler (n) dengan Persamaan (3)

$$\phi(n) = (p-1)(q-1) \quad (3)$$

4. Pilih bilangan bulat untuk kunci publik (e) yang memenuhi persyaratan $gcd(e, \phi(n)) = 1$
5. Hitung kunci untuk dekripsi (d) dengan Persamaan (4).

$$d = 1 + k.\phi(n)/e \quad (4)$$

Langkah berikutnya setelah pembuatan kunci RSA adalah proses enkripsi. Persamaan yang digunakan untuk melakukan proses enkripsi pada algoritma RSA dapat dilihat pada Persamaan (5).

$$c = m^e (mod\ n) \quad (5)$$

Proses dekripsi dalam algoritma RSA hampir sama dengan proses enkripsi. Persamaan yang digunakan untuk melakukan proses dekripsi dalam algoritma RSA dapat dilihat pada Persamaan (6).

$$m = c^d (mod\ n) \quad (6)$$

Dalam konteks RSA, signature dibuat dengan memanfaatkan kunci privat, sedangkan verifikasinya dilakukan menggunakan kunci publik. Pada NodeMCU ESP32 signature dihasilkan dengan Persamaan (7).

$$signature = m^d\ mod\ n \quad (7)$$

Di sisi server, signature diverifikasi menggunakan kunci publik RSA (e dan n) menggunakan Persamaan (8).

$$m' = signature^e\ mod\ n \quad (8)$$

2.1.2.3 ASCII (American Standard Code for Information Interchange)

ASCII merupakan standar internasional untuk pengkodean karakter dan symbol, yang digunakan secara luas untuk mendukung proses komunikasi data secara digital [21]. Sebagai contoh, nilai 124 digunakan untuk merepresentasikan karakter "|". konversi bilangan ke karakter ASCII berbasis 128 menggunakan prinsip representasi bilangan berbasis posisi (positional notation) [22], dengan setiap digit dikonversi ke karakter ASCII dapat dihitung menggunakan Persamaan (9).

$$N = d_k \times 128^k + d_{k-1} \times 128^{k-1} + \cdots + d_0 \times 128^0 \quad (9)$$

Dari Persamaan (10) untuk membalikkan Encrypted Energy (ASCII) kembali ke Encrypted Energy (RSA) (nilai integer), perlu dilakukan proses kebalikan dari konversi basis-128 dengan Persamaan (10).

$$N = \sum_{i=0}^{k} (ascii[i] \times 128^{k-i}) \quad (10)$$

Adapun tabel ASCII dapat dilihat Tabel 2.

Tabel 2. ASCII

Dec	Hex	Oct	Char
0	0	000	NULL (null)
1	1	001	SOH (start of text)
2	2	002	STX (start of text)
…	…	…	…
126	7E	176	~
127	7F	177	DEL

2.1.2.4 Packet Loss

Packet loss adalah suatu ukuran yang menunjukkan keadaan dimana jumlah total paket yang hilang [17]. Perhitungan packet loss dapat dilakukan dengan menggunakan Persamaan (11).

$$PL = \frac{(packet\ data\ sent - packet\ data\ received)\ x\ 100\%}{packet\ data\ sent} \quad (11)$$

Perhitungan packet loss dilakukan berdasarkan data log pengiriman yang dicatat oleh NodeMCU ESP32 sebagai packet data sent dan hasil pencatatan di server database sebagai packet data received. Dengan demikian, perbandingan antara jumlah data yang dikirim dan data yang berhasil tersimpan ke database digunakan untuk menghitung nilai packet loss. Performa jaringan berdasarkan standar TIPHON (Telecommunications and Internet Protocol Harmonization Over Networks) untuk parameter packet loss dapat dilihat pada Tabel 3 [23].

Tabel 3. Kategori Packet Loss Jaringan Berdasarkan Standar TIPHON

Kategori Packet Loss	Packet Loss (%)	Indeks
Sangat Bagus	0 – 2	4
Bagus	3 – 14	3

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

Kategori Packet Loss	Packet Loss (%)	Indeks
Sedang	15 – 24	2
Buruk	> 25	1

2.1.3 Implementasi, Pengujian, dan Kesimpulan

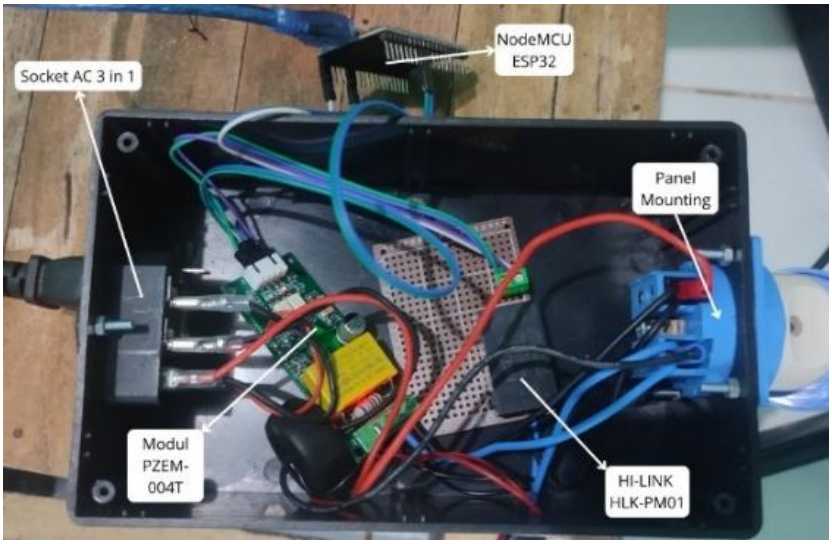
Setelah sistem selesai dirancang, dilakukan implementasi perangkat keras dan lunak. Sistem diuji dengan mengamati hasil enkripsi, pengiriman data, serta validasi oleh server. Selain itu, dilakukan simulasi serangan ARP spoofing untuk mengevaluasi ketahanan sistem terhadap manipulasi data dan menganalisis packet loss. Hasil pengujian digunakan untuk menarik kesimpulan dan memberikan saran pengembangan selanjutnya.

3. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil implementasi dan pengujian sistem monitoring energi listrik berbasis IoT yang dilengkapi dengan algoritma kriptografi RSA. Pembahasan dimulai dari penerapan perangkat keras dan lunak, dilanjutkan dengan pengujian sistem tanpa dan dengan enkripsi data. Seluruh hasil pengujian dianalisis untuk menilai efektivitas sistem dalam menjaga integritas dan keaslian data selama proses transmisi.

3.1 Implementasi Perangkat Keras

Hasil implementasi perangkat keras yang dibuat sesuai dengan skema perangkat keras pada Gambar 3, dapat dilihat pada Gambar 4.



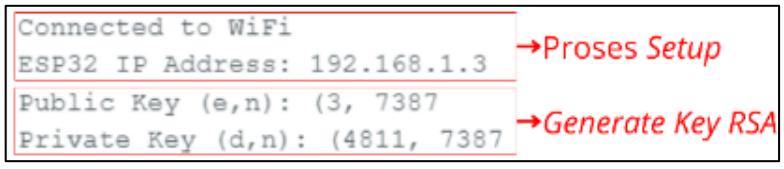
Gambar 4. Implementasi Sistem Monitoring Penggunaan Energi Listrik

3.2 Implementasi Perangkat Lunak

Pembahasan mengenai penerapan perangkat lunak dalam penelitian ini dibagi menjadi dua aspek, yaitu penerapan perangkat lunak pada NodeMCU ESP32, serta penerapan perangkat lunak pada server lokal untuk melaksanakan autentikasi, dekripsi, dan menampilkan data pada dashboard monitoring.

3.2.1 Implementasi Pada NodeMCU ESP32

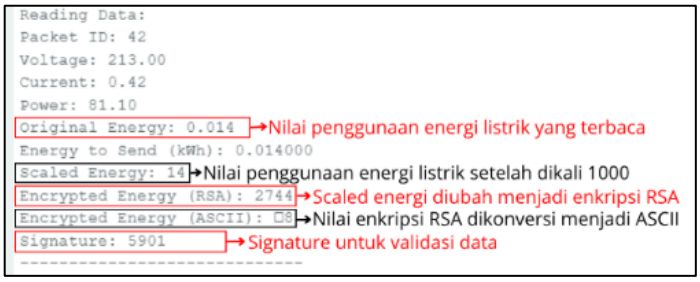
Terdapat tiga langkah utama dalam implementasi perangkat lunak pada NodeMCU ESP32, yaitu pengambilan data dari modul PZEM-004T, proses enkripsi data, dan pengiriman data ke server lokal. Ketiga proses ini akan dilaksanakan dalam void loop dengan selang waktu selama 10 menit. Diawali dengan proses pengaturan pada NodeMCU ESP32. Setelah pengaturan selesai, NodeMCU ESP32 akan melakukan proses pembuatan kunci RSA, kemudian dilanjutkan dengan pembacaan data dari modul PZEM-004T. Sebelum data dienkripsi, data tersebut akan dikalikan dengan 1000 untuk menghasilkan data dalam bilangan bulat, karena nilai plainteks yang akan dienkripsi harus berada dalam rentang [0, 0-1]. Proses setup dan generate key RSA dapat dilihat pada Gambar 5.



Gambar 5. Proses Pengaturan NodeMCU ESP32 dan Generate Key RSA

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

Setelah pengaturan selesai, NodeMCU ESP32 akan membaca data yang diperoleh dari modul PZEM-004T. Data penggunaan energi listrik yang telah dikalikan 1000 selanjutnya akan dienkripsi dengan menggunakan kunci publik RSA. Hasil enkripsi tersebut kemudian dikonversi menjadi format ASCII, dan diperoleh signature RSA yang dihasilkan dengan kunci privat. Hasil enkripsi, konversi, dan signature dapat dilihat pada Gambar 6.



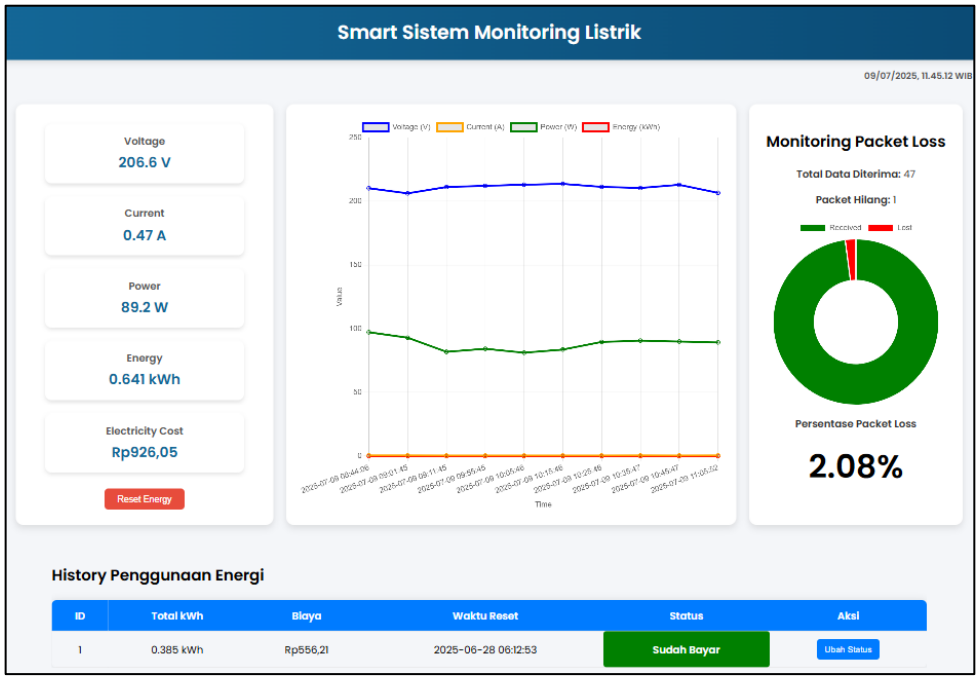
Gambar 6. Enkripsi, Konversi, dan Signature

Data yang telah didapatkan dikirim ke server lokal untuk disimpan dalam database. Jika proses ini berhasil, maka data penggunaan energi listrik akan direset. Namun, jika proses ini gagal, data penggunaan energi listrik akan disimpan sementara sampai pengiriman data berhasil. Berikut proses pengiriman data dari NodeMCU ESP32 ke server lokal yang dapat dilihat pada Gambar 7.

Gambar 7. Proses Pengiriman Data

3.2.2 Implementasi pada Server Lokal

Terdapat lima langkah utama dalam pelaksanaan perangkat lunak di server lokal, yaitu memverifikasi data yang diterima di server lokal dengan kunci signature RSA, mengubah ASCII menjadi enkripsi RSA, mendekripsi RSA menggunakan kunci privat, setelah data didekripsi selanjutnya data dibagi dengan 1000 untuk memperoleh data asli (plaintext) mengenai penggunaan energi listrik. Setelah itu, server lokal juga akan menghitung packet loss selama proses pengiriman data. Apabila seluruh langkah telah diselesaikan, maka informasi akan ditampilkan pada dashboard monitoring. Dashboard monitoring dapat dilihat pada Gambar 8.



Gambar 8. Tampilan Dashboard Monitoring

Dari grafik monitoring packet loss pada dashboard monitoring selama pengujian, diketahui bahwa terdapat 1 paket yang hilang karena sebelumnya dilakukan penyerangan ARP spoofing, sehingga pengujian performa jaringan terhadap packet loss menunjukkan bahwa sistem mampu bekerja dengan baik dengan kehilangan 1 data selama transmisi karena penyerangan. Packet loss dihitung dengan Persamaan (11).

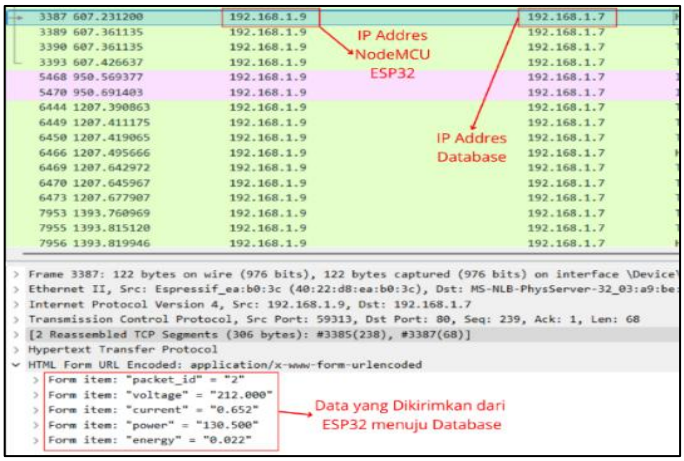
Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

$$Packet\ Loss = \frac{(48-47)\ x\ 100\%}{48} = 2{,}08\%$$

Berdasarkan standar TIPHON pada Tabel 2, Nilai packet loss sebesar 2,08% termasuk kategori Sangat Baik berdasarkan standar TIPHON. Namun, nilai ini tidak hanya merepresentasikan kualitas jaringan, melainkan juga keberhasilan sistem dalam mendeteksi dan menolak data hasil manipulasi akibat serangan ARP spoofing. Dari log pengujian, paket yang hilang bukan disebabkan oleh gangguan fisik jaringan, melainkan oleh mekanisme keamanan yang menolak paket dengan tanda tangan digital tidak valid. Dengan demikian, tingkat packet loss dapat berfungsi sebagai indikator dini adanya aktivitas penyerangan terhadap sistem IoT. Ini menunjukkan bahwa sistem tidak hanya mampu mentransmisikan data secara andal, tetapi juga mengintegrasikan aspek deteksi intrusi melalui validasi kriptografis.

3.3 Pengujian Tanpa Enkripsi Data

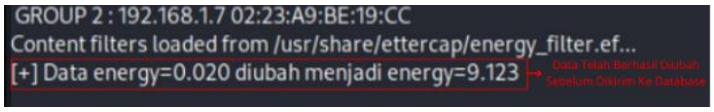
Pengujian tanpa enkripsi data dilakukan dengan mengirimkan data yang tidak melalui proses kriptografi RSA. Tujuan dari pengujian ini adalah untuk mengamati format data yang dikirimkan pada saat tidak mengalami proses enkripsi menggunakan algoritma kriptografi RSA. Untuk hasil dari pengujian ini data di capture menggunakan Wireshark yang dapat dilihat pada Gambar 9.



Gambar 9. Hasil Capture Data Tanpa Enkripsi

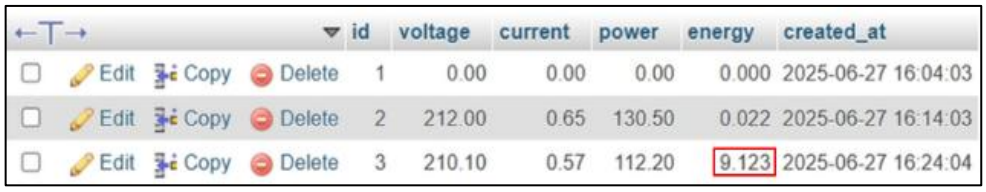
Gambar 9 menunjukkan bahwa data dari NodeMCU ESP32 ke database dikirim dalam bentuk plaintext dan terbaca jelas menggunakan Wireshark. Hal ini memungkinkan pelaku melakukan serangan Man-in-the-Middle (MitM), seperti ARP spoofing, yang bekerja dengan memalsukan alamat MAC agar data terlebih dahulu melewati perangkat penyerang. Dalam simulasi ini, Ettercap digunakan di sistem Kali Linux berbasis Virtual Box untuk memanipulasi data penggunaan energi listrik. Hasil tangkapan Wireshark menunjukkan data penting seperti IP NodeMCU, IP database, packet ID, tegangan, arus, daya, dan energi dapat diakses dan dimodifikasi oleh pihak tidak berwenang.

Informasi yang dibutuhkan untuk melakukan serangan ARP spoofing dari tools Wireshark berupa IP address dari NodeMCU ESP32, IP address dari database, dan data energy. Selama proses penyerangan, akan dibuat suatu filter yang digunakan untuk mengubah data penggunaan energi listrik dari 0,009 hingga 0,045 menjadi 9,123. Hasil dari serangan ini menunjukkan bahwa data penggunaan energi listrik 0.020 berhasil diubah menjadi 9.123, yang dapat dilihat pada Gambar 10.



Gambar 10. Hasil Perubahan Data

Serangan yang dilakukan berhasil sehingga data yang tercatat pada database akan berubah yang seharusnya 0.020 menjadi 9.123. Hasil manipulasi data penggunaan energi listrik dapat dilihat pada Gambar 11.



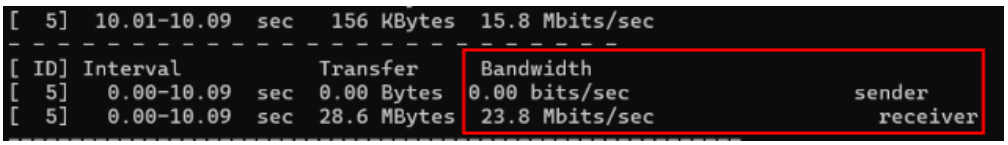
	id	voltage	current	power	energy	created_at
Edit Copy Delete	1	0.00	0.00	0.00	0.000	2025-06-27 16:04:03
Edit Copy Delete	2	212.00	0.65	130.50	0.022	2025-06-27 16:14:03
Edit Copy Delete	3	210.10	0.57	112.20	9.123	2025-06-27 16:24:04

Gambar 11. Hasil Manipulasi Data Penggunaan Energi Listrik Tanpa Enkripsi Data

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

3.4 Pengujian Dengan Enkripsi Data

Pengujian sistem monitoring listrik dengan RSA diawali dengan proses enkripsi data yang kemudian dikonversi ke format ASCII sebelum dikirim ke database. Proses ini membuat data sulit dimengerti atau dimanipulasi tanpa mengetahui algoritma konversi dan nilai signature RSA yang sesuai, karena sistem hanya menerima data yang tervalidasi. Sebelum pengujian, bandwidth jaringan diukur menggunakan IPerf3 dan menunjukkan kapasitas 23,8 Mbps yang dapat dilihat pada Gambar 12.



Gambar 12. Bandwith yang Digunakan

Selama pengujian didapatkan data berapa byte yang digunakan oleh NodeMCU ESP32 pada Wireshark. Berdasarkan hasil pengamatan yang ditampilkan pada Gambar 13, total data yang dikirimkan dari NodeMCU ESP32 ke server lokal adalah sebesar 21.504 byte, dengan durasi pengiriman selama 17.414 detik.

Ethernet · 1 | IPv4 · 1 | IPv6 | TCP · 31 | UDP

Address A	Address B	Packets	Bytes	Stream ID	Total Packets	Percent Filtered	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.180.16	192.168.180.113	193	21 kB	5	350	55.14%	193	21 kB	0	0 bytes	23.312454	17414.4109	9 bits/s	0 bits/s

Gambar 13. Data Berapa Byte yang Digunakan NodeMCU ESP32

Untuk mengetahui seberapa efisien sistem dalam penggunaan jaringan berdasarkan bandwidth yang disediakan, dilakukan perhitungan throughput dengan menggunakan Persamaan (1).

$$Throughput = 21.504 \times 8/17.414 = 172.032/17.414 \approx 9.878\ bit/s$$

Nilai throughput yang diperoleh sebesar ±9,88 bit/s menunjukkan bahwa jumlah data yang benar-benar dikirim dari NodeMCU ESP32 ke server relatif kecil dibandingkan kapasitas jaringan yang tersedia, yaitu bandwidth 23,8 Mbps. Hal ini menandakan bahwa beban lalu lintas data sistem sangat rendah, sehingga jaringan masih memiliki kapasitas yang jauh lebih besar daripada kebutuhan aktual sistem monitoring ini. Dengan demikian, packet loss yang terjadi tidak disebabkan oleh keterbatasan bandwidth atau kualitas sinyal jaringan, melainkan oleh mekanisme keamanan sistem yang secara sengaja menolak paket hasil manipulasi akibat serangan ARP spoofing.

Selanjutnya dilakukan proses enkripsi yang diawali dengan mengubah nilai energi asli menjadi scaled energy, yaitu nilai energi yang dikalikan 1000 agar menjadi bilangan bulat, misalnya diperoleh nilai yang awalnya 0.014 menjadi 14. Nilai ini kemudian dienkripsi menggunakan algoritma RSA, yang membutuhkan dua bilangan prima (p = 83 dan q = 89) untuk menghasilkan kunci RSA. Setelah itu, nilai n (hasil perkalian dua bilangan prima) akan diperoleh dengan Persamaan (2).

$$n = 83\ x\ 89 = 7387$$

Setelah mendapatkan nilai n, maka dilanjutkan dengan mencari totient euler (n) dari Persamaan (3)

$$\phi(n) = (83 - 1)(89 - 1) = 7216$$

Setelah menentukan nilai totient Euler ϕ(n), kunci publik e dipilih sebagai bilangan bulat yang relatif prima terhadap ϕ(n), yang dapat dinyatakan secara matematis yaitu gcd(e, ϕ(n)) = 1. Dengan demikian, diperoleh nilai e yaitu 3, 5, 7, 13, . . ., 7213 Karena nilai e mempengaruhi waktu komputasi dalam RSA, nilai e yang disarankan adalah 3. Langkah selanjutnya adalah menghitung kunci dekripsi (d) dengan menggunakan Persamaan (4).

$$d = 1 + k.7216/3 = 4811$$

Setelah nilai d diperoleh, langkah-langkah untuk menghasilkan kunci RSA telah diselesaikan, sehingga diperoleh kunci publik pasangan (e,n) yaitu (3,7387) dan pasangan kunci privat (d,n) yaitu (4811,7387). Setelah itu, dilakukan proses enkripsi plaintext (scaled energy) menggunakan Persamaan (5).

$$14^3 (mod\ 7387) = 2744$$

Setelah hasil enkripsi RSA diperoleh, langkah berikutnya adalah mengubah hasil enkripsi tersebut menjadi format ASCII menggunakan Persamaan (9).

$$2744 = 21 \times 128^1 + 56 \times 128^0$$

Berdasarkan hasil perhitungan konversi dari RSA ke ASCII, didapatkan bahwa karakter pertama adalah 21, yang dalam tabel ASCII merepresentasikan negative acknowledge (NAK), sedangkan karakter kedua adalah 56, yang merepresentasikan angka 8 dalam tabel ASCII. Dengan demikian, hasil enkripsi yang telah dikonversi

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

menjadi bentuk ASCII adalah "NAK8", karena karakter 0 hingga 32 dalam tabel ASCII tergolong sebagai karakter non-printable, yaitu karakter yang tidak dapat ditampilkan sebagai simbol visual yang terlihat, sehingga akan tampil seperti berikut pada serial monitor Arduino IDE dan database □8. Setelah memperoleh nilai konversi ASCII, selanjutnya melakukan perhitungan untuk menentukan nilai signature menggunakan Persamaan (7).

$$signature = 14^{4811} \, mod \, 7387 = 5901$$

Data terenkripsi yang ditampilkan pada serial monitor Arduino IDE kemudian dikirim ke database, mencakup packet ID, tegangan, arus, daya, energi terenkripsi (ASCII), dan signature. Sebelum penyimpanan, sistem memvalidasi kecocokan hasil dekripsi terhadap signature RSA. Jika valid, data disimpan ke database dan jika tidak, data ditolak dan akan diperbarui pada interval pengiriman berikutnya. Proses dekripsi diawali dengan konversi dari ASCII ke bentuk enkripsi RSA menggunakan Persamaan (10).

$$N = ascii(0) \times 128^1 + ascii(1) \times 128^0$$

$$N = 21 \times 128^1 + 56 \times 128^0 = 2744$$

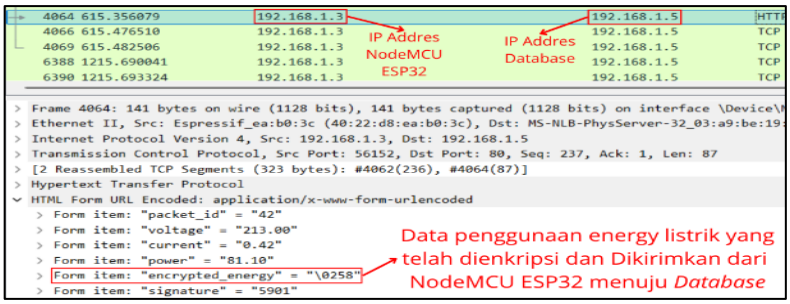
Setelah memperoleh hasil konversi dari ASCII ke RSA, langkah berikutnya adalah mendekripsi RSA menjadi informasi penggunaan energi listrik. Hasil perhitungan dekripsi dari RSA ke data penggunaan energi listrik dilakukan dengan menggunakan Persamaan (7).

$$m = 2744^{4811} (mod \, 7387) = 14$$

Selanjutnya dilakukan perhitungan verifikasi signature RSA menggunakan Persamaan (8).

$$m' = 5901^3 \, mod \, 7387 = 14$$

Hasil perhitungan dekripsi RSA dan verifikasi signature dinyatakan valid karena datanya sama, sehingga data tersebut akan disimpan dalam database. Ketika data dikirim ke database, telah di capture dengan Wireshark. Hasil capture menggunakan Wireshark dapat dilihat pada Gambar 14.



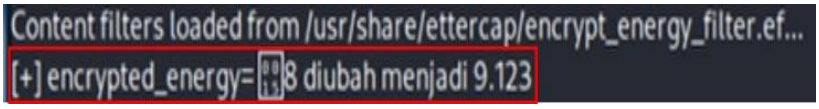
Gambar 14. Hasil Capture Wireshark

Selanjutnya, akan dilakukan serangan ARP spoofing. Langkah-langkah untuk melakukan serangan perubahan data enkripsi yang dikirim dari ESP32 ke database di server lokal adalah dengan melakukan capture menggunakan Wireshark. Dari hasil capture, akan digunakan 4 data sebagai sampel untuk melakukan perubahan data. Hasil dari capture 4 data yang berbeda dapat dilihat pada Tabel 4.

Tabel 4. Capture Data Energy Listrik

Packet ID	Encrypted Energy	Hasil capture
41	}	Form item: "encrypted_energy" = "}" Form item: "signature" = "6073"
42	\0258	Form item: "encrypted_energy" = "\0258" Form item: "signature" = "5901"
43	\021\025	Form item: "encrypted_energy" = "\021\025" Form item: "signature" = "2129"
46	\032/	Form item: "encrypted_energy" = "\032/" Form item: "signature" = "2478"

Pada Tabel 4, terlihat bahwa hasil capture terakhir untuk sampel berada di Packet ID 46. Maka, akan dibuat terlebih dahulu filter Ettercap sebelum melakukan proses serangan yang dimulai dari data Packet ID 47. Hasil serangan dengan menggunakan Ettercap dapat dilihat pada Gambar 15.



Gambar 15. Hasil Penyerangan Data

Proses serangan dimulai dari Packet ID 47, dan diperoleh hasil tangkapan Packet ID 47 encrypted_energy yang sama dengan yang tertera pada filter Ettercap yaitu \0258, namun data tersebut tidak tersimpan dalam

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

database, karena energi yang terenkripsi (ASCII) dan signature RSA tidak valid, sehingga mengakibatkan serangan pada data penggunaan energi listrik yang telah dienkripsi tidak berhasil. Data penggunaan energi listrik yang terenkripsi dan diperoleh selama proses serangan dapat dilihat pada Tabel 5.

Tabel 5. Data Penggunaan Energi Listrik yang Dienkripsi dan Didapatkan Selama Proses Penyerangan

Packet ID	Terbaca Pada Serial Monitor	Capture pada Wireshark	Status Data
41	}	}	Belum Diserang
42	□8	\0258	Belum Diserang
43	□|	\021\025	Belum Diserang
44	□/	\032/	Belum Diserang
45	□/	\032/	Belum Diserang
46	□/	\032/	Belum Diserang
47	□8	9.123	Diserang, Tetapi Data Gagal Disimpan Didalam Database
48	%g		Data Hasil Penjumlahan Dengan Data yang Diserang Sebelumnya

Pengujian kriptografi dilakukan sebanyak 30 kali untuk menilai konsistensi proses enkripsi dan dekripsi data energi listrik. Tujuan dari pengujian ini adalah untuk memastikan bahwa informasi dapat dienkripsi dan didekripsi dengan tepat selama proses transmisi data. Data dikirim ke server lokal setiap 10 menit. Hasil dari pengujian disajikan pada Tabel 6.

Tabel 6. Pengujian Enkripsi dan dekripsi pada Sistem

Packet Id	Encrypted (ASCII)	Decrypted Energy	Data energy Seharusnya	Signature
1	□	0.046	0.046	3178
2	□□	0.047	0.047	6426
...	...	...	...	...
29	□R	0.021	0.021	3926
30	□□	0.013	0.013	2129

Dari Tabel 6, diketahui bahwa semua data berhasil dienkripsi (dalam format ASCII), kemudian didekripsi kembali menjadi nilai energi, dan dikonfirmasi menggunakan signature RSA. Kesesuaian antara data yang telah dienkripsi, hasil dekripsi, dan data energi menunjukkan bahwa metode kriptografi telah diterapkan dengan akurat dan konsisten.

4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penerapan algoritma RSA pada sistem monitoring listrik di kamar indekos berhasil menjaga keamanan pengiriman data melalui proses enkripsi, autentikasi, dan dekripsi. Data yang diperoleh dari sensor PZEM-004T dienkripsi menggunakan kunci publik RSA sebelum dikirim ke server, sehingga memastikan kerahasiaan dan integritas data. Selain itu, sistem ini juga mampu mencegah manipulasi data, termasuk serangan Man-in-the-Middle (MitM) yang diuji menggunakan Ettercap. Dengan demikian, penggunaan kriptografi asimetris seperti RSA terbukti efektif dalam mengamankan komunikasi data dalam jaringan IoT, khususnya dalam konteks pengawasan konsumsi energi listrik. Selain aspek keamanan, penelitian ini juga menguji keandalan jaringan dengan mengukur tingkat packet loss, yang hasilnya sebesar 2,08% termasuk dalam kategori "Sangat Bagus" berdasarkan standar TIPHON. Namun, packet loss dalam sistem ini tidak disebabkan oleh gangguan jaringan, tetapi oleh penolakan server terhadap data yang tidak valid akibat penerapan RSA. Hal ini menunjukkan bahwa packet loss dapat menjadi indikator awal adanya upaya serangan atau manipulasi data. Meskipun demikian, sistem tetap menjaga integritas data dengan mengirim ulang paket yang hilang, sehingga akurasi dan keandalan sistem tidak terganggu. Namun, penelitian ini memiliki beberapa keterbatasan, antara lain belum adanya fitur pencatatan log serangan atau notifikasi otomatis saat terjadi ancaman keamanan. Selain itu, antarmuka pemantauan masih dapat ditingkatkan dengan visualisasi data yang lebih interaktif, seperti grafik konsumsi harian, mingguan, atau bulanan. Keterbatasan ini membuka peluang pengembangan lebih lanjut, seperti integrasi sistem deteksi intrusi atau pengiriman notifikasi real-time kepada pengguna. Dengan demikian, penelitian ini tidak hanya memberikan solusi keamanan yang efektif tetapi juga menjadi landasan bagi inovasi-inovasi berikutnya dalam pengembangan sistem monitoring energi listrik yang lebih canggih dan aman.

REFERENCES

[1] B. Baharuddin, J. W. Sitopu, M. Sigid Safarudin, Muh. W. S. Adam, and Muh. Safar, “Copyright: Mengenal Internet of Things (IoT): Penerapan Konsep dan Manfaatnya dalam Kehidupan Sehari-hari,” Journal of Human And Education, vol. 4, no. 4, p. 827, 2024, doi: https://doi.org/10.31004/jh.v4i4.1348.

Journal of Information System Research (JOSH)
Volume 7, No. 3, April 2026, pp 622−632
ISSN 2686-228X (media online)
https://ejurnal.seminar-id.com/index.php/josh/
DOI 10.47065/josh.v7i3.8232

[2] D. Aryo Atmanto, R. Wardana Nanditama, W. Suteddy, and A. Adiwilaga, "Sistem Monitoring Konsumsi Energi Listrik Berbasis IoT Menggunakan Fuzzy Logic Mamdani," TELKA, vol. 11, no. 2, pp. 151–166, 2025, doi: https://doi.org/10.15575/telka.v11n2.151-166.
[3] A. Aman, "Pengujian Keamanan Jaringan Nirkabel Melalui Simulasi Serangan Man In The Middle Attack Di Sekolah XYZ," Digital Transformation Technology, vol. 3, no. 2, pp. 824–831, Dec. 2023, doi: 10.47709/digitech.v3i2.3378.
[4] H. Fereidouni, O. Fadeitcheva, and M. Zalai, "IoT and Man-in-the-Middle Attacks," 2023, doi: https://doi.org/10.48550/arXiv.2308.02479.
[5] H. Putri, L. Virna, T. Febrianti, and T. Sutabri, "Pengamanan Data Transmisi Aplikasi Web Menggunakan Algoritma Kriptografi RSA: Studi Kasus dan Analisis," MIFORTEKH (Jurnal Manajemen Informatika & Teknologi) , vol. 5, no. 1, pp. 153–170, 2025, doi: https://doi.org/10.51903/rdbsne23.
[6] N. Zaatsiyah and Djuniadi, "Implementing Digital signature with RSA and MD5 in Securing E-Invoice Document," Jurnal Pendidikan Teknologi Informasi, vol. 5, pp. 129–140, Oct. 2021, doi: 10.22373/cj.v5i2.10359.
[7] R. Riza Ibrahim and B. Yulianti, "Rancang Bangun Monitoring Pemakaian Arus Listrik PLN Berbasis IoT," JURNAL TEKNOLOGI INDUSTRI, pp. 43–51, 2022, doi: https://doi.org/10.35968/jti.v11i2.953.g926.
[8] H. P. Muhamad, E. Susanto, and A. S. Wibowo, "Perancangan Alat Sistem Monitoring Energi Listrik Kos-kosan Berbasis Internet of Things (IoT)," eProceedings of Engineering, vol. 8, no. 5, pp. 4377–4388, 2021.
[9] M. T. Mbejo and I. Sufian, "ANALISIS TANTANGAN KEAMANAN JARINGAN IOT DAN STRATEGI MITIGASINYA," Jurnal Responsive Teknik Informatika, vol. 9, no. 1, 2025, doi: https://doi.org/10.36352/jr.v9i01.1178.
[10] R. Akbar, S. Bahri, and I. Nirmala, "Implementasi Algoritma RSA untuk Proses Enkripsi-Autentikasi Publish-Subscribe pada Protokol MQTT Menggunakan ESP8266 Berbasis IoT," Coding : Jurnal Komputer dan Aplikasi, pp. 23–32, 2024, doi: https://dx.doi.org/10.26418/coding.v12i1.70151.
[11] S. J. Siregar, N. B. Nugroho, and H. Sigalingging, "Implementasi Algoritma Kriptografi RSA (Rivest Shamir Adleman) Dalam Pengamanan Data Gaji Karyawan Di Kantor BSPJI," Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer), vol. 22, pp. 528–538, 2023, doi: https://doi.org/10.53513/JIS.V22I2.9409.
[12] P. Nur Setiawati and Fitriyani, "SISTEM MONITORING REALTIME KUALITAS AIR BERBASIS IOT DENGAN SENSOR TDS DAN NODEMCU ESP32," JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang , vol. 9, no. 3, pp. 2722–2713, 2025, doi: http://dx.doi.org/10.31000/jika.v9i3.14442.
[13] R. Wicaksono, R. Jasa Kusumo Haryo, A. Ravi Dwi Santoso, E. Dwi Wibowo, and P. Ridho Hanafi, "Perancangan Time Synchronization Sebagai Alat Bantu Pengujian Intertrip Relay Distance," Jurnal JEETech, vol. 5, no. 2, pp. 151–163, Oct. 2024, doi: 10.32492/jeetech.v5i2.5206.
[14] Kamal, Firdayanti, U. M. Tyas, A. A. Buckhari, and Pattasang, "Implementasi Aplikasi Arduino IDE pada Mata Kuliah Sistem Digital," JURNAL PENDIDIKAN DAN TEKNOLOGI, vol. 1, no. 1, pp. 1–10, 2023, doi: https://doi.org/10.59638/teknos.v1i1.40.
[15] Z. M. Luthfansa and U. D. Rosiani, "Pemanfaatan Wireshark untuk Sniffing Komunikasi Data Berprotokol HTTP pada Jaringan Internet," JIEET (Journal Information Engineering and Educational Technology), vol. 5, no. 1, pp. 34–39, 2021, doi: https://doi.org/10.26740/jieet.v5n1.p34-39.
[16] Y. Winawang, "Implementasi Keamanan Jalur Internet Menggunakan IP Tunneling pada OpenVPN Access Server dengan Protokol OpenVPN dan Protokol DNS Over HTTPS," Jurnal Syntax Admiration, vol. 2, no. 4, pp. 712–730, Apr. 2021, doi: 10.46799/jsa.v2i4.207.
[17] D. Junesco, E. Supriyanto, A. Hasan, and M. Mukhlisin, "QoS analysis of WSN (Wireless Sensor Network) using node MCU and accelerometer sensors on bridge monitoring systems," IOP Conf Ser Mater Sci Eng, vol. 1108, no. 1, p. 012025, Mar. 2021, doi: 10.1088/1757-899x/1108/1/012025.
[18] Sujarwo, "Analisis Kunci Algoritma Rivest Shamir Adleman," Remik: Riset dan E-Jurnal Manajemen Informatika Komputer, vol. 8, no. 3, pp. 772–778, 2024, doi: https://doi.org/10.33395/remik.v8i3.13787.
[19] M. S. Dairi, M. Setiani Asih, and Khairunnisa, "Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan," JurnalIlmuKomputerdanSistem Informasi (JIRSI), vol. 2, no. 1, pp. 98–107, 2023, doi: https://doi.org/10.70340/jirsi.v2i1.44.
[20] M. SD. Dairi, M. S. Asih, and Khairunnisa, "Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan Implementation Of RSA Cryptographic Algorithms in Library Information System Applications," Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI), vol. 2, no. 1, pp. 214–223, 2023, doi: https://doi.org/10.70340/jirsi.v2i1.44.
[21] E. Yuslita Dewi, D. Aricho Sundawa, B. Hermansyah, and A. Nur Azizah, "CAESAR TIME ASCII CIPHER (CAESAR CIPHER VERSI WAKTU + ASCII)," in Prosiding SNPM (Seminar Nasional Pendidikan Matematika), 2025, pp. 576–582. Accessed: Oct. 07, 2025. [Online]. Available: https://snpm.unipasby.ac.id/prosiding/index.php/snpm/article/view/361
[22] M. H. Widianto, "Konversi Bilangan," binus.ac.id. Accessed: Oct. 07, 2025. [Online]. Available: https://binus.ac.id/bandung/2019/12/konversi-bilangan/
[23] M. S. Rafinaldo, I. Iskandar, N. S. Harahap, and R. M. Candra, "Analisis Kualitas Jaringan Internet pada SMK Menggunakan Metode Quality of Service," KLIK: Kajian Ilmiah Informatika dan Komputer, vol. 3, no. 6, pp. 977–984, 2023, doi: 10.30865/klik.v3i6.903.