Implementasi Pengamanan Pada Jaringan LoRaWAN Untuk Mengatasi Serangan *Sniffing* Dengan Menggunakan Metode *Digital Signature*

Rahayu Indah Lestari[1], Vera Suryani[2], Aulia Arif Wardana[3]

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]rahayuindahlestarii@student.telkomuniversity.ac.id, [2]verasuryani@telkomuniversity.ac.id,
[3]auliawardan@telkomuniversity.ac.id

Abstrak

LoRa atau *Long Range* dengan teknologi LoRaWAN adalah contoh baru dari kelas jaringan dan sedang digunakan dalam skala besar di beberapa negara. Kekurangan dari jaringan LoRaWAN yaitu tidak adanya proses enkripsi pada *payload data*. Saat proses pengiriman pesan sedang berjalan antar perangkat, *sniffing process* dapat mengetahui semua pesan yang dikirim maupun yang diterima oleh perangkat, sehingga besar kemungkinan dapat terjadi serangan dari proses *sniffing* tersebut yang menyebabkan tidak adanya privasi dalam *payload data*. Penelitian ini menggunakan metode *digital signature* untuk mengamankan pesan yang dikirimkan oleh perangkat jaringan LoRaWAN menggunakan algoritma *Advanced Encryption Standard* (AES) pada proses enkripsi dan dekripsi pesan, serta menggunakan algoritma Ed25519 pada proses *signature*, penelitian ini juga menganalisis *overhead* dari penerapan metode *digital signature*. Tujuan penerapan *digital signature* dalam sistem ini adalah untuk memverifikasi bahwa *payload data* yang dikirim adalah asli dan tidak berubah selama proses transmisi, serta menjamin kerahasiaan pada *payload data*. Penambahan mekanisme keamanan pada jaringan LoRaWAN seperti proses enkripsi, dekripsi dan hasil verifikasi telah menghasilkan *overhead* pada beberapa aspek. Setelah dilakukan analisis, terdapat peningkatan ukuran untuk beberapa aspek saat diterapkannya metode *digital signature*.

Kata kunci : LoRaWAN, sniffing, digital signature, AES, overhead.

Abstract

LoRa or Long Range with LoRaWAN technology is a new example of a network class and is being used on a large scale in several countries. The weakness of the LoRaWAN network is that there is no encryption process in the data payload. When the process of sending messages is running between devices, the sniffing process can find out all the messages sent and received by the device, so there is a big possibility that attacks can occur from the sniffing process that causes no privacy in the data payload. This study uses a digital signature method to secure messages sent by LoRaWAN network devices using the Advanced Encryption Standard (AES) algorithm in the message encryption and decryption process, and using the Ed25519 algorithm in the signature process, this study also analyzes the overhead of applying the digital signature method. The purpose of implementing digital signatures in this system is to verify that the data payload sent is original and does not change during the transmission process, as well as guaranteeing the confidentiality of the data payload. The addition of security mechanisms on the LoRaWAN network such as encryption, decryption and verification results has resulted in overhead in several aspects. After the analysis, there was an increase in size for several aspects when the digital signature method was applied.

Keywords: LoRaWAN, sniffing, digital signature, AES, overhead.

1. Pendahuluan

Latar Belakang

Internet of Things (IoT) merupakan komunikasi yang menggunakan pemrograman untuk menghasilkan interaksi antar mesin tanpa campur tangan manusia [1]. Jaringan IoT dapat diklasifikasikan yaitu pada lapisan radio fisiknya, laju bit yang dapat dicapai dan konsumsi daya atau rentang komunikasi. Jaringan yang beroperasi jarak jauh, menggunakan daya rendah, serta dapat mentoleransi bit rate rendah akan cenderung menggunakan perangkat jaringan seperti LoRa [2].

Perangkat LoRaWAN merupakan infrastruktur jaringan dari teknologi modulasi radio *Long Range* (LoRa) dengan kelemahan tidak adanya proses enkripsi pada *payload data*. Kelemahan tersebut terjadi karena adanya *sniffing process* saat perangkat saling mengirim pesan [3]. *Sniffing* merupakan proses pengendusan paket data pada sistem jaringan, yang diantaranya dapat memonitor dan menangkap semua lalu lintas jaringan yang lewat

tanpa peduli kepada siapa paket itu dikirimkan, sehingga ketika proses *sniffing* tersebut berjalan kemungkinan besar akan muncul serangan pada perangkat LoRaWAN saat menerima data dari perangkat sebelumnya [4].

Berdasarkan permasalahan di atas cara yang digunakan untuk mengantisipasi agar tidak terjadi serangan yang disebabkan oleh proses kerja *sniffing* adalah dengan melakukan pencegahan enkripsi menggunakan *digital signature*. *Digital signature* bertujuan untuk otentikasi data *payload* dan untuk memastikan bahwa informasi tidak diperbarui selama transmisi [5]. *Digital signature* ini memastikan isi pesan tidak mengalami perubahan sampai ke tangan penerima, dengan demikian penerima yakin bahwa pesan yang diterimanya benar-benar asli dari pihak pengirim [6]. Penerapan metode *digital signature* dalam penelitian ini mengimplementasikan algoritma *Advanced Encryption Standard* (AES) pada proses enkripsi, dekripsi pesan dan menggunakan algoritma Ed25519 pada proses *signature* [7]. Algoritma AES digunakan karena ringan dan efisien dalam perangkat lunak, serta perangkat keras dan juga dapat diterapkan pada metode *digital signature* [8]. Algoritma Ed25519 dipilih karena menerapkan algoritma Curve25519, di mana dari hasil penelitian sebelumnya pada sub bab *Preliminaries*, penggunaan kedua algoritma tersebut lebih efisien dan juga mudah untuk diterapkan pada metode *digital signature*.

Penambahan mekanisme keamanan pada jaringan tentu akan menghasilkan *overhead* komputasi pada sistem untuk mengukur berapa banyak sumber daya tambahan yang dibutuhkan dalam mekanisme keamanan yang diterapkan [9]. Hasil analisis *overhead* yang dilakukan pada sistem yang telah dibuat untuk setiap *sender* dan *receiver* adalah dapat melihat *payload length, memory usage,* RAM *usage* dan *response processing time* [10].

Penelitian ini dilakukan untuk menganalisis penggunaan *digital signature* yang disebabkan oleh proses *sniffing* yang terjadi ketika perangkat saling mengirim pesan dan juga untuk menganalisis *overhead* yang terjadi akibat adanya penambahan mekanisme keamanan, sehingga dapat diketahui berapa banyak *overhead* yang dihasilkan dari beberapa aspek saat diterapkannya metode *digital signature* pada sistem.

Topik dan Batasannya

Berdasaran latar belakang yang telah dijelaskan, permasalahan yang diteliti adalah untuk menganalisis penggunaan *digital signature* yang disebabkan oleh proses *sniffing* yang terjadi saat perangkat saling mengirim pesan dan juga untuk menganalisis *overhead* yang terjadi akibat adanya penambahan mekanisme keamanan. Metode *digital signature* pada proses enkripsi dan dekripsi pesan menerapkan algortima AES 128, AES 256 dan pada proses *signature* menggunakan algoritma Ed25519. *Overhead* pada perangkat node dilakukan untuk mengukur ukuran *payload length*, *memory usage* dari *device* yang menjalankan program, penggunaan RAM dan *response processing time* saat node 2 *receiver* berhasil menerima pesan dari node 1 *sender* dan sebaliknya. Penelitian ini menggunakan 2 buah perangkat 868MHz LoRa Shield Module dan 2 buah perangkat Arduino Mega 2560 sebagai node 1 *sender* dan node 2 *receiver*, serta 1 buah perangkat Raspberry Pi untuk LoRaWAN sebagai *sniffing*. Bahasa pemrograman yang digunakan adalah C++ dengan tipe data yang dikirimkan adalah tipe data *string*. Pengujian dilakukan dengan *testing man-in-the-middle* untuk *sniffing* sebelum diterapkannya proses *encrypt signature* dan setelah diterapkannya proses *encrypt signature*, sehingga dapat meyakinkan bahwa program dapat mengamankan *payload data* yang dikirim.

Tujuan

Tugas akhir ini bertujuan untuk menganalisis penggunaan *digital signature* yang disebabkan oleh proses *sniffing*. Algoritma enkripsi yang digunakan saat perangkat node saling berkomunikasi adalah AES 128, AES 256 dan algoritma *signature* Ed25519. Pemilihan algoritma ini digunakan karena dapat mencegah penyalahgunaan *payload data* yang dikirim dan juga dapat mengetahui nilai *overhead* yang dihasilkan dari penerapan metode *digital signature* pada sistem.

Organisasi Tulisan

Penulisan tugas akhir ini disusun dalam 5 Bab bagian, yaitu Bab 1 – Pendahuluan yang berisikan latar belakang, topik dan batasannya, serta tujuan. Bab 2 – Studi terkait menjelaskan tentang literatur apa saja yang telah diteliti sebelumnya sebagai acuan untuk penelitian ini. Bab 3 – Perancangan Sistem yaitu menjelaskan alur sistem yang dibuat. Bab 4 – Implementasi dan Evaluasi menjelaskan tentang pengimplementasian sistem dan analisis dari program yang telah dibuat. Bab 5 – Penutup yang berisi kesimpulan dari hasil analisis dan saran.

2. Studi Terkait

Penelitian mengenai penggunaan metode *digital signature* dalam mencegah terjadinya serangan pada perangkat jaringan LoRaWAN yang disebabkan oleh proses *sniffing* masih belum banyak dilakukan, pada penelitian yang telah dilakukan dari berbagai *reference* kebanyakan masih menggunakan metode selain *digital signature*.

Jelasnya berdasarkan *paper* [11] pengembangan sistem perantara pengiriman data menggunakan modul komunikasi LoRa dan protokol MQTT pada *Wireless Sensor Network*. Fokus *paper* adalah menambahkan sistem pengiriman data atau *gateway* pada komunikasi antar perangkat LoRa. Permasalahannya yaitu tidak adanya perantara *gateway* yang berperan sebagai media komunikasi antara node sensor dengan *server*. Hasilnya yaitu penambahan sistem *gateway* dapat menghubungkan komunikasi antara node sensor dengan *server* menggunakan modul komunikasi LoRa dan protokol MQTT, sehingga kelebihan dari hasil yang didapatkan adalah berdasarkan karakteristik IoT sistem dapat mendukung penggunaan *bandwidth* yang kecil pada perangkat LoRa yang juga menerapkan penggunaan *gateway* sebagai komunikasi antar node dan *server*, serta kekurangannya adalah jarak antara ketiga perangkat dalam berkomunikasi harus lebih dari 400 meter agar perangkat *gateway* dapat berfungsi dengan baik.

Paper [2] *selective jamming of LoRaWAN using commodity hardware*. Fokus *paper* adalah melakukan penelitian serangan *selective jamming* yang terjadi pada jaringan LoRaWAN dengan menerapkan proses *sniffer* dan *jammer* di dalamnya. Permasalahan yang terjadi adalah keamanan dalam LoRa dan LoRaWAN yang muncul karena pilihan jenis modulasi yang kuat tetapi lambat dalam protokol. Sehingga hasil yang didapatkan adalah menyempurnakan serangan pada LoRa *traffic* yang mengeksploitasi keterampilan pesan LoRa terhadap interferensi yang disebabkan oleh pesan-pesan LoRa saat disinkronkan dengan kekuatan sinyal yang lebih tinggi. Kelebihannya yaitu dapat membuat serangan *selective jamming* untuk dapat dilakukan analisis, serta teknik-teknik investigasi serangan yang muncul dan kekurangan dalam paper tersebut yaitu tidak terdapat metode penyelesaian untuk mengatasi serangan yang terjadi karena hanya berfokus pada pembuatan serangan untuk di analisis saja.

Pada *paper* selanjutnya [4] dijelaskan tentang monitoring jaringan *wireless* terhadap serangan *packet sniffing* dengan menggunakan IDS. Fokus *paper* adalah pengawasan terhadap *traffic* jaringan dan pengawasan terhadap kegiatan-kegiatan yang mencurigakan didalam sebuah sistem jaringan. Permasalahan yang terjadi yaitu *access point* sangat rentan terhadap berbagai ancaman serangan, salah satu contoh serangan adalah dengan menggunakan *packet sniffing* karena komunikasi yang terjadi bersifat terbuka. Hasilnya adalah penggunaan protokol HTTP untuk melakukan aktivitas internet tidak akan menyebabkan terjadinya gangguan dalam jaringan *access point*, sehingga kelebihan dari *paper* tersebut yaitu *tools ettercap* dapat merekam dengan baik ketika *user* melakukan aktivitas internet menggunaka protokol HTTP, serta untuk kekurangannya adalah perlu dilakukan penambahan tindakan pada saat terjadinya serangan dari jaringan internet.

Paper [12] *security risk analysis of LoRaWAN and future directions*. Fokus *paper* adalah analisis terhadap ancaman LoRaWAN seperti penangkapan fisik perangkat akhir, *gateway* yang didalamnya terdapat proses *sniffing* dan *self-replay* yang memerlukan perhatian khusus oleh pengembang dan organisasi yang menerapkan jaringan LoRa. Permasalahan yang terjadi yaitu risiko keamanan komprehensif dari protokol dan bagaimana mencari solusi untuk risiko keamanan tersebut. Sehingga hasil dan kelebihan yang didapatkan adalah terciptanya katalog ancaman untuk LoRaWAN dengan melakukan diskusi dan analisis dalam pandangan skala, dampak dan kemungkinan dari setiap ancaman, serta untuk kekurangannya kemungkinan akan berdampak pada beberapa ancaman keamanan perangkat jaringan yang relavan.

Paper [7] *a comparative survey of symmetric and asymmetric key cryptography*. Fokus *paper* adalah pengamanan pengiriman data tanpa kehilangan kerahasiaan dan integritas dengan menggunakan klasifikasi *cryptography types Symmetric Key Cryptography and Asymmetric Key.* Permasalahannya yaitu kurangnya pengetahuan mengenai teknik *cryptography* sebagai penyedia sementara transmisi data pada keamanan jaringan. Hasilnya adalah *symmetric key cryptography* menggunakan kunci yang sama pada proses enkripsi dan dekripsi dari pihak pengirim maupun penerima pesan, dan algoritma yang dikembangkan untuk menggambarkan kunci simetris kriptografi adalah AES, DES, 3DES dan *Blowfish*. Sedangkan untuk *asymmetric key cryptography* pada proses enkripsi, dekripsi pesan pengirim dan penerima menggunakan kunci yang berbeda, serta algoritma yang biasa digunakan untuk mengimplementasikan *asymmetric key cryptography* ini adalah RSA, *Diffie-Hellman* dan ECC.

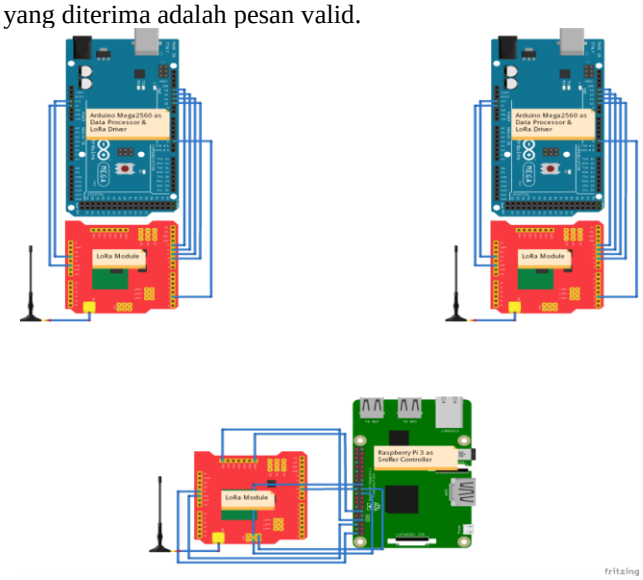
Paper [13] *Onboarding and Software Update Architecture for IoT*. Fokus *paper* adalah Ed25519 merupakan turunan dari tanda tangan skema EdDSA, algoritma Ed25519 menerapkan *symmetric key* dengan menggunakan SHA-512, anggota keluarga SHA-2 pada proses *hashing*. Hasilnya adalah EdDSA memberikan resistensi serangan yang setara dengan 128 bit *symmetric ciphers*, dengan menggunakan *public key* ukuran 16 bytes dan *signature key* sebanyak 64 bytes untuk algoritma Ed25519.

Sistem ini menggunakan *paper* [11] sebagai referensi dan acuan untuk penggunaan IoT yang dapat mendukung penggunaan *bandwidth* yang kecil pada perangkat jaringan LoRa, lalu *paper* [2] [4] dan [12] sebagai referensi dalam mengetahui ancaman atau serangan yang terjadi ketika proses *sniffing* berjalan. Dalam penerapan sistem penggunaan metode *digital signature* umumnya digunakan untuk otentikasi *payload data* [14]. Pada algoritma AES dan Ed25519 yang diterapkan pada sistem menggunakan *paper* [7] dan [13] sebagai referensi untuk menerapkan konsep komparatif *symmetric key cryptography*. Skema *digital signature* yang diterapkan dalam penelitian ini adalah *Encrypt then Sign* karena ketika terjadi eksploitasi pesan dari pihak yang bertindak sebagai *sniffing*, maka pihak pengirim dan penerima pesan dapat mengetahui siapakah yang

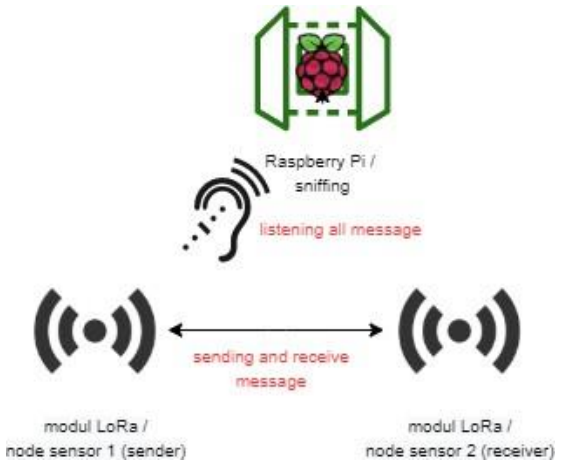
melakukan eksploitasi pesan tersebut, hal ini karena *signature key* yang didapatkan bukan lagi milik pihak pengirim pesan tetapi milik pihak *sniffing*, sehingga dengan menerapkan skema *Encrypt then Sign* maka kemungkinan ekspoloitasi pesan sangat kecil untuk terjadi. Sedangkan jika menerapkan skema *Sign then Encrypt* ketika pihak yang bertindak sebagai *sniffing* melakukan eksploitasi pesan dari pihak pengirim dan mengirim pesan tersebut kepada pihak penerima pesan, pesan yang diterima tersebut masih mencantumkan *signature key* milik pihak pengirim pesan, hal tersebut karena pihak *snifiing* hanya melakukan perubahan pesan dari proses dekripsi pada *plaintext* saja tidak pada *signature key* milik pengirim pesan, sehingga pihak penerima dan pengirim pesan tidak dapat mengetahui jejak pihak *sniffing* yang telah mengeksploitasi pesan tersebut [15]. Pada penelitian ini juga dilakukan analisis *overhead* untuk setiap *sender* dan *receiver* dengan mengukur beberapa aspek seperti mengukur ukuran *payload length*, *memory usage*, RAM *usage*, serta *response processing time* saat sebelum dan sesudah diterapkannya metode *digital signature* [10].

3. Sistem yang Dibangun

Penelitian sistem pengamanan perangkat jaringan LoRaWAN dengan metode *digital signature* menggunakan algoritma AES 128, AES 256 untuk proses enkripsi dan dekripsi pesan, serta algoritma Ed25519 untuk proses *signature*. Gambar 1 merupakan *sketch* dari perangkat yang digunakan, serta Gambar 2 merupakan arsitektur sistem secara keseluruhan, di mana perangkat node 2 sebagai *receiver* hanya akan bereaksi jika data yang diterima berisi ID yang sama dengan yang dimiliki oleh node 1 sebagai *sender*, ketika node 2 berhasil menerima pesan yang dikirim maka selanjutnya perangkat node 2 sebagai *receiver* akan mengirimkan konfirmasi kepada node 1 bahwa pesan yang diterima adalah pesan valid.



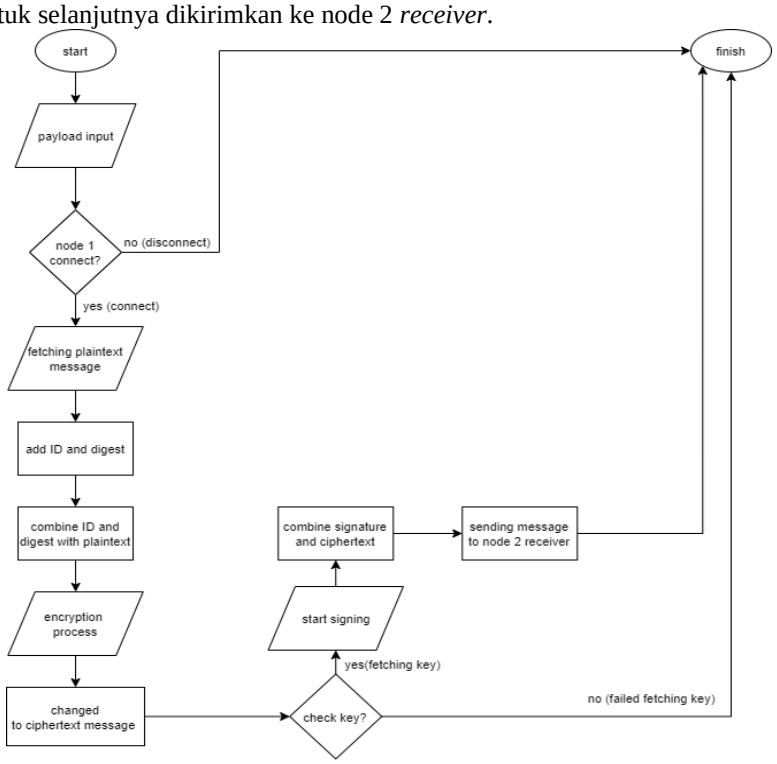
Gambar 1. Sketch Device



Gambar 2. Arsitektur Sistem

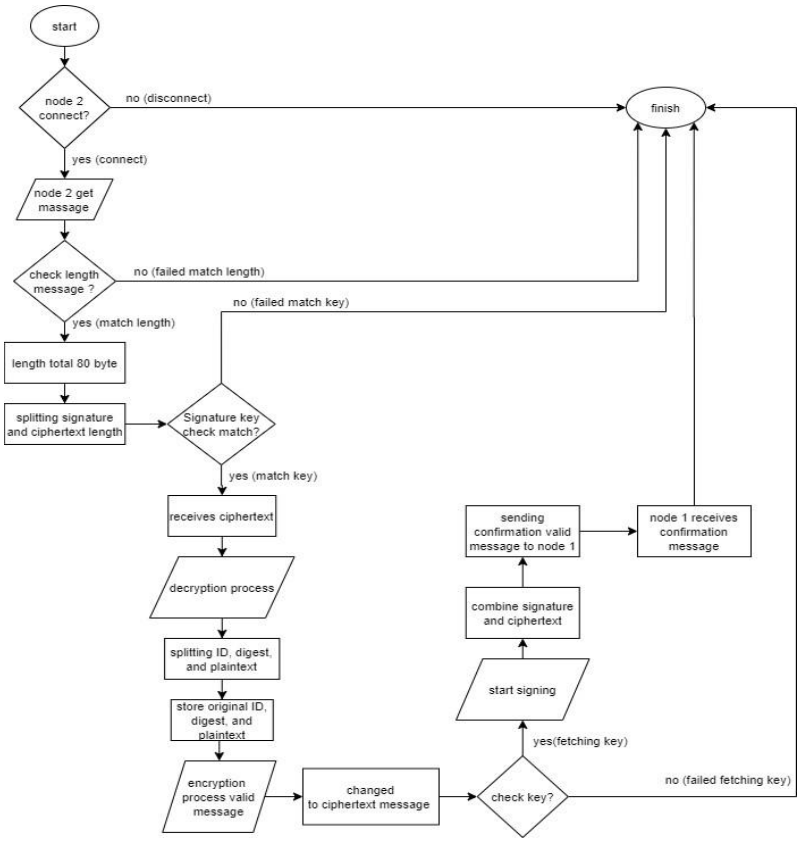
Skenario pada proses sistem menggunakan metode *end-to-end* yang terdiri dari skenario diagram alur kerja node sensor 1, serta diagram alur kerja node sensor 2. Gambar 3 merupakan diagram alur kerja node sensor 1.

Proses pertama yang dilakukan adalah perangkat node 1 sebagai *sender* harus terkoneksi terlebih dahulu dengan node 2, jika koneksi berhasil maka *plaintext massage* akan diambil untuk selanjutnya ditambahkan ID dan *digest*, kemudian digabungkan dengan *plaintext message.* Selanjutnya dilakukan proses enkripsi sehingga pesan akan berubah menjadi *ciphertext message*, setelah itu dilakukan pengecekan *key* yang akan digunakan untuk proses penandatanganan atau *signing*, jika proses *signing* berhasil maka *signature key* akan digabungkan dengan *chipertext message* untuk selanjutnya dikirimkan ke node 2 *receiver*.



Gambar 3. Diagram Alur Sender

Gambar 4 adalah diagram alur kerja node sensor 2 sebagai *receiver*. Proses pertama adalah perangkat node 2 harus terkoneksi dengan node sensor 1 sebagai perangkat pengirim pesan, jika berhasil terkoneksi maka node 2 akan mengambil pesan tersebut. Selanjutnya dilakukan proses pencocokan *length total* 80 bytes, di mana 64 bytes adalah *length signature key* dan 16 bytes adalah *length ciphertext message*, jika jumlah *length message* sama yaitu 80 bytes maka akan dilakukan proses *splitting length signature key and ciphertext message.* Kemudian proses dekripsi diperlukan untuk mengubah pesan berupa *ciphertext* yang diterima. Node sensor 2 tidak akan menampilkan pesan yang dikirimkan selama pesan tersebut belum terdekripsi, untuk itu sebelum dilakukan proses dekripsi node 2 akan mencocokan *signature key* dari node 1, setelah berhasil menemukan *key* yang sama dengan *key* yang digunakan oleh node 1 pada proses *signing* maka dekripsi akan dilakukan pada proses *'decryption'*. Hasil yang didapatkan dari proses dekripsi adalah ID, *digest* dan *plaintext*, oleh karena itu maka diperlukannya pemisahan antara ketiga hasil dekripsi tersebut, jika proses *splitting* telah dilakukan maka selanjutnya ketiga kategori hasil tersebut akan disimpan. Setelah disimpan selanjutnya akan dilakukan proses enkripsi konfirmasi pesan valid, sehingga pesan berubah menjadi *ciphertext* kembali, kemudian node 2 juga akan melakukan proses pengecekan *key* yang akan digunakan untuk proses penandatanganan atau *signing*, jika proses *signing* berhasil maka *signature key* akan digabungkan dengan *chipertext*. Selanjutnya konfirmasi pesan valid yang telah di proses pada node 2 akan dikirim ke node 1 *sender*.



Gambar 4. Diagram Alur Receiver

4. Evaluasi

4.1 Hasil Pengujian

Skenario pengujian dilakukan dengan *testing man-in-the-middle* untuk proses *sniffing*, dengan menggunakan 3 mode yaitu mode 0 *"MODE_NON_SIGNATURE",* mode 1 *"MODE_SIGNATURE_AES128"* dan mode 2 *"MODE_SIGNATURE_AES256".* Percobaan ini dimulai dengan mode 0 terlebih dahulu, kemudian mode 1 dan mode 2, dengan cara kerja sistem berdasarkan diagram alur sistem yang telah dibuat.

Gambar 5 pada halaman lampiran merupakan inisialisasi *sender* ID, serta *private key*, *public key*, *encrypt key* AES 128, AES 256 dan *digest* yang masing-masing dimiliki oleh node 1 dan node 2. Gambar 6 pada halaman lampiran adalah hasil pengujian proses *sniffing* dengan menggunakan mode 0 *"MODE_NON_SIGNATURE"*, mode 1 *"MODE_SIGNATURE_AES128"* ditunjukkan pada halaman lampiran Gambar 7 dan mode 2 *"MODE_SIGNATURE_AES256"* ditunjukkan pada halaman lampiran Gambar 8.

Setelah melakukan pengujian *man-in-the-middle* untuk proses *sniffing*, pengujian selanjutnya adalah dengan melakukan tes *overhead* pada sistem yang dibuat. Untuk menghasilkan nilai *overhead* maka digunakan 3 mode yang sama seperti pengujian proses *sniffing*. Pengujian *overhead* dilakukan dengan mengirimkan data *string* oleh node 1 *sender* ke node 2 *receiver*. Analisis *overhead* yang dilakukan adalah analisis *payload length*, *memory usage*, RAM *usage* dan *response processing time*. Tabel 2 pada halaman lampiran menunjukkan hasil perbandingan *overhead* dari ketiga mode yang digunakan. Hasil pengujian *overhead* yang dilakukan dapat dilihat pada halaman lampiran. Kode program dapat dilihat pada *repository* git berikut : https://github.com/rahayuindahlestari/1301188560.

4.2 Analisis Hasil Pengujian

Berdasarkan hasil pengujian, program dapat berjalan dengan sebagaimana mestinya sesuai dengan diagram *flow* sistem yang dibuat. Seperti yang tertera pada Gambar 6 pengujian dengan menggunakan mode 0, perangkat yang berperan sebagai *sniffing* mengetahui semua *payload data* asli dari kedua perangkat node yang saling berkomunikasi, tidak adanya proses enkripsi pada *payload data* tersebut memungkinkan terjadinya serangan dan penyalahgunaan pesan sehingga dapat menghambat komunikasi antar perangkat LoRaWAN. Sedangkan untuk pengujian mode 1 pada Gambar 7 dan mode 2 pada Gambar 8, dapat dilihat pada serial monitor bahwa perangkat yang berperan sebagai *sniffing* juga mengetahui semua komunikasi antar perangkat, hanya saja *payload data* yang didapatkan telah di enkripsi dan ditandatangani, sehingga keaslian *payload data* dapat terjaga dengan baik

dan aman sampai kepada penerima pesan. Berdasarkan analisa tersebut dapat disimpulkan bahwa *payload data* yang sebelumnya tidak terenkripsi telah berhasil terenkripsi dengan baik setelah diterapkannya metode keamanan *digital signature*.

Pada hasil pengujian proses *sniffing* yang telah dilakukan sebelumnya, maka dapat diketahui nilai *overhead* saat sebelum dan sesudah penerapan metode *digital signature* yang terdiri dari beberapa kategori, seperti yang dilampirkan pada Tabel 2 untuk setiap sisi *sender* maupun *receiver*. Analisis *overhead* yang pertama adalah analisis ukuran panjang *payload data*, dari hasil pengujian pada Tabel 2 hasil pengujian *payload length* dapat dijabarkan lebih spesifik ke dalam Tabel 1 seperti di bawah ini, dapat dilihat bahwa pada mode 1 dan mode 2 yang digunakan dalam pengujian mempunyai *header* data tambahan. Pada mode 0 ukuran *payload length* yang digunakan sebesar 16 bytes saja, dengan 4 bytes adalah data tambahan yang terdiri dari 2 bytes ID *sender* dan 2 bytes *digest*, serta 12 bytes *real data*. Sedangkan pada mode 1 dan mode 2 *payload length* yang digunakan sebesar 80 bytes, dengan 64 bytes adalah *header* data tambahan yaitu *signature key* dan 2 bytes ID *sender*, 2 bytes *digest* serta 12 bytes *real data*. Panjang pesan pada mode 0 menyesuaikan dengan *encrypt key* yang digunakan pada algoritma AES dimana total panjang kunci adalah 32 bit, oleh karena itu untuk setiap algoritma AES tersebut dibagi dengan 8bit, sehingga untuk 128 bit = 16 bytes dan untuk 256 bit = 32 bytes, kemudian berdasarkan *library* algoritma AES yang digunakan, untuk *plaintext* yang terenkripsi adalah hanya 16 bytes saja baik ketika menggunakan AES 128 atau AES 256 [16]. Pada mode 1 dan mode 2 terlihat jelas bahwa adanya penambahan *header* akan menyebabkan *payload length* yang dihasilkan 4 kali lebih banyak dibandingkan dengan mode 0 yang hanya menggunakan 12 bytes *real data* dan 4 bytes data tambahan, oleh karena itu dari sisi *overhead payload length* yang dihasilkan memang lebih besar, hanya saja dari sisi keamanan sistem akan lebih baik. Untuk penggunaan mode 1 dan mode 2 hasil *payload length* yang didapatkan memang sama, tetapi dari sisi penggunaannya mode 1 lebih efisien digunakan daripada mode 2, karena untuk sistem yang telah dibuat penggunaan *plaintext* 16 bytes sudah cukup terenkripsi dengan menggunakan algoritma AES 128 dibandingkan dengan algoritma AES 256 karena menyisahkan lebih banyak *blok size* yang tidak terpakai.

Tabel 1. Hasil Pengujian Payload Length

Analisis Pengujian	Header data tambahan			Real Data	Payload Length
	ID sender	digest	signature key		
Mode 0	2	2	0	12	16 bytes
Mode 1	2	2	64	12	80 bytes
Mode 2	2	2	64	12	80 bytes

Analisis *overhead* yang kedua adalah *memory usage*, pengecekkan dilakukan dengan melihat *sketch* saat program dijalankan. Hasil pada Tabel 2 menunjukkan bahwa mode 1 dan mode 2 menggunakan lebih banyak *memory* dibandingkan dengan mode 0, hal ini terjadi karena saat metode *digital signature* diterapkan sistem akan menambahkan lebih banyak fungsi pada program sehingga penggunaan *memory* juga lebih besar dibandingkan sebelum penerapan metode *digital signature* pada mode 0. Untuk analisis *overhead* yang ketiga yaitu RAM *usage*, pengecekkan dilakukan dengan melihat *global variables use* ketika program dijalankan. Sama seperti *memory usage* penggunaan RAM pada mode 1 dan mode 2 juga lebih banyak dibandingkan dengan mode 0, penyebab terjadinya pun sama seperti *memory usage* karena pada mode 1 dan mode 2 sistem telah diterapkan metode *digital signature*. Oleh sebab itu, *overhead memory and RAM usage* pada sistem terjadi karena semakin banyak fungsi yang diterapkan pada program maka semakin banyak pula konsumsi *memory* dan RAM yang dibutuhkan, akan tetapi sistem menjadi lebih aman dalam penggunaanya.

Analisis *overhead* yang keempat adalah *Response processing time sender to receiver*. Pada Tabel 2 menunjukkan bahwa mode 1 dan mode 2 menggunakan waktu respon yang lebih lama saat *sender* mengirim *payload data* kepada *receiver*, hal ini terjadi karena sebelum pesan dikirim dilakukan proses enkripsi *payload data* dan *signing* terlebih dahulu. Setelah pesan dari *sender* berhasil diterima, *receiver* akan melakukan proses validasi *signature key* dan memberikan respon validasi *key* kepada *sender*. Pada analisis *overhead* yang kelima *Response processing time receiver to sender*, waktu respon yang digunakan pada mode 1 dan mode 2 juga lebih lama dibandingkan dengan mode 0, karena setelah *payload data* diterima node perangkat *receiver* akan mengirim konfirmasi kepada node perangkat *sender* bahwa *payload data* yang diterima adalah pesan valid, tetapi sebelum proses pengiriman konfirmasi tersebut dilakukan, *payload data* harus diubah ke dalam bentuk *encrypt* dan dilakukan proses *signing* terlebih dahulu. Oleh karena itu dari kedua hasil analisis pengujian *response processing time* pada perangkat LoRaWAN, setelah diterapkan metode *digital signature* pembatasan waktu transmisi mengalami *overhead*, karena banyaknya proses yang digunakan dan berdasarkan penelitian yang telah dilakukan sebelumnya seharusnya jika siklus kerja diterapkan 1% maka sebuah node diperbolehkan mengirim hanya selama 36 detik/jam atau sekitar 360 ms [17], sedangkan pada penelitian ini setelah sistem keamanan diterapkan dengan siklus kerja yaitu 14% dari efek penggunaan algoritma *encryption* dan *signature*, maka hasil maksimal *response processing time* adalah 16063.53 ms, tentu saja sistem mengalami *overhead* dari batas waktu yang telah ditentukan, hanya saja dari sisi keamanan penggunaan sistem akan lebih terjaga dengan baik.

5. Kesimpulan

Berdasarkan pengujian proses *sniffing* dan *overhead* sistem yang telah dilakukan dapat disimpulkan bahwa, penggunaan sistem dengan menerapkan metode *digital signature* dapat berjalan sesuai alur diagram dan tujuan penelitian. Penelitian ini menggunakan algoritma enkripsi AES 128, AES 256 dan algoritma *signature* Ed25519. Hasil pengujian proses *sniffing* pada mode 1 dan mode 2, yaitu proses *sniffing* mengetahui semua komunikasi antar kedua perangkat node melalui serial monitor perangkat *sniffing*, tetapi *payload data* yang diketahui hanya dalam bentuk *ciphertext* atau pesan yang sudah terenkripsi, serta dalam bentuk byte sehingga keaslian dan keamanan pesan dapat terjaga dengan baik. Tujuan analisis *overhead* dalam penelitian ini adalah untuk mengukur panjang *payload*, penggunaan *memory* dan RAM, serta respon waktu proses antar kedua perangkat. *Overhead* dilakukan dengan melihat hasil pada pengujian proses *sniffing* dengan menggunakan 3 mode. Hasil analisis *overhead* untuk kelima kategori dalam penelitian ini adalah, setelah penambahan mekanisme keamanan pada kedua perangkat node menghasilkan nilai *overhead* yang tinggi dibandingkan sebelum diterapkannya metode keamanan *digital siganture*. Dari hasil pengujian juga diketahui bahwa sistem ini lebih efisien diterapkan pada algoritma enkripsi AES 128 dibandingkan dengan algoritma AES 256 karena menyisahkan lebih banyak *blok size* yang tidak terpakai.

Daftar Pustaka

[1] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of mqtt communication protocol in iot system," in *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2017.

[2] E. Aras, N. Small, G. S. Ramachandran, S. Delbruel, W. Joosen, and D. Hughes, "Selective jamming of LoRaWAN using commodity hardware," in *ACM International Conference Proceeding Series*, 2017.

[3] D. V. Sandi and M. Arrofiq, "Implementasi Analisis NIDS Berbasis Snort Dengan Metode Fuzy Untuk Mengatasi Serangan LoRaWAN," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, 2018.

[4] A. Fauzi and I. Suartana, "MONITORING JARINGAN WIRELESS TERHADAP SERANGAN PACKET SNIFFING DENGAN MENGGUNAKAN IDS," *J. Manaj. Inform.*, vol. 8, no. 2, 2018.

[5] R. Dhagat and P. Joshi, "New approach of user authentication using digital signature," in *2016 Symposium on Colossal Data Analysis and Networking, CDAN 2016*, 2016.

[6] M. Ihwani, "MODEL KEAMANAN INFORMASI BERBASIS DIGITAL SIGNATURE DENGAN ALGORITMA RSA," *CESSJournal Comput. Eng. Syst. Sci.*, 2016.

[7] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography," in *2014 International Conference on Electronics, Communication and Computational Engineering, ICECCE 2014*, 2014.

[8] M. A. Mughal, X. Luo, A. Ullah, S. Ullah, and Z. Mahmood, "A lightweight digital signature based security scheme for human-centered internet of things," *IEEE Access*, 2018.

[9] A. Rizzardi, S. Sicari, D. Miorandi, and A. Coen-Porisini, "AUPS: An Open Source AUthenticated Publish/Subscribe system for the Internet of Things," *Inf. Syst.*, 2016.

[10] H. Hidayat, P. Sukarno, and A. A. Wardana, "Overhead Analysis on the Use of Digital Signature in MQTT Protocol," in *Proceedings of the International Conference on Electrical Engineering and Informatics*, 2019.

[11] H. Arijuddin, A. Bhawiyuga, and K. Amron, "Pengembangan Sistem Perantara Pengiriman Data Menggunakan Modul Komunikasi LoRa dan Protokol MQTT Pada Wireless Sensor Network," *Pengemb. Teknol. Inf. dan Ilmu Komput.*, 2019.

[12] I. Butun, N. Pereira, and M. Gidlund, "Security risk analysis of LoRaWAN and future directions," *Futur. Internet*, 2018.

[13] H. Gupta and P. C. Van Oorschot, "Onboarding and Software Update Architecture for IoT Devices," in *2019 17th International Conference on Privacy, Security and Trust, PST 2019 - Proceedings*, 2019.

[14] Z. A. Alizai, N. F. Tareen, and I. Jadoon, "Improved IoT Device Authentication Scheme Using Device Capability and Digital Signatures," in *ICAEM 2018 - 2018 International Conference on Applied and Engineering Mathematics, Proceedings*, 2018.

[15] A. Joux, *Algorithmic Cryptanalysis*. 2009.

[16] Asiyanik, "STUDI TERHADAP ADVANCED ENCRYPTION STANDARD (AES) DANALGORITMA KNAPSACK DALAM PENGAMANAN DATA," *SANTIKA*, 2017.

[17] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things," *Comput. Commun.*, 2020.

Lampiran

```
LoRa As SENDER OK: 868.10
[Private Key]
9d 61 b1 9d ef fd 5a 60 ba 84 4a f4 92 ec 2c c4 44 49 c5 69 7b 32 69 19 70 3b ac 03 1c ae 7f 60
[Public Key]
d7 5a 98 01 82 b1 0a b7 d5 4b fe d3 c9 64 07 3a 0e e1 72 f3 da a6 23 25 af 02 1a 68 f7 07 51 1a
[Encrypt Key 128]
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
[Encrypt Key 256]
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
[Digest/Salt]
ab ac
[Sender ID]
30 31
Type any text (max: 12 chars):
--------------------------------
```

```
LoRa As RECEIVER OK: 868.10
[Private Key]
44 49 c5 69 7b 32 69 19 70 3b ac 03 1c ac 7f b4 9d 61 b1 9d ef fd 5a 60 ba 84 4a f4 92 ec 2c 70
[Public Key]
d7 5a 98 01 82 b1 0a b7 d5 4b fe d3 c9 64 07 3a 0e e1 72 f3 da a6 23 25 af 02 1a 68 f7 07 51 1a
[Encrypt Key 128]
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
[Encrypt Key 256]
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f
[Digest/Salt]
ba bc
--------------------------------
```

Gambar 5. Inisialisasi Node 1 dan Node 2

Gambar 6. Mode 0 : Node 1 Sender (a), Node 2 Receiver (b), Sniffing (c)

Gambar 7. Mode 1 : Node 1 Sender (a), Node 2 Receiver (b), Sniffing (c)

Gambar 8. Mode 2 : Node 1 Sender (a), Node 2 Receiver (b), Sniffing (c)

Tabel 2. Hasil Pengujian Overhead

No	Mode 0		Mode 1		Mode 2	
	Sender	Receiver	Sender	Receiver	Sender	Receiver
1. Payload length	16 bytes	16 bytes	80 bytes	80 bytes	80 bytes	80 bytes
2. Memory usage	13506 bytes (5%)	12304 bytes (4%)	36770 bytes (14%)	35680 bytes (14%)	37080 bytes (14%)	35998 bytes (14%)
3. RAM usage	1754 bytes (21%)	1374 bytes (16%)	2564 bytes (31%)	2091 bytes (25%)	2644 bytes (32%)	2171 bytes (26%)
4. Response processing time sender to receiver	174.64 ms	34.68 ms	16063.53 ms	9884.53 ms	15763.17 ms	9584.58 ms
5. Response processing time receiver to sender	42.14 ms	169.07 ms	9683.87 ms	6187.79 ms	10004.44 ms	6187.77 ms

Gambar 9. Hasil Pengujian Overhead Mode 0

Payload Length

```
[Plain Text in Char]
testing1234
[Plain Text]
74 65 73 74 69 6e 67 31 32 33 34 00
[ID + Plain Text + Digest]
30 31 74 65 73 74 69 6e 67 31 32 33 34 00 ab ac
[Encrypted Message]
eb db 22 96 7f 99 50 17 34 bf e9 0b 08 71 7b c2
[Signing -> OK]
[Signature]
1e d8 cd ef 78 e6 1c dc 20 ef b6 d0 17 32 97 db 2a de 33 c1 28 52 60 62 b9 39 29 e3 a6 a2 5b 27
43 97 5d aa 73 4c 59 2e 6e e9 d5 50 b7 bf 66 f4 4a d1 63 31 d5 3f 2e f4 0f 49 44 97 7a cb e3 0c
[Sending Payload -> Sent] -> 80
```

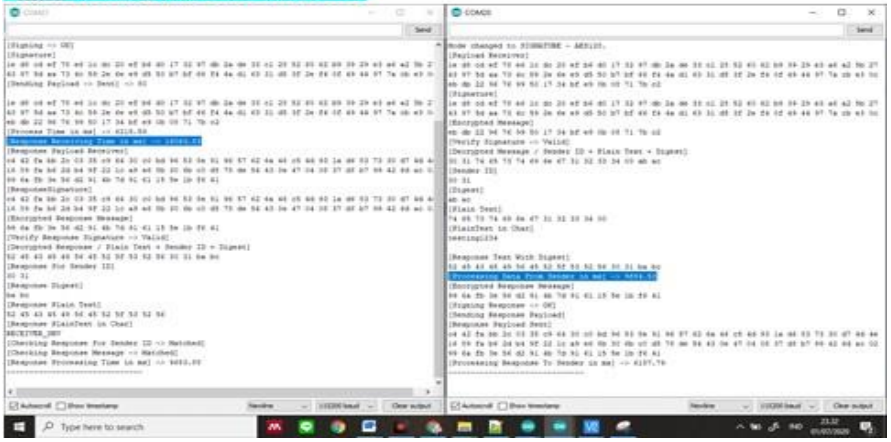
Memory and RAM Usage Sender

```
Sketch uses 34770 bytes (11%) of program storage space. Maximum is 253952 bytes.
Global variables use 2564 bytes (31%) of dynamic memory, leaving 5628 bytes for local variables. Maximum is 8192 bytes.
```

Memory and RAM Usage Receiver

```
Sketch uses 35680 bytes (14%) of program storage space. Maximum is 253952 bytes.
Global variables use 2091 bytes (25%) of dynamic memory, leaving 6101 bytes for local variables. Maximum is 8192 bytes.
```

Respon Processing Time Sender and Receiver



Gambar 10. Hasil Pengujian Overhead Mode 1

Payload Length

```
[Plain Text in Char]
testing1234
[Plain Text]
74 65 73 74 69 6e 67 31 32 33 34 00
[ID + Plain Text + Digest]
30 31 74 65 73 74 69 6e 67 31 32 33 34 00 ab ac
[Encrypted Message]
c3 56 d9 cd 2e 61 00 d2 60 11 f5 08 22 b5 2a 22
[Signing -> OK]
[Signature]
43 26 13 dd a2 e7 2a c9 dc cf 9c 6d f2 22 d7 4a 34 d3 12 7b 6f 92 57 ea ad 92 41 b8 30 3b 5e 7f
04 49 2d ef c8 ec c4 47 f0 98 32 2a cd 1b 81 e1 2e 00 26 24 9f 05 fe 2e 14 e3 f8 eb b1 0d e9 05
[Sending Payload -> Sent] -> 80
```

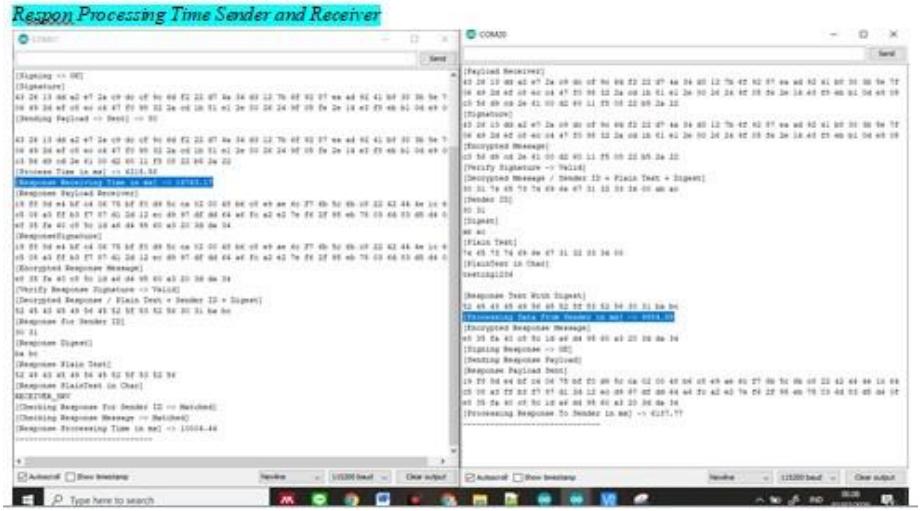
Memory and RAM Usage Sender

```
Sketch uses 37060 bytes (14%) of program storage space. Maximum is 253952 bytes.
Global variables use 2644 bytes (32%) of dynamic memory, leaving 5548 bytes for local variables. Maximum is 8192 bytes.
```

Memory and RAM Usage Receiver

```
Sketch uses 35998 bytes (14%) of program storage space. Maximum is 253952 bytes.
Global variables use 2171 bytes (26%) of dynamic memory, leaving 6021 bytes for local variables. Maximum is 8192 bytes.
```

Respon Processing Time Sender and Receiver



Gambar 11. Hasil Pengujian Overhead Mode 2