Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 1, No. 1, Januari 2024, hlm. x-x http://j-ptiik.ub.ac.id

Implementasi dan Analisis Performa Algoritma Enkripsi ChaCha20 Berbasis Protokol Komunikasi ESP-NOW Pada Wireless Sensor Network

Naufal Farras Trikusuma[1], Agung Setia Budi[2]

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]naufalfarr@student.ub.ac.id, [2]agungsetiabudi@ub.ac.id

Abstrak

Wireless Sensor Network (WSN) telah berkembang pesat untuk mendukung berbagai proses otomasi, seperti pengumpulan data sensor, pengendalian sistem nirkabel, dan *monitoring*. Namun, salah satu tantangan WSN adalah memastikan keamanan data yang sering kali sensitif dan rentan terhadap ancaman seperti *data breach*, tantangan selanjutnya adalah keterbatasan sumber daya perangkat, seperti daya komputasi, memori, dan energi. Untuk mengatasi hal ini, *lightweight cryptography* (LWC) diperlukan untuk memberikan keamanan namun tetap efisien. Penelitian ini menganalisis performa algoritma enkripsi ChaCha20 pada protokol komunikasi ESP-NOW menggunakan perangkat berbasis ESP8266 yang tidak dilengkapi *cryptographic hardware accelerators*. Pengujian dilakukan secara komparatif dengan algoritma AES-CBC, Clefia, dan Snow-V, berdasarkan kecepatan enkripsi-dekripsi, konsumsi daya, serta keamanan data. Hasil menunjukkan ChaCha20 memiliki keunggulan dalam waktu komputasi, menjadi algoritma tercepat untuk data 5 KB dan 10 KB, dengan konsumsi energi terendah. Dalam uji keamanan, tingkat *randomness* ciphertext ChaCha20 juga cukup baik, dengan nilai *Shannon entropy* 49.375 (61,72%) dan uji *Chi-Square* menunjukkan keseragaman distribusi byte (p-value 0.7415). Meski keamanan ChaCha20 sedikit lebih rendah dibandingkan AES-CBC dan Clefia, algoritma tetap cukup efektif menjaga kerahasiaan data. Tanpa bergantung pada tambahan perangkat keras khusus, efisiensi sumber daya dan kecepatan waktu komputasi ChaCha20 menjadi solusi ideal untuk perangkat WSN dengan karakteristik sumber daya terbatas.

Kata kunci: *Wireless Sensor Network, ChaCha20, Lightweight Cryptography, ESP-NOW, Keamanan Data, Efisiensi Sumber Daya, Algoritma Enkripsi.*

Abstract

Wireless Sensor Networks (WSN) have rapidly advanced to support various automation processes, such as sensor data collection, wireless system control, and monitoring. However, one of the challenge for WSNs is ensuring data security, as the transmitted data is often sensitive and vulnerable to threats like data breaches. Another challenge is the limited resources of the devices, such as computational power, memory, and energy. To address this, lightweight cryptography (LWC) is needed to provide security while remaining efficient. This study analyzes the performance of the ChaCha20 encryption algorithm on the ESP-NOW communication protocol using ESP8266-based devices, which do not have cryptographic hardware accelerators. The study compares ChaCha20 with AES-CBC, Clefia, and Snow-V algorithms, focusing on encryption-decryption speed, power consumption, and data security. The results show that ChaCha20 excels in computational time, being the fastest algorithm for both 5 KB and 10 KB data, with the lowest energy consumption. The security test imply the randomness of ChaCha20's ciphertext is also quite good, with a Shannon entropy value of 49.375 (61.72%), and the Chi-Square test shows byte distribution uniformity (p-value 0.7415). Although ChaCha20's security is slightly less secure than AES-CBC and Clefia, it is still effective in maintaining data confidentiality. Without relying on additional hardware, the computational time and energy efficiency of ChaCha20 make it an ideal solution for WSN devices with limited resources characteristic.

Keywords: *Wireless Sensor Network, ChaCha20, Lightweight Cryptography, ESP-NOW, Data Security, Energy Efficiency, Encryption Algorithm.*

1. PENDAHULUAN

Revolusi industri dan kemajuan teknologi komunikasi nirkabel telah mendorong pengembangan sensor pintar yang kecil, hemat daya, dan berbiaya rendah. Sensor ini bekerja secara otonom dengan komunikasi nirkabel, meskipun memiliki sumber daya terbatas, seperti unit sensor, pemrosesan, dan transmisi (Hamami and Nassereddine, 2020) Wireless Sensor Network (WSN) adalah kumpulan node sensor yang saling terhubung untuk mengumpulkan, mengirim, dan menerima data secara nirkabel. Teknologi WSN telah berkembang pesat dengan berbagai fungsi, seperti pengumpulan data sensor, pengendalian sistem, dan monitoring, meskipun memiliki keterbatasan sumber daya dan komunikasi (Astuti and Wibisono, 2017).

Meskipun demikian, tantangan utama dalam implementasi WSN adalah keterbatasan sumber daya perangkat keras, seperti daya komputasi, penyimpanan, dan konsumsi energi. Di sisi lain, aspek keamanan data menjadi hal krusial, terutama pada aplikasi yang melibatkan data sensitif yang rawan terhadap ancaman seperti *data breach* atau akses tidak sah (Sarker et al., 2020). Oleh karena itu, diperlukan metode kriptografi yang mampu memberikan perlindungan pada data tanpa mengorbankan efisiensi perangkat.

Salah satu pendekatan yang sedang berkembang adalah *lightweight cryptography* (LWC), yang dirancang untuk memenuhi kebutuhan keamanan pada perangkat dengan sumber daya terbatas (Gunathilake et al., 2019). Algoritma ChaCha20, sebuah *stream cipher* berkecepatan tinggi yang dikembangkan oleh Bernstein pada tahun 2008 yang dirancang berdasarkan prinsip *rounds function* pada algoritma Salsa20, tetapi memiliki keunggulan kecepatan dibandingkan Salsa20 dengan tingkat keamanan yang setara (Kebande, 2023). ChaCha20 dapat menjadi solusi dengan kebutuhan sumber daya yang rendah sekaligus memberikan tingkat keamanan yang setara dengan algoritma modern lainnya. Dalam konteks WSN, di mana perangkat seperti ESP8266 yang tidak memiliki akselerator perangkat keras kriptografi sering digunakan, ChaCha20 menjadi kandidat yang menarik sudah dirancang cepat tanpa bergantung pada dukungan perangkat keras khusus. Menurut Jin (2022) pada penelitian sebelumnya menunjukkan bahwa algoritma seperti AES yang memanfaatkan akselerator perangkat keras dapat meningkatkan performa secara signifikan, tetapi jika fitur ini tidak tersedia, efisiensi dapat menurun drastis hingga 257,8% kecepatannya.

Dalam penelitian ini, dilakukan analisis performa algoritma enkkripsi ChaCha20 pada protokol komunikasi ESP-NOW berbasis mikrokontroler ESP8266. Penelitian ini bertujuan untuk mengevaluasi waktu komputasi algoritma, pengaruhnya terhadap penggunaan sumber daya komputasi, serta efektivitasnya dalam meningkatkan aspek kerahasiaan data. Dengan membandingkan performa ChaCha20 dengan algoritma lain seperti AES, Clefia, dan Snow-V, penelitian ini diharapkan dapat memberikan gambaran menyeluruh terkait efisiensi dan keamanan algoritma enkripsi dalam skenario sistem tertanam yang memiliki keterbatasan sumber daya.

Metodologi yang digunakan meliputi perancangan desain prototipe menggunakan tiga perangkat ESP8266 yang dilengkapi sensor DHT22 yang berfungsi dalam pengambilan data uji, lalu implementasi yang mencakup beberapa sistem terpisah yaitu perangkat node pengirim, node penerima, dan node pengukur daya, dan pengujian pada sistem meliputi beberapa aspek yaitu kecepatan enkripsi-dekripsi, konsumsi daya, dan aspek keamanan khususnya *confidentiality* (kerahasiaan) pada data. Pengujian algoritma dilakukan secara komparatif, dengan mempertimbangkan parameter pengujian yang sama, efisiensi komputasi dan aspek keamanan data melalui pengujian test *chi-square* dan pengukuran *entropy* pada *ciphertext*. Lalu pengukuran konsumsi energi menggunakan sensor arus dan daya INA219.

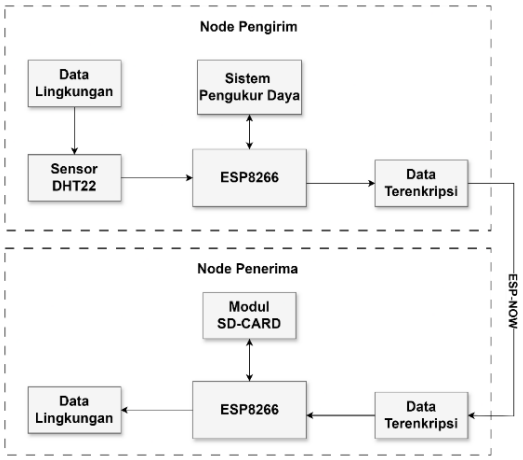
Hasil dari penelitian ini diharapkan dapat memberikan kontribusi signifikan pada pengembangan WSN, terutama dalam memilih algoritma enkripsi yang efisien, cepat, dan sesuai untuk perangkat dengan sumber daya terbatas. Selain itu, penelitian ini juga diharapkan dapat memberikan referensi bagi pengembang atau peneliti untuk mengintegrasikan metode keamanan yang lebih hemat sumber daya pada sistem WSN di masa depan.

2. PERANCANGAN DAN IMPLEMENTASI

2.1 Perancangan Sistem

Pada bagian ini dilakukan perancangan sistem

sesuai kebutuhan penelitian yaitu analisis performa dengan menggunakan perangkat dan parameter yang sama agar komparasi dilakukan setara setiap algoritma.

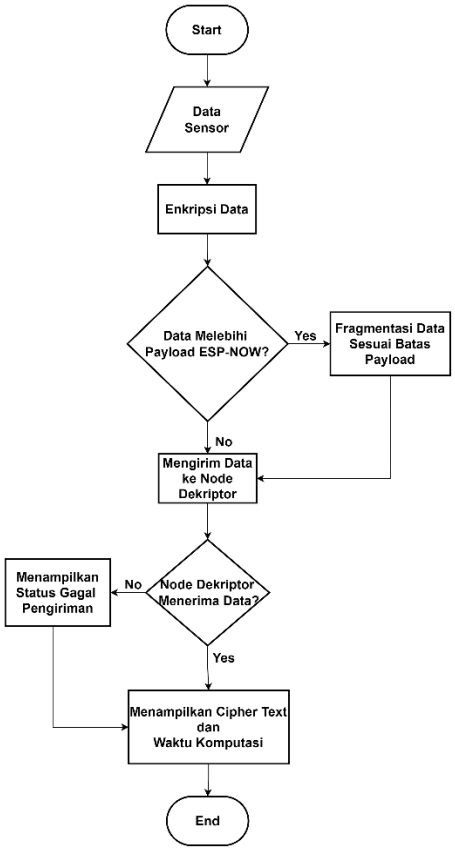


Gambar 1. Diagram Blok Perancangan Sistem

Pada gambar 1, sistem utama dirancang menjadi 2 yaitu pada node pengirim dan node penerima yang saling berkomunikasi dengan protokol ESP-NOW. Berdasarkan diagram blok diatas bahwa node pengirim menggunakan mikrokontroler ESP8266 dan akan menggunakan data uji yang berasal dari sensor DHT22 untuk dilakukan enkripsi dan mengirimkan data dalam bentuk *ciphertext* kepada node pengirim, yang dimana ketika proses komputasi akan dilakukan pengukuran daya oleh sistem pengukur daya untuk analisis penggunaan sumber daya. Selanjutnya, pada node penerima menggunakan mikrokontroler yang sama namun bertugas untuk menerima data ter-enkripsi dan melakukan dekripsi data sehingga yang diterima sudah dalam bentuk data asli. Terakhir, node pengirim akan menyimpan data yang sudah ter-dekripsi pada kartu mikro SD untuk dianalisis hasilnya.

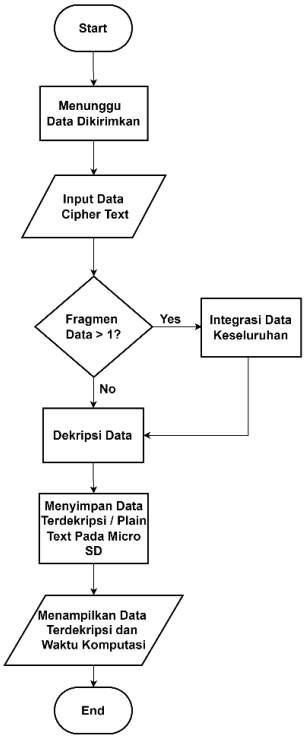
A. Perancangan Node Pengirim

Pada Gambar 5, dapat dilihat gambaran umum dari sistem dimana proses dimulai dengan input data uji dari sensor yang sudah didapatkan sebelumnya dengan ukuran 5 KB dan 10 KB, lalu menjalankan proses enkripsi dengan ChaCha20 serta beberapa algoritma enkripsi lainnya untuk komparasi. Setelah itu, sistem memeriksa apakah ukuran data yang dienkripsi melebihi batas payload protokol ESP-NOW. Jika melebihi, data akan dipecah agar sesuai dengan batas tersebut. Jika tidak, proses fragmentasi data dilewati. Setelah data siap, data dikirim ke node penerima (node dekriptor). Sistem memeriksa apakah data sudah diterima, dan jika gagal maka akan menampilkan status gagal dalam pengiriman. Jika data sudah diterima, maka proses selesai.



Gambar 5. Diagram Alir Sistem Node Pengirim

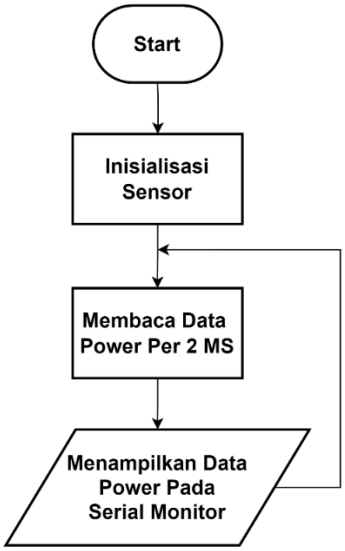
B. Perancangan Node Penerima



Gambar 6. Diagram Alir Sistem Node Penerima

Berdasarkan gambar 6, alur kerja node penerima secara umum adalah proses dimulai dengan sistem yang menunggu hingga data terenkripsi diterima. Data yang diterima dalam bentuk *cipher text* kemudian diperiksa untuk melihat apakah terdiri dari lebih dari satu fragmen. Jika lebih, maka data akan digabungkan untuk membentuk data yang lengkap. Jika hanya ada satu fragmen, penggabungan tidak diperlukan dan sistem melanjutkan ke tahap berikutnya. Setelah data utuh diperoleh, data tersebut didekripsi untuk mengembalikannya ke teks asli (plain text). Data yang telah didekripsi kemudian disimpan dan ditampilkan pada *serial monitor* bersama waktu komputasi untuk dianalisis. Lalu tentunya node penerima juga akan menjalankan fungsi dekripsi ChaCha20 serta beberapa algoritma enkripsi lainnya untuk komparasi.

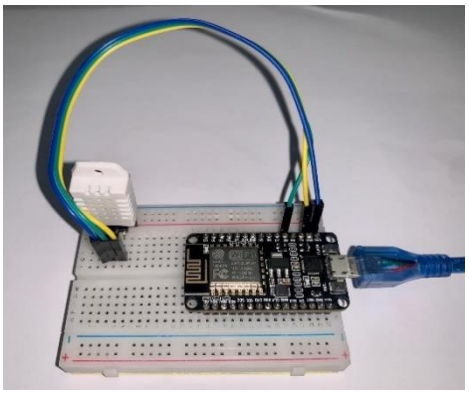
C. Perancangan Node Pengukur Daya



Gambar 7. Diagram Alir Program Node Pengukur

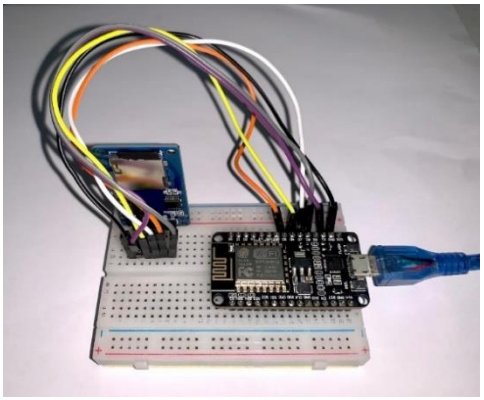
Dari gambar 7 mengenai diagram alir, proses dimulai dengan menginisialisasi sensor INA219. Setelah itu, sistem membaca data daya setiap 2 milidetik untuk mendapatkan data penggunaan daya yang hampir *real-time*. Data yang terkumpul kemudian ditampilkan pada serial monitor dan disimpan yang memungkinkan pengguna untuk memantau penggunaan daya secara langsung dan menganalisis energi yang digunakan oleh masing-masing algoritma.

2.2 Implementasi Sistem



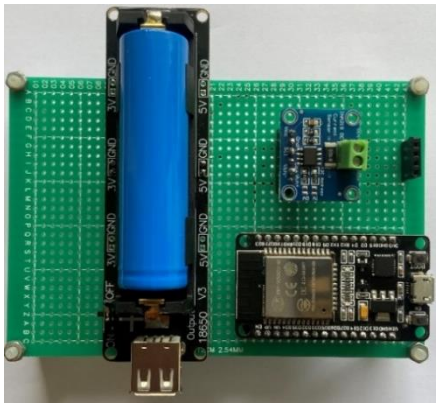
Gambar 8. Implementasi Sistem Pengirim

Berdasarkan gambar 8, menunjukkan implementasi sistem node pengirim yang dirancang menggunakan modul ESP8266 dan sensor DHT22. Sistem ini melakukan pengambilan data uji dan proses enkripsi menggunakan beberapa algoritma, lalu data dikirimkan dengan protokol ESP-NOW.



Gambar 9. Implementasi Sistem Penerima

Pada gambar 9 menunjukkan menunjukkan implementasi sistem node penerima yang dirancang menggunakan modul ESP8266, modul kartu mikro SD, dan kartu mikro SD yang terpasang. Sistem ini digunakan untuk proses dekripsi menggunakan beberapa algoritma dan menyimpan data hasil dekripsi ke dalam kartu mikro SD. Modul ESP8266, yang memiliki kemampuan komunikasi nirkabel melalui protokol ESP-NOW untuk menerima data dari node pengirim.



Gambar 10. Implementasi Sistem Pengukur Daya

Gambar 10 diatas adalah sistem pengukur yang dirancang menggunakan modul ESP32, sensor arus INA219 yang disolder pada papan PCB. Sistem ini digunakan untuk proses pengukuran daya yang akan di sambungkan pada node pengirim, dimana sumber daya akan dipisahkan. Daya yang digunakan oleh node pengirim berasal dari baterai 5V dan ESP32 menggunakan daya dari komputer sekaligus untuk melakukan analisis data dari serial monitor. Hal ini bertujuan untuk memastikan perhitungan daya pada sensor INA219 lebih akurat.

3. PENGUJIAN DAN ANALISIS

Pengujian akan dilakukan dilakukan pada 4 aspek yang menjadi tujuan dari penelitian ini. Aspek-aspek tersebut adalah melakukan validasi kerja algoritma, kecepatan komputasi algoritma, penggunaan sumber daya, dan keamanan algoritma.

A. Validitas Algoritma[NF1] dan Pengiriman Data dengan ESP-NOW

Tujuan utama dari validasi algoritma enkripsi adalah untuk memastikan bahwa algoritma dapat mengenkripsi dan mendekripsi data dengan benar sesuai spesifikasi, serta mengirimkan data terenkripsi menggunakan protokol ESP-NOW. Prosedur pengujian meliputi mengenkripsi data menggunakan beberapa algoritma enkripsi, mengirimkan data terenkripsi dari node pengirim ke node penerima melalui protokol ESP-NOW, dan mendekripsi data yang diterima untuk memastikan hasil dekripsi sesuai dengan data asli. Pengujian dilakukan sebanyak 10 kali percobaan enkripsi-pengiriman-dekripsi untuk setiap algoritma guna memastikan konsistensi hasil.

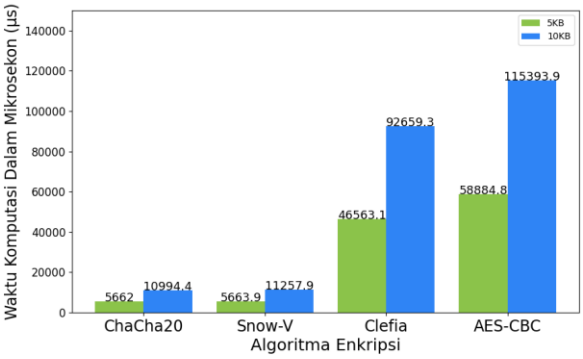
Tabel 1. Uji Validitas Algoritma

Algoritma	Skenario	Hasil Skenario	Keberhasilan
ChaCha20	Enkripsi	Ciphertext	100%
	Dekripsi	Plaintext	100%
AES-CBC	Enkripsi	Ciphertext	100%
	Dekripsi	Plaintext	100%
Snow-V	Enkripsi	Ciphertext	100%
	Dekripsi	Plaintext	100%
Clefia	Enkripsi	Ciphertext	100%
	Dekripsi	Plaintext	100%

Berdasarkan hasil evaluasi kinerja algoritma enkripsi pada Tabel 1, sistem yang menggunakan algoritma ChaCha20, AES-CBC, Snow-V, dan Clefia menunjukkan tingkat akurasi 100% yang menunjukkan keberhasilan penggunaan protokol ESP-NOW yang berbasis ESP8266 dalam proses enkripsi dan dekripsi, dengan kemampuan mengubah *plaintext* menjadi *ciphertext* dan mengembalikannya ke *plaintext* asli tanpa kesalahan. Hal ini membuktikan bahwa seluruh algoritma tersebut mampu menjaga integritas data dan dapat diandalkan dalam skenario pengujian, sehingga valid untuk digunakan dalam penerapan yang membutuhkan keamanan data.

B. Kecepatan Komputasi

Pada pengujian ini, hasil pengujian kecepatan komputasi algoritma enkripsi dianalisis dengan membandingkan waktu yang dibutuhkan untuk proses enkripsi dan dekripsi pada berbagai ukuran data uji dari beberapa algoritma. Pengukuran waktu menggunakan pustaka *chrono* yang menghitung waktu komputasi berdasarkan *clock* perangkat dengan satuan hingga mikrosekon agar data yang didapat lebih presisi.



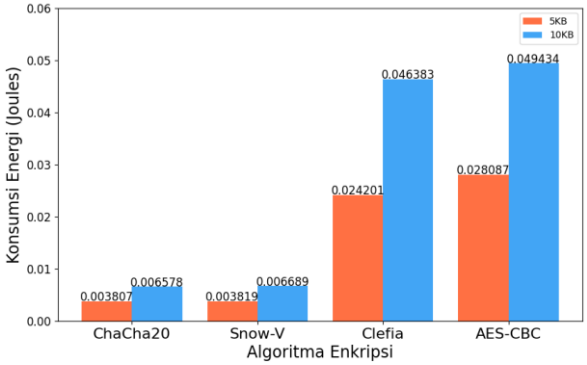
Gambar 11. Grafik Waktu Komputasi Algoritma

Berdasarkan hasil pengujian pada gambar 11, kecepatan komputasi, algoritma ChaCha20 menunjukkan waktu komputasi terendah, baik untuk data 5 KB maupun 10 KB, menjadikannya

yang tercepat dibandingkan algoritma lainnya. Algoritma lain seperti Snow-V memiliki waktu komputasi sedikit lebih lama, sementara Clefia dan AES-CBC memerlukan waktu yang lebih signifikan, dengan AES-CBC mencatatkan waktu komputasi tertinggi. Dengan demikian, ChaCha20 unggul dalam hal kecepatan dibandingkan algoritma lainnya.

C. Penggunaan Sumber Daya Komputasi

Pengujian ini mengukur konsumsi sumber daya selama proses enkripsi, dengan fokus pada penggunaan daya listrik dan energi yang diukur menggunakan sensor INA219, energi akan diukur dengan 3 metode integral yaitu *trapezoid, right rectangle,* dan *left rectangle rule* dan dirata-ratakan untuk presisi data. Data ini digunakan untuk menilai efisiensi algoritma dalam memanfaatkan sumber daya perangkat keras



Gambar 12. Grafik Konsumsi Energi Algoritma

Berdasarkan grafik konsumsi energi, algoritma ChaCha20 menunjukkan konsumsi energi terendah, yakni 0,003807 Joules untuk data 5 KB dan 0,006578 Joules untuk 10 KB dalam sekali enkripsi. Ini menandakan efisiensi tinggi dalam penggunaan sumber daya. Dibandingkan dengan algoritma lainnya, seperti Snow-V yang memiliki konsumsi energi hampir setara, ChaCha20 tetap menjadi pilihan lebih efisien, membuatnya cocok untuk aplikasi yang membutuhkan penghematan energi, seperti *Wireless Sensor Networks* (WSN).

D. Keamanan

Aspek ini bertujuan untuk memastikan algoritma enkripsi menjaga kerahasiaan data sesuai dengan prinsip *confidentiality* dalam prinsip *CIA Triad*. Pengujian menggunakan dua metrik utama: *Shannon Entropy* dan Tes *Chi-Square*. *Shannon Entropy* mengukur tingkat keacakan *ciphertext*, dengan nilai lebih tinggi menunjukkan data yang lebih sulit diprediksi polanya. Nilai *entropy* dihitung dalam rentang 0 hingga 8, di mana 8 menunjukkan keacakan sempurna. Persentase keacakan dihitung berdasarkan perbandingan nilai *entropy* terhadap nilai maksimal. Sementara itu, *Chi-Square Test* mengevaluasi distribusi byte dalam *ciphertext.* Nilai *Chi-Square* yang rendah menunjukkan distribusi yang merata dan lebih aman. P-Value yang lebih besar dari 0.05 menunjukkan distribusi *uniform* dan hasil uji dianggap lolos. Jika hasil pengujian menunjukkan nilai entropi tinggi dan distribusi merata, algoritma dianggap dapat menghasilkan *ciphertext* yang sulit ditebak dan cukup efeketif dalam menjaga kerahasiaan data, sementara entropi rendah atau distribusi tidak seragam menunjukkan kelemahan algoritma.

Berikut merupakan persamaan *Shannon entropy* pada persamaan (1) yang menghitung tingkat *randomness* pada *ciphertext*.

$$x^2 = \sum \left(\frac{(Observed\ Value - Expected\ Value)^2}{Expected\ Value} \right) \quad (1)$$

Selanjutnya perhitungan pada *Chi-Square Test* digunakan persamaan (2) digunakan dalam mengukur keseragaman distribusi data pada *ciphertext.*

$$H = -\sum_{i=1}^{n} p_i \, log_b(pi) \quad (2)$$

Berdasarkan hasil pengujian dengan metode *Shannon Entropy*, ChaCha20 menunjukkan nilai entropi 4.9375 dengan persentase *randomness* 61.72%, yang menandakan tingkat *randomness* yang cukup baik. Meskipun AES-CBC dan Clefia sedikit lebih unggul dalam hal keacakan, ChaCha20 tetap menunjukkan performa yang solid dalam menjaga keacakan data. Sebaliknya, Snow-V memiliki keacakan yang lebih rendah, menunjukkan performa yang kurang dibandingkan ketiga algoritma lainnya.

Tabel 2. Hasil Uji Shannon Entropy

Algoritma	Shannon Entropy	Max Entropy	Randomness Percentage
ChaCha20	49.375	8	61.72%
AES-CBC	54.366	8	67.96%
Snow-V	46.956	8	58.70%
Clefia	54.599	8	68.25%

Berdasarkan hasil uji *Chi-Square* pada tabel 3, ChaCha20, AES-CBC, dan Clefia menunjukkan distribusi byte yang merata dan aman, dengan nilai *p-value* yang lebih besar dari 0.05, yang berarti Algoritma tersebut lolos uji dengan distribusi yang *uniform* atau

keseragaman distribusi. Sebaliknya, Snow-V gagal dalam uji ini karena *p-value* yang sangat rendah, mengindikasikan adanya pola dalam *ciphertext* yang menurunkan keamanannya.

Tabel 3. Hasil Uji Chi-Square

Algoritma	Chi-Square Statistic	P-Value	Passes Test	Distribution
ChaCha20	240	0.7415	TRUE	Uniform
AES-CBC	232.26	0.8435	TRUE	Uniform
Snow-V	331.61	0.0009	FALSE	Non-uniform
Clefia	240	0.7415	TRUE	Uniform

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

ChaCha20 menunjukkan performa terbaik dalam waktu komputasi pada perangkat ESP8266, mengungguli algoritma lain seperti Snow-V, Clefia, dan AES-CBC, baik untuk data 5 KB maupun 10 KB. Selain itu, ChaCha20 juga paling efisien dalam hal konsumsi energi, dengan penggunaan daya yang jauh lebih rendah dibandingkan algoritma lainnya. Ini membuatnya ideal untuk aplikasi dengan sumber daya terbatas, seperti WSN. Dalam hal konfidensialitas data, ChaCha20 efektif menjaga kerahasiaan dengan hasil uji *Shannon Entropy* yang menunjukkan tingkat keacakan yang tinggi dan distribusi byte yang *uniform*, meskipun tidak sebaik AES-CBC atau Clefia. Secara keseluruhan, ChaCha20 terbukti sebagai algoritma enkripsi yang efisien dan aman untuk perangkat berbasis ESP8266 tanpa memerlukan tambahan *cryptographic hardware accelerators*.

4.2 Saran

Saran yang dapat diberikan penulis adalah Untuk mengoptimalkan hasil, pengujian dapat dilakukan dengan variasi data uji yang lebih beragam, seperti perbedaan format dan ukuran data. Selain itu, untuk analisis keamanan yang lebih mendalam, pengujian dengan metrik tambahan lainnya yang dapat membantu mengevaluasi setiap algoritma enkripsi secara lebih komprehensif.

5. DAFTAR PUSTAKA

Astuti, L.D. and Wibisono, W., 2017. Peningkatan Networklifetimepada Wireless Sensor Network Menggunakan Clustered Shortest Geopath Routing (C-SGP) Protocol. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, *4*(3), pp.148-15.

Gunathilake, N.A., Buchanan, W.J. and Asif, R., 2019, April. Next generation lightweight cryptography for smart IoT devices:: implementation, challenges and applications. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)* (pp. 707-710). IEEE.

Jin, Q., 2022. Performance Evaluation of Cryptographic Algorithms on ESP32 with Cryptographic Hardware Acceleration Feature.

Kebande, V.R., 2023. Extended-Chacha20 Stream Cipher with Enhanced Quarter Round Function. *IEEE Access*.

Sarker, V.K., Gia, T.N., Tenhunen, H. and Westerlund, T., 2020, June. Lightweight security algorithms for resource-constrained IoT-based sensor nodes. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.