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

**Naufal Farras Trikusuma1, Agung Setia Budi2**

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: 1naufalfarr@student.ub.ac.id, 2penulis.dua@ub.ac.id

**Abstrak**

*Wireless Sensor Network* (WSN) telah berkembang pesat dalam mendukung proses otomasi, seperti pengumpulan data sensor, pengendalian sistem nirkabel, dan *monitoring*. Namun, keamanan data menjadi tantangan utama karena data yang ditransmisikan sering kali bersifat sensitif dan rentan terhadap ancaman seperti data breach dan akses tidak sah. Keterbatasan sumber daya perangkat WSN, seperti daya komputasi, memori, dan energi, juga menjadi hambatan dalam penerapan algoritma kriptografi konvensional yang membutuhkan sumber daya besar. Oleh karena itu, diperlukan lightweight cryptography (LWC) yang efisien namun tetap aman. 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 terhadap algoritma AES-CBC, Clefia, dan Snow-V, dengan mengevaluasi kecepatan enkripsi-dekripsi, konsumsi daya, serta aspek keamanan terkait konfidensialitas data. Implementasi sistem menggunakan ESP8266 serta sensor INA219 untuk mengukur konsumsi daya pada sistem pengukur. Hasil pengujian menunjukkan ChaCha20 memiliki performa terbaik dalam hal waktu komputasi dan efisiensi energi, baik untuk ukuran data 5 KB maupun 10 KB. Tanpa bergantung pada akselerator perangkat keras, ChaCha20 menjadi solusi ideal bagi perangkat WSN dengan keterbatasan sumber daya. Temuan ini mendukung ChaCha20 sebagai algoritma enkripsi yang efisien untuk meningkatkan keamanan data tanpa mengorbankan efisiensi.

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

**Abstract**

*Wireless Sensor Networks (WSNs) have significantly advanced in supporting automation processes, such as sensor data collection, wireless system control, and monitoring. However, data security remains a critical challenge since transmitted data is often sensitive and vulnerable to threats such as data breaches and unauthorized access. The limited resources of WSN devices, including computational power, memory, and energy, further hinder the implementation of conventional cryptographic algorithms, which require substantial resources. Therefore, lightweight cryptography (LWC) is needed to provide efficient yet secure solutions. This study analyzes the performance of the ChaCha20 encryption algorithm on the ESP-NOW communication protocol using ESP8266-based devices, which lack cryptographic hardware accelerators. The evaluation is conducted in comparison with AES-CBC, Clefia, and Snow-V algorithms, focusing on encryption-decryption speed, energy consumption, and data confidentiality. The system implementation employs a DHT22 temperature and humidity sensor as the data source and an INA219 sensor to measure energy consumption. The results demonstrate that ChaCha20 achieves the best performance in terms of computational speed and energy efficiency for data sizes of 5 KB and 10 KB. Without relying on hardware accelerators, ChaCha20 emerges as an ideal solution for WSN devices with limited resources. These findings support ChaCha20 as an efficient encryption algorithm to enhance data security while maintaining computational and energy efficiency.*

**Keywords**: *Wireless Sensor Network, ChaCha20, Lightweight Cryptography, ESP-NOW, Data Security, Energy Efficiency, Cryptographic Hardware Accelerators.*

# 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 dan 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 dan 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, 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 karena tidak 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%.

Dalam penelitian ini, dilakukan analisis performa 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.

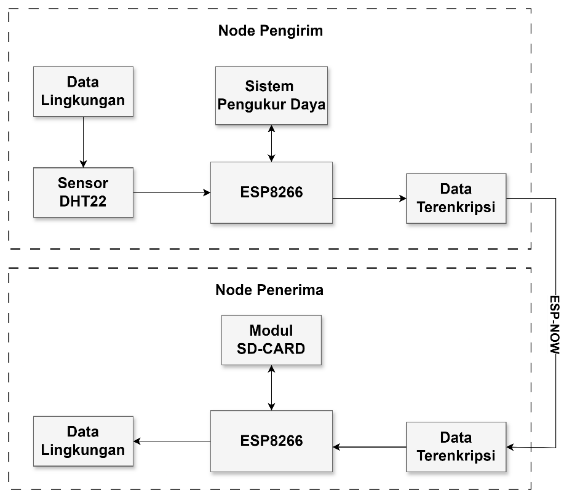
Metodologi yang digunakan meliputi perancangan desain prototipe menggunakan tiga perangkat ESP8266 yang dilengkapi sensor DHT22 yang berfungsi dalam pengambilan data uji, lalu implementasi yang mencakup beberapa sistem terpisah yaitu perangkat node pengirim, node penerima, dan node pengukur daya, dan 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 validasi data dan 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.

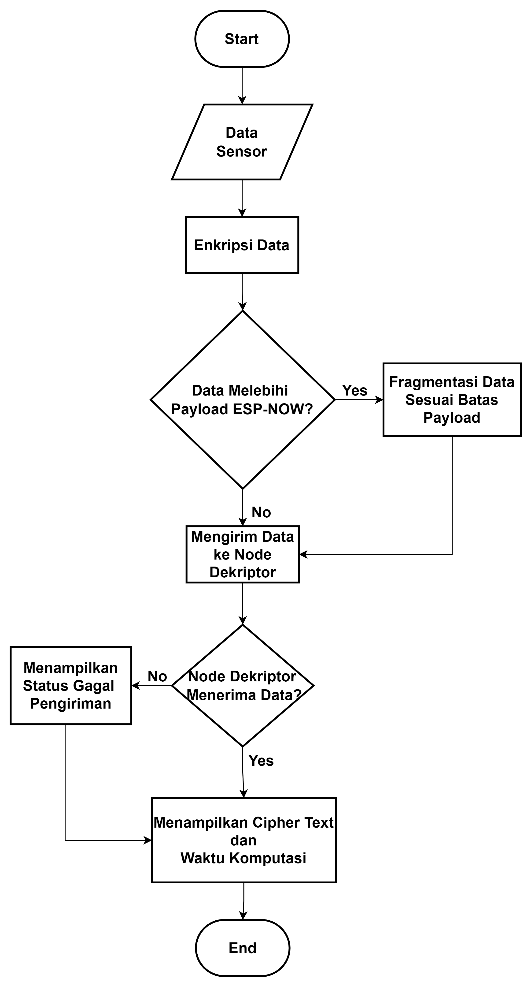


Gambar 1. Diagram Blok Perancangan Sistem

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.

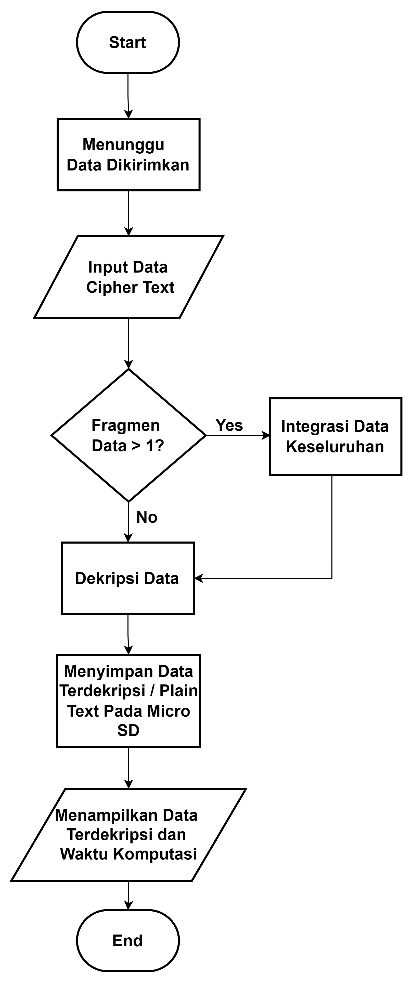
A. Perancangan Node Pengirim

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, proses selesai.



Gambar 5. Diagram Alir Sistem Node Pengirim

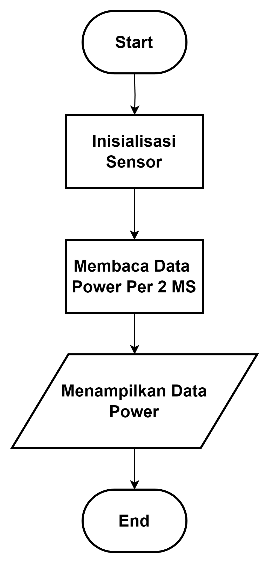
B. Perancangan Node Penerima



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.

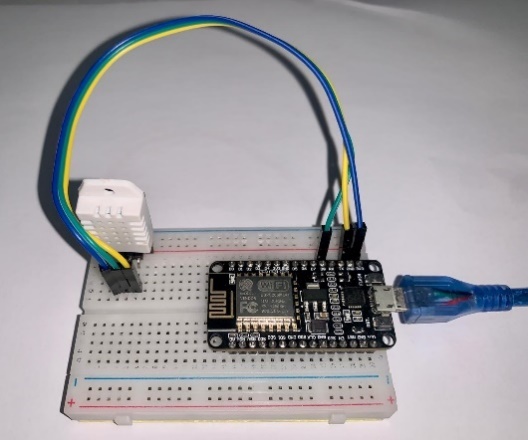
C. Perancangan Node Pengukur Daya



Gambar 7. Diagram Alir Program Node Pengukur

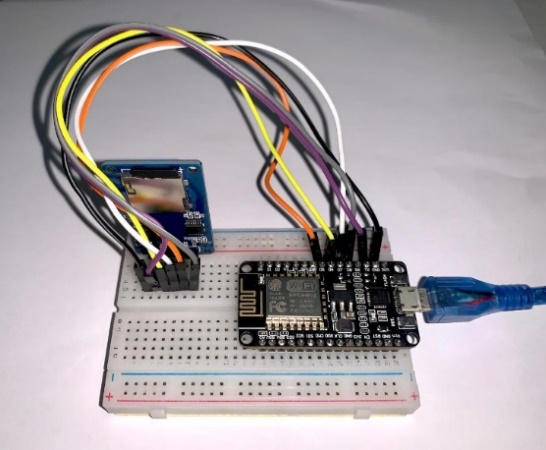
Dari gambar 7 mengenai diagram alir, proses dimulai dengan menginisialisasi sensor INA219. Setelah itu, sistem membaca data daya setiap 2 milidetik untuk mendapatkan data penggunaan daya yang hampir *real-time*. Data yang terkumpul kemudian ditampilkan pada serial monitor dan disimpan yang memungkinkan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

# 2.2 Implementasi Sistem



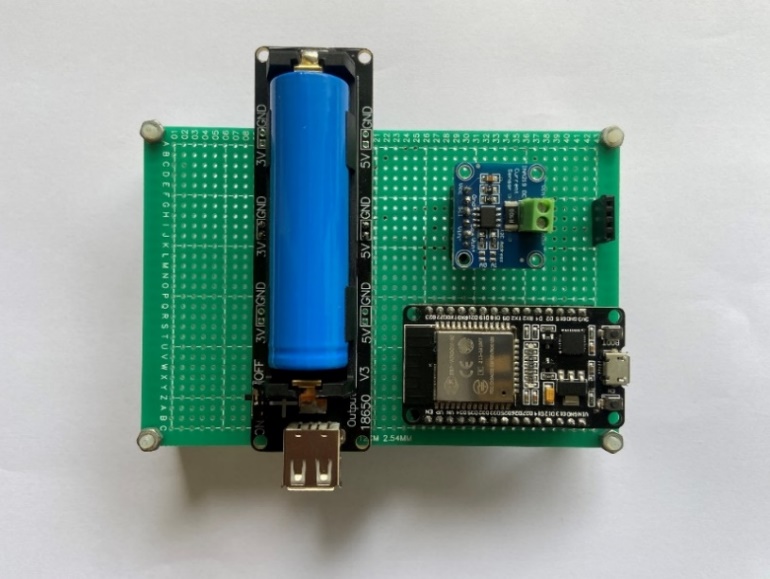
Gambar 8. Implementasi Sistem Pengirim

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 diatas 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.

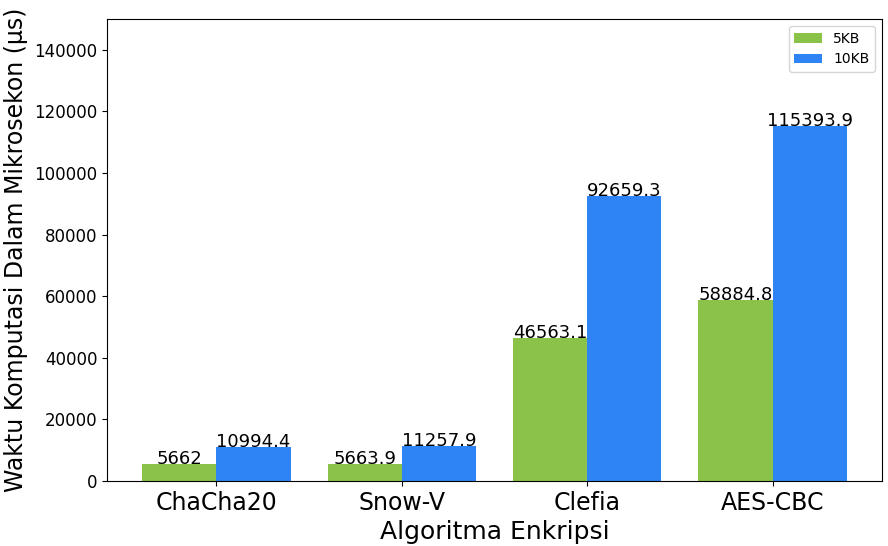
# 3. PENGUJIAN DAN ANALISIS

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

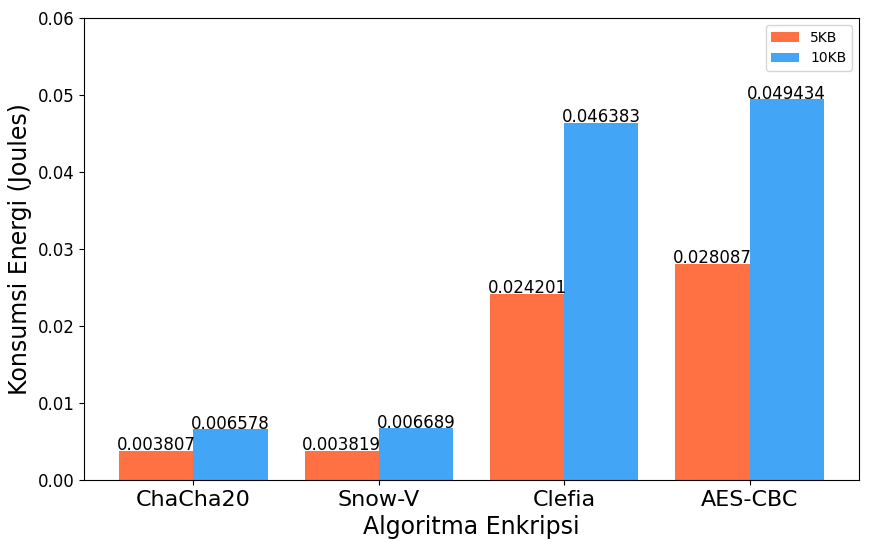
A. Validitas Algoritma

|  |  |  |  |
| --- | --- | --- | --- |
| **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% |

B. Waktu Komputasi Algoritma



B. Penggunaan Sumber Daya Komputasi



C. Keamanan

# 4. KESIMPULAN DAN SARAN

# 4.1 Kesimpulan

# 4.2 Saran

# 4. DAFTAR PUSTAKKA

Dao, S. D. & Marian, R. 2011. Optimisation of precedence-constrained production sequencing and scheduling using genetic algorithms. *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, 16-18 March, Hong Kong.

Gen, M. & Cheng, R. 2000. *Genetic Algorithms and Engineering Optimization.* John Wiley & Sons, Inc., New York.

Liliana, D. Y. & Mahmudy, W. F. 2006. Penerapan Algoritma Genetika pada Otomatisasi Penjadwalan Kuliah. *Laporan Penelitian DPP/SPP.* FMIPA Universitas Brawijaya, Malang.

Marian, R. M., Luong, L. & Dao, S. D. 2012. Hybrid genetic algorithm optimisation of distribution networks—a comparative study. *Dalam:* AO, S. I., CASTILLO, O. & HUANG, X. (editor.) *Intelligent Control and Innovative Computing.* Springer, US.

Phanden, R. K., Jain, A. & Verma, R. 2013. An approach for integration of process planning and scheduling. *International Journal of Computer Integrated Manufacturing*, 26(4), 284-302.

Ridok, A. 2014. Peringkasan dokumen Bahasa Indonesia berbasis non-negative matrix factorization. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 1(1), 39-44.

Tala, F. Z. 2003. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. *Ph.D. Thesis*. Universiteit van Amsterdam.

Wang, L. 2007. *Process planning and scheduling for distributed manufacturing.* Springer, London.

Wibawa, A. P., Nafalski, A. & Mahmudy, W. F. 2013. Javanese `speech levels machine translation: improved parallel text alignment based on impossible pair limitation. *IEEE International Conference on Computational Intelligence and Cybernetics*, 3-4 December, Yogyakarta, Indonesia. 16-20.