**Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)** p-ISSN: 2355-7699
Vol. 12, No. 3, Juni 2025, hlm. 525-532 e-ISSN: 2528-6579

# IMPLEMENTASI ALGORITMA ENKRIPSI SPECK UNTUK PENGAMANAN *MNEMONIC PHRASE* PADA *CRYPTOCURRENCY WALLET*

**Dimas Tri Mustakim[*1], Ari Kusyanti[2] , Primantara Hari Trisnawan[3]**

[1,2,3]Universitas Brawijaya Malang
Email: [1]tri.dimas@student.ub.ac.id, [2]ari.kusyanti@ub.ac.id, [3]prima@ub.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 29 November 2024, diterima untuk diterbitkan: 18 Juni 2025)

**Abstrak**

*Cryptocurrency wallet* memiliki peran penting dalam menyimpan dan mengelola aset digital pada jaringan *blockchain*. Keamanan *wallet*, terutama dalam penyimpanan frasa mnemonik sebagai kunci utama, menjadi aspek krusial yang perlu diperhatikan. Penelitian ini bertujuan untuk mengimplementasikan algoritma enkripsi Speck dalam mengamankan penyimpanan frasa mnemonik pada *Hierarchical Deterministic (HD) Wallet*. Metode yang digunakan adalah eksperimen implementatif dengan merancang prototipe sistem *Wallet* yang menggunakan algoritma Speck. Implementasi mencakup komponen-komponen utama *wallet* serta integrasi algoritma Speck untuk pengamanan frasa mnemonik. Pengujian dilakukan melalui analisis properti berupa *avalanche effect*, *uniformity test*, dan *entropy test*, serta pengujian keamanan menggunakan metode *brute force*. Hasil penelitian menunjukkan bahwa algoritma Speck dapat diimplementasikan dengan baik pada sistem *Wallet*, mampu menghasilkan *ciphertext* dengan properti statistik yang baik, dan dapat mengamankan data mnemonik. Pengujian kinerja menunjukkan hasil yang sangat baik, dengan waktu rata-rata eksekusi di bawah 2 mikro detik dan penggunaan memori sekitar 3 MB. Uji *avalanche effect* menghasilkan nilai yang mendekati 0,5 yang menunjukkan sensitivitas yang baik terhadap perubahan *input*. *Uniformity test* menunjukkan distribusi *ciphertext* yang seragam dengan nilai *chi-square* di bawah ambang kritis. Pengujian *entropy* menghasilkan peningkatan signifikan dari rata-rata 4 bit per *byte* pada *plaintext* menjadi sekitar 6 bit per *byte* pada *ciphertext*. Pengujian keamanan melalui *brute force* menunjukkan ketahanan yang sangat baik, dengan estimasi waktu pemecahan kunci mencapai $10^{67}$ tahun. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan solusi keamanan yang efisien untuk *cryptocurrency wallet*.

**Kata kunci**: *Cryptocurrency Wallet*, Frasa Mnemonik, Enkripsi, Algoritma Speck

# *IMPLEMENTATION OF THE SPECK ENCRYPTION ALGORITHM FOR SECURING MNEMONIC PHRASES IN CRYPTOCURRENCY WALLETS*

***Abstract***

*Cryptocurrency wallets play a crucial role in storing and managing digital assets on blockchain networks. The security of wallets, especially in storing mnemonic phrases as master keys, is a critical aspect that requires attention. This research aims to implement the Speck encryption algorithm for securing mnemonic phrase storage in Hierarchical Deterministic (HD) Wallets. The method used is an implementative experiment, designing and developing a Wallet system prototype that use Speck. The implementation encompasses the main components of the wallet, as well as integrating the Speck algorithm for sensitive data encryption. Testing was conducted through cryptographic analysis including avalanche effect, uniformity test, and entropy test, as well as security testing using brute force. The results show that the Speck algorithm can be effectively implemented in the Wallet system and is capable of generating ciphertext with good statistical properties, and effectively securing mnemonic data. Performance testing yielded excellent results, with average execution times below 2 microseconds and memory usage of approximately 3 MB.. The avalanche effect test yielded values approaching 0.5, indicating good sensitivity to input changes. The uniformity test showed uniform ciphertext distribution with chi-square values below the critical threshold. The entropy test resulted in a significant increase from an average of 4 bits per byte in plaintext to about 6 bits per byte in ciphertext. Security testing through brute force demonstrates excellent resilience, with an estimated key-breaking time reaching 10^67 years. This research contributes to the development of efficient security solutions for cryptocurrency wallets.*

**Keywords**: *Cryptocurrency Wallet, Mnemonic Phrase, Encryption, Speck*

## 1. PENDAHULUAN

*Cryptocurrency* merupakan bentuk mata uang digital yang menggunakan teknik kriptografi dan didasarkan pada teknologi *blockchain* (Arias-Oliva, Pelegr'in-Borondo and Mat'ias-Clavero, 2019). Aset digital ini dapat dikirim antar pihak dan digunakan sebagai alat tukar. Salah satu karakteristik *cryptocurrency* adalah desentralisasi, yang menyediakan metode pembayaran yang tidak dikendalikan otoritas pusat. Teknologi *blockchain* memungkinkan transaksi yang aman tanpa memerlukan perantara (Fang et al., 2022).

Keunggulan utama *cryptocurrency* terletak pada potensinya untuk merevolusi lanskap keuangan dengan menawarkan transparansi dan kemandirian (Chahooki and KJ, 2023). Teknologi ini telah menjadi populer sebagai aset investasi karena potensinya untuk mendapatkan keuntungan finansial dan komunitas pengguna yang terus bertambah.

*Blockchain*, teknologi dibalik *cryptocurrency*, menggunakan kunci kriptografi publik/privat untuk melakukan autentikasi dan autorisasi pengguna dalam jaringan. Kunci privat digunakan untuk membuktikan kepemilikan aset digital, sedangkan kunci publik digunakan untuk menerima aset digital yang dikirim. Hilangnya kunci kriptografi tersebut dapat menyebabkan kehilangan aset secara permanen.

*Crypto wallet*, berperan untuk memudahkan pengguna dalam melakukan segala hal yang berhubungan dengan *cryptocurrency*. Fungsi utamanya adalah untuk menyimpan kunci pribadi yang dibutuhkan untuk mengakses dan mengelola aset pengguna. *Wallet* juga memungkinkan pengguna untuk membuat, menyimpan, dan mengelola kunci kriptografi, menandatangani transaksi, dan melacak aset *crypto* (Uddin, Mannan and Youssef, 2021). *Crypto wallet* memiliki peran yang sangat pentingn dalam berinteraksi dengan jaringan *blockchain*. *Wallet* menyediakan identitas unik bagi pengguna, dengan kriptografi publik, sehingga memungkinkan pengiriman dan penerimaan aset.

*Hierarchical Deterministic (HD) Wallet* merupakan jenis *crypto wallet* yang paling banyak digunakan. *HD wallet* memungkinkan pembuatan beberapa akun turunan dari satu kunci utama (Lesavre et al., 2019). HD wallet menggunakan frasa mnemonik atau yang juga biasa disebut *seed phrase* sebagai kunci induk. Mnemonik sangat penting untuk memulihkan akses ke *wallet* jika terjadi kehilangan, pencurian, atau kerusakan pada *wallet* asli.

*Crypto wallet* menawarkan kemudahan dalam pengelolaan aset digital. Namun, keamanan *crypto wallet* sangat bergantung pada perlindungan yang memadai terhadap informasi sensitif seperti mnemonik. Kehilangan informasi ini dapat mengakibatkan hilangnya aset digital secara permanen.

Menyimpan mnemonik dengan aman merupakan hal yang sangat penting karena pada dasarnya mnemonik mengandung kunci mata uang digital. Jika mnemonik jatuh ke tangan yang salah, orang yang tidak berwenang dapat berpotensi mendapatkan akses ke aset milik pengguna. Untuk mengamankan akses pada data mnemonik, *crypto wallet* menggunakan teknik enkripsi. Enkripsi ini berperan untuk mencegah informasi tersebut jatuh pada tangan yang salah.

Terdapat berbagai algoritma enkripsi yang tersedia, meskipun tidak semuanya cocok untuk keperluan khusus dalam mengamankan *crypto wallet*. Penelitian ini mengusulkan penggunaan algoritma Speck untuk mengamankan informasi sensitif pada *crypto wallet*. Algoritma Speck merupakan salah satu algoritma kriptografi ringan yang dikembangkan oleh Badan Keamanan Nasional (NSA) Amerika Serikat. Algoritma ini memiliki kelebihan berupa konsumsi sumber daya yang rendah dan kinerja komputasi yang tinggi. Algoritma ini juga mengungguli algoritma Simon dan AES dalam hal *delay* komunikasi dan penggunaan memori (Yustiarini, Dewanta and Nuha, 2022).

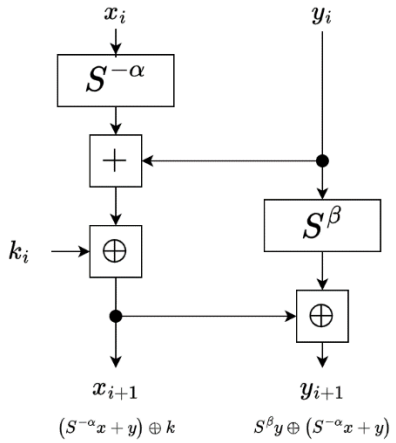
Penelitian ini menawarkan solusi alternatif dalam mengamankan mnemonik pada *crypto wallet*. Penggunaan kriptografi ringan Speck memungkinkan peningkatan keamanan pada perangkat dengan sumber daya terbatas, seperti kartu pintar atau perangkat keras khusus. Hal ini membuka peluang untuk pengembangan solusi penyimpanan *cryptocurrency* yang lebih efisien dan aman pada berbagai jenis perangkat. Selain itu, penelitian ini berkontribusi pada upaya memperluas adopsi *cryptocurrency* dengan menyediakan metode keamanan yang dapat diterapkan pada lingkungan dengan keterbatasan komputasi. Implementasi Speck juga berpotensi meningkatkan kinerja *wallet* tanpa mengorbankan tingkat keamanan, yang pada gilirannya dapat meningkatkan kualitas pengalaman pengguna. Lebih lanjut, penelitian ini memberi wawasan berharga tentang bagaimana kriptografi ringan dapat diintegrasikan ke dalam teknologi *crypto wallet*, membuka jalan bagi inovasi lebih lanjut dalam keamanan aset digital. Dengan demikian, studi ini tidak hanya relevan untuk pengembangan *crypto wallet*, tetapi juga memiliki implikasi lebih luas dalam meningkatkan aksesibilitas dan keamanan teknologi ini secara keseluruhan.

## 2. DASAR TEORI

### 2.1. Speck

Speck merupakan keluarga dari *lightweight block cipher* yang dirilis oleh *National Security Agency* (NSA) Amerika Serikat (Beaulieu et al., 2013). Terdapat sepuluh varian berbeda dengan ukuran blok dan kunci yang berbeda-beda pula. Algoritma ini menawarkan kelebihan berupa performa yang sangat baik pada platform perangkat lunak dan fleksibel.

Speck beroperasi pada blok 32, 48, 64, 96, dan 128 serta mendukung ukuran kunci mulai dari 64 bit hingga 256 bit sehingga terdapat 10 total varian (Beaulieu et al., 2015). Speck memiliki jumlah *round* yang berbeda-beda tergantung dengan ukuran blok dan kunci yang digunakan. Pada Speck128/256, digunakan jumlah *round* sebanyak 34. Putaran (*rounds*) terdiri dari dua fungsi putaran AND-RX yang diterapkan secara bergantian pada bagian blok dalam struktur jaringan Feistel. Algoritma ini memiliki fungsi *key schedule* yang sama dengan *round function*. Fungsi putaran terdiri dari operasi XOR, penambahan modulo, dan rotasi bit. Fungsi putaran Speck dapat dilihat pada Gambar 1.




Gambar 1. *Round Function* Algoritma Speck.
Sumber: (Sleem and Couturier, 2021)

Fungsi tersebut beroperasi pada dua *words* n-bit $(x, y)$ dan sebuah *round key* $k$. Di mana $S^{-\alpha}$ dan $S^{-\beta}$ adalah pergeseran melingkar kanan dan kiri sebanyak alpha dan beta bit, + adalah penjumlahan modulo $2^n$, dan  adalah XOR. Jumlah rotasi yang digunakan sebesar 8 dan 3 bit untuk semua varian algoritma kecuali variasi Speck32/64 yang menggunakan jumlah rotasi 7 dan 2.

Proses *key schedule* dimulai dengan membuat *master key* dengan jumlah antara 2, 3, atau 4 (berdasarkan variasi yang digunakan) yang didapat dari *encryption key*. Hasil inisialisasi tersebut kemudian digunakan untuk membuat *round keys* lainnya menggunakan *round function* pada Gambar 1 dengan jumlah sesuai total *rounds* yang digunakan. Masing-masing kunci yang dihasilkan kemudian digunakan pada masing-masing *round* untuk proses enkripsi dan dekripsi. Formula yang digunakan adalah sebagai berikut.

$$l_{i+m-1} = (\mathrm{S}^{-\alpha} l_i + k_i) \oplus i \quad (1)$$

$$k_{i+1} = \mathrm{S}^{-\beta} k_i \oplus l_{i+m-1} \quad (2)$$

Pada proses enkripsi, dilakukan dengan membuat data $x$ dan $y$ awal yang didapat dari *plaintext* yang dipisah menjadi dua. Data tersebut kemudian masuk pada *round function* seperti pada Gambar 1 menggunakan *round keys* dengan indeks yang sama. Hasil x dan y terakhir merupakan hasil enkripsi algoritma ini.

### 2.2 *Hierarchical Deterministic (HD) Wallet*

*Hierarchical Deterministic (HD) Wallet* merupakan sistem pengelolaan kunci yang menggunakan konsep hierarki deterministik untuk membuat kunci turunan dari sebuah kunci induk. Semua kunci dan alamat berasal dari satu kunci induk.

Dengan menggunakan HD *wallet*, proses pencadangan hanya perlu menggunakan satu *seed* untuk memulihkan semua kunci dan alamat, dapat meningkatkan privasi dengan memungkinkan untuk membuat alamat baru untuk setiap transaksi sehingga menyulitkan pelacakan, serta memungkinkan pengelolaan dana yang terorganisir di berbagai akun dan sub-akun.

### 2.3 Frasa Mnemonik

Frasa mnemonik (*mnemonic phrase*), juga dikenal dengan *seed phrase*, merupakan representasi yang dapat dibaca oleh manusia dari kunci kriptografi yang digunakan pada *HD Wallet*. Mnemonik terdiri dari serangkaian kata yang berfungsi sebagai mekanisme pencadangan dan pemulihan untuk kunci pada *crypto wallet*. Penggunaan mnemonik dapat memudahkan pengguna untuk mengingat kunci daripada menggunakan teks acak yang susah untuk diingat. Dengan memasukkan frasa mnemonik ke dalam aplikasi dompet yang kompatibel, pengguna dapat membuat ulang kunci pribadi mereka dan mendapatkan kembali akses ke dana mereka (Shah et al., 2023). Mnemonik merupakan informasi yang sangat penting, karena siapa pun yang mengetahui frasa ini dapat memperoleh kontrol penuh terhadap aset kripto yang disimpan dalam *wallet* (Sans, Liu and Oh, 2023).

### 2.4 Bitcoin Improvement Proposal (BIP)

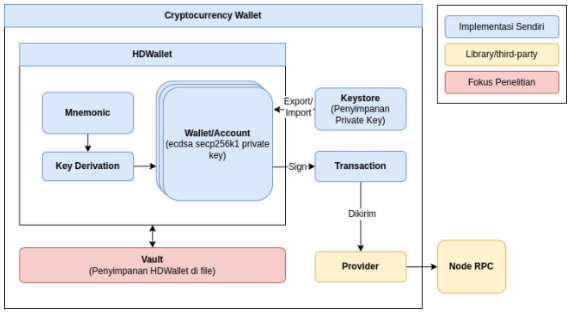
BIP merupakan proposal formal yang digunakan untuk mengganti atau memperbaiki protokol Bitcoin. BIP digunakan untuk mengusulkan perubahan pada protokol, termasuk spesifikasi teknis, pedoman, dan proses yang terkait dengan pengembangan Bitcoin. Pada penelitian ini, spesifikasi BIP digunakan sebagai referensi pembuatan mekanisme fungsional dari prototipe *crypto wallet* yang akan dibuat. Dari beberapa spesifikasi yang tersedia, tiga yang utama digunakan yakni: BIP32 yang menjelaskan standar untuk *HD Wallet* (Wuille, 2012), BIP39 yang menjelaskan mekanisme pembuatan frasa mnemonik (Palatinus et al., 2013), dan BIP44 yang menjelaskan tentang hierarki logis dari *HD Wallet* (Palatinus and Rusnak, 2014). Dengan mengikuti spesifikasi ini, dapat dipastikan prototipe implementasi akan kompatibel dengan implementasi *wallet* lainnya.

## 3. PERANCANGAN

### 3.1 Perancangan Sistem

Tahap perancangan sistem merupakan fase krusial dalam pengembangan perangkat lunak yang melibatkan penyusunan model dan arsitektur sistem. Proses ini mencakup identifikasi kebutuhan sistem, deskripsi fungsional, serta perumusan strategi pengujian. Pendekatan sistematis ini memastikan bahwa setiap aspek perancangan berkontribusi terhadap pencapaian tujuan.

Perancangan sistem dilakukan berdasarkan pada analisa kebutuhan yang telah didefinisikan sebelumnya. Penelitian ini merupakan penelitian implementatif, dan sistem *wallet* dirancang dengan memisahkan fungsi dan algoritma yang diperlukan menjadi komponen-komponen atau modul terpisah. Berdasarkan hal tersebut, terdapat beberapa modul yang diperlukan yaitu: modul mnemonik, *key derivation*, *wallet*, *keystore*, *transaction*, dan *vaults*. Gambar 2 menunjukkan gambaran umum sistem yang akan dibuat.



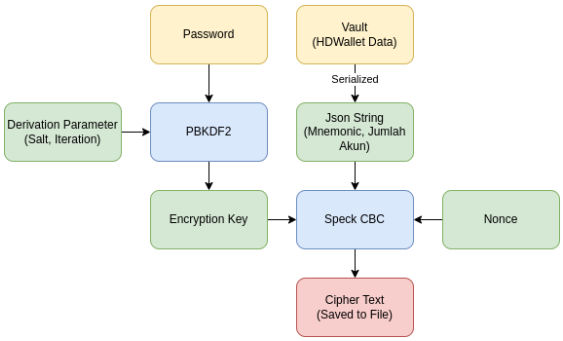

Gambar 2. Gambaran Rancangan Sistem *Wallet*.

Dari Gambar 2 tersebut dapat dilihat jika sistem yang akan dikembangkan terdiri dari beberapa modul yang saling terintegrasi untuk mengelola aset *crypto* secara aman dan efisien. *HDWallet* berperan sebagai komponen utama yang mengelola mnemonik dan menurunkan akun baru, sementara *Wallet* bertanggung jawab atas penyimpanan kunci dan penandatanganan transaksi. Modul *Mnemonic* mengurus pembuatan dan pengelolaan frasa mnemonik, sedangkan *Key Derivation* menghasilkan kunci dari mnemonik tersebut. *Keystore* menyediakan penyimpanan terenkripsi untuk kunci privat individual, dan *Vault* mengamankan mnemonik serta *metadata* penting lainnya. Komponen *Transaction* memfasilitasi pembuatan transaksi, dan *Provider* menjembatani koneksi ke jaringan *cryptocurrency*. Keseluruhan arsitektur ini dirancang untuk memberikan keamanan berlapis, fleksibilitas dalam manajemen akun, dan kemudahan dalam melakukan transaksi.

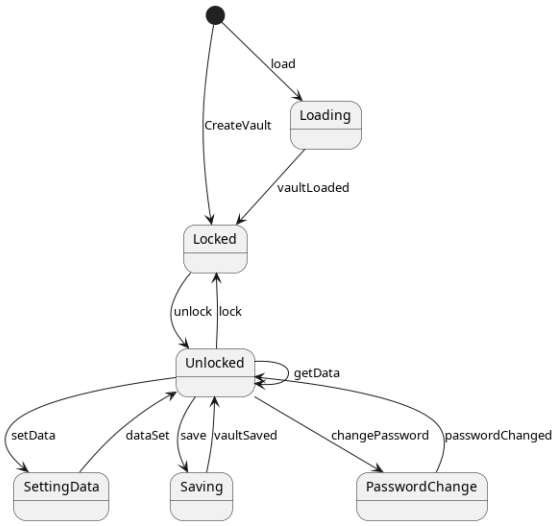
### 3.2 Perancangan Penyimpanan Mnemonik

Komponen *vaults* merupakan salah satu komponen utama yang berfungsi untuk menyimpan frasa mnemonik dengan aman pada sistem. Komponen ini bertanggung jawab untuk mengenkripsi dan mendekripsi data menggunakan algoritma Speck128/256 dalam mode CBC dan mekanisme *hashing* kunci menggunakan PBKDF2. Hal ini memastikan bahwa data yang disimpan tetap rahasia dan hanya dapat diakses oleh pemilik asli. Alur mekanisme enkripsi mnemonik yang dilakukan dapat dilihat Pada Gambar 3.

Alur penggunaan komponen *vault* juga dapat dijelaskan menggunakan *state diagram* yang dapat dilihat pada Gambar 4. Mekanisme pada diagram tersebut dirancang untuk mengelola dan melindungi data yang disimpan dalam *vault* dengan cara yang aman dan terstruktur. Pada kondisi *unlocked*, pengguna memiliki akses penuh untuk melihat dan mengubah data, termasuk mengganti kata sandi enkripsi. Ketika data atau kata sandi diubah, *vault* akan mengenkripsi ulang data tersebut dan menyimpannya dalam memori sementara hingga pengguna melakukan perintah untuk menyimpan data, yang kemudian menyimpan data tersebut ke dalam *file* secara permanen. Tujuan dari mekanisme ini adalah untuk memastikan bahwa data selalu terlindungi melalui proses enkripsi dan hanya dapat diakses atau diubah oleh pengguna yang memiliki izin, sehingga menjaga integritas dan kerahasiaan informasi yang disimpan.




Gambar 3. Alur Enkripsi Mnemonik.




Gambar 4. *State Diagram* Komponen *Vaults*.

### 3.1 Perancangan Pengujian

Pada tahap ini dilakukan pengujian terhadap prototipe sistem yang telah dibuat. Penulis melakukan beberapa macam pengujian. Jika dibagi berdasarkan fungsinya, yaitu menguji fungsionalitas sistem, dan menguji keabsahan penggunaan algoritma yang diusulkan pada kasus ini.

Pada pengujian fungsionalitas sistem, dilakukan pengujian unit dan pengujian integrasi. Pengujian unit digunakan untuk menguji setiap komponen berdasarkan *test vector* yang didapatkan pada spesifikasi BIP dan jurnal algoritma Speck. Kemudian pengujian integrasi dilakukan untuk mengecek fungsi sistem secara keseluruhan. Hal yang dicek meliputi fungsi membuat dan menyimpan mnemonik, akun, hingga melakukan transaksi *cryptocurrency*.

Selanjutnya pada pengujian algoritma kriptografi, dilakukan beberapa pengujian meliputi pengujian performa, dan pengujian statistik untuk mengukur properti statistik berupa *avalanche effect*, *uniformity*, dan *entropy*. Pengujian statistik ini memberikan wawasan tentang seberapa baik algoritma dapat menyembunyikan pola dan menghasilkan *output* yang acak. Pengujian statistik berfokus pada karakteristik intrinsik dari algoritma dan bagaimana *output* yang dihasilkan mematuhi sifat-sifat teoritis yang diinginkan. Pengujian ini memberikan wawasan penting mengenai kualitas dan kekuatan algoritma dalam aspek-aspek seperti difusi, keacakan, *confusion*, dan keseragaman distribusi *output*.

## 4. HASIL DAN PEMBAHASAN

Pada bagian ini dibahas hasil dari implementasi dari algoritma Speck untuk mengamankan frasa mnemonik pada *cryptocurrency wallet*. Algoritma ini diimplementasikan pada komponen *vaults* seperti pada rancangan pada Gambar 2 dengan alur enkripsi seperti pada Gambar 3.

### 4.1. Pengujian *Test Vector*

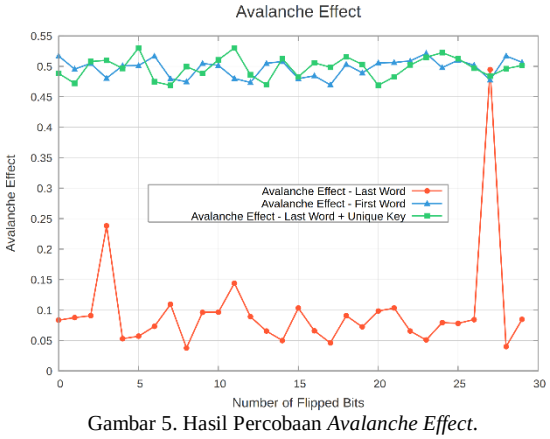
Pengujian *test vector* untuk algoritma Speck dilakukan dengan menggunakan *test vector* dari jurnal referensi asli. Berdasarkan pengujian ini, implementasi algoritma Speck yang dibuat oleh penulis sesuai dengan spesifikasi pada jurnal, yang dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian *Test Vector*

| tipe | sistem | *test vector* (jurnal) |
|---|---|---|
| *Key* | 1f1e1d1c1b1a1918 | 1f1e1d1c1b1a1918 |
| | 1716151413121110 | 1716151413121110 |
| | 0f0e0d0c0b0a0908 | 0f0e0d0c0b0a0908 |
| | 0706050403020100 | 0706050403020100 |
| *plaintext* | 65736f6874206e49 | 65736f6874206e49 |
| | 202e72656e6f6f70 | 202e72656e6f6f70 |
| *ciphertext* | 4109010405c0f53e | 4109010405c0f53e |
| | 4eeeb48d9c188f43 | 4eeeb48d9c188f43 |

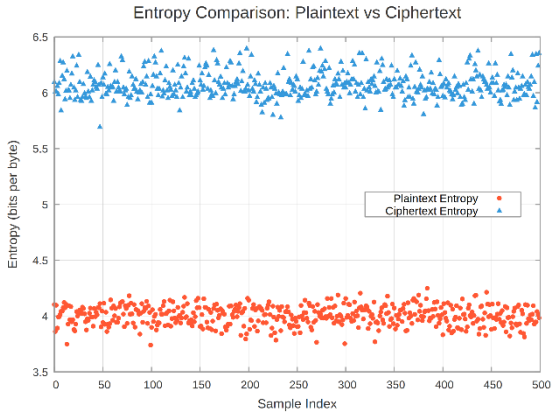
### 4.2. Pengujian Statistik

Terdapat beberapa properti statistik yang diuji. Pertama, ada *avalanche effect*. Pengujian ini dilakukan dengan melakukan perubahan kata pada mnemonik dengan panjang 24 kata pada bagian awal dan akhir kata dengan kata lainnya. Hasil dari percobaan ini dapat dilihat pada Gambar 5. Pada gambar tersebut dapat dilihat bahwa secara umum, algoritma Speck menunjukkan ketahanan yang baik terhadap perubahan *input*, dengan efek *avalanche* yang konsisten berkisar antara 0,47-0,53 untuk mayoritas kasus. Namun, perbedaan yang mencolok terlihat pada modifikasi kata terakhir mnemonik yang menghasilkan pola yang berbeda, dengan nilai yang jauh lebih rendah, berkisar antara 0,05-0,5.



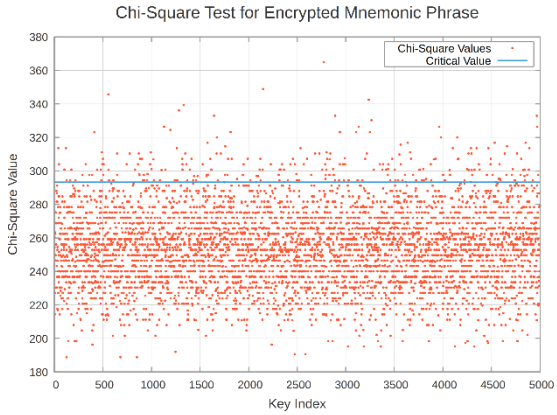

Gambar 5. Hasil Percobaan *Avalanche Effect*.

Fenomena ini dapat dijelaskan melalui karakteristik mode operasi seperti CBC dan CTR yang digunakan dalam implementasi. Dalam CBC, setiap blok *ciphertext* bergantung pada blok *plaintext* sebelumnya. Konsekuensinya, modifikasi pada kata terakhir mnemonik hanya mempengaruhi blok-blok terakhir dalam proses enkripsi, sementara blok-blok awal tetap tidak terpengaruh. Hal ini mengakibatkan propagasi perubahan yang terbatas dan efek *avalanche* yang rendah untuk modifikasi kata terakhir. Namun, penting untuk dicatat bahwa permasalahan nilai efek *avalanche* yang rendah ketika hanya data terakhir yang diubah tidak terjadi ketika kunci yang digunakan juga diganti, serta dalam praktiknya mnemonik yang *valid* satu dengan yang lain sangat berbeda satu dengan yang lain.




Gambar 6. Hasil Pengukuran Entropi.

Selanjutnya, pada pengujian entropi dengan hasil pada Gambar 6. Terdapat peningkatan entropi yang signifikan. Entropi *ciphertext* secara konsisten lebih tinggi dibandingkan *plaintext*, dengan nilai berkisar antara 5,75 hingga 6,5 bit per *byte*, sementara entropi *plaintext* berada dalam rentang 3,75 hingga 4,25 bit per *byte*. Distribusi entropi *ciphertext* menunjukkan keseragaman yang stabil, dengan mayoritas nilai berada antara 6,0 hingga 6,25 bit per *byte*. Peningkatan entropi rata-rata sekitar 2 bit per *byte* ini mengindikasikan efektivitas algoritma Speck dalam mengacak data dan meningkatkan keamanan mnemonik secara substansial. Nilai entropi *ciphertext* yang mendekati 6,5 bit per *byte* menunjukkan tingkat pengacakan yang baik, mendekati nilai maksimal entropi pada 8 bit per *byte*.

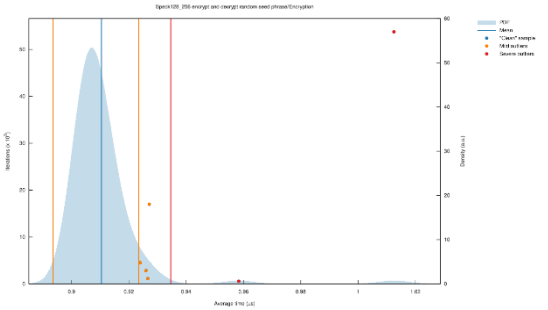



Gambar 7. Hasil Uji *Uniformity*.

Terakhir, dilakukan pengujian *uniformity* dengan hasil yang dapat dilihat pada Gambar 7. Berdasarkan pengujian, sebesar 95% hasil enkripsi mnemonik memiliki hasil *chi-square test* dibawah nilai kritis, yang menunjukkan persebaran yang merata (*uniform*) dari *ciphertext*.

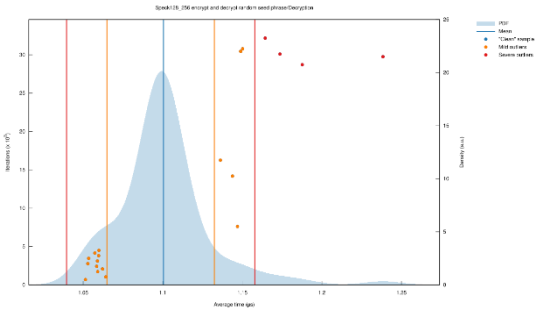
### 4.3. Pengujian Kecepatan Eksekusi

Pengujian kinerja dilakukan dengan mengukur kecepatan eksekusi enkripsi dan dekripsi yang dapat dilihat pada Gambar 8 untuk enkripsi dan Gambar 9 untuk dekripsi. Gambar tersebut menunjukkan bahwa waktu rata-rata enkripsi ada di sekitar 1 mikro detik, dan waktu dekripsi sedikit lebih lama pada angka 1,1 mikro detik.



Gambar 8. Kinerja Enkripsi Algoritma Speck.

Hasil pengujian waktu eksekusi yang sangat baik ini menunjukkan kinerja algoritma yang sangat baik dalam kasus enkripsi frasa mnemonik pada *cryptocurrency wallet*. Terutama jika akan digunakan pada perangkat khusus dengan sumber daya terbatas.



Gambar 8. Kinerja Dekripsi Algoritma Speck.

### 4.3. Pengujian Penggunaan Memori

Pengujian ini dilakukan dengan menggunakan menjalankan program enkripsi dan dekripsi Speck dengan berbagai panjang mnemonik (12, 15, 18, 21, dan 24 kata), serta mengukur penggunaan memori menggunakan program Linux yaitu dengan membaca */proc filesystem* dan pmap. Hasil pengujian menunjukkan bahwa penggunaan memori algoritma Speck relatif konsisten untuk berbagai panjang mnemonik, dengan VmPeak (puncak ukuran memori virtual) yang tetap konstan pada 3148. Total penggunaan memori juga konsisten pada 3088 KB untuk semua operasi dan panjang mnemonik, sementara RSS (*Resident Set Size*) bervariasi sedikit antara 2120 KB hingga 2236 KB. Hal ini menunjukkan bahwa algoritma Speck memiliki *overhead* memori yang konsisten dan efisien, terlepas dari ukuran *input*, menjadikannya pilihan yang baik untuk implementasi pada perangkat dengan sumber daya terbatas.

### 4.5. Pengujian Keamanan

Pengujian keamanan dilakukan dengan menggunakan metode *brute force* (*ciphertext only attack*). Pengujian dilakukan menggunakan Google Cloud Compute Engine tipe e2-highcpu-16 dengan

16 vCPU dan 16 GB RAM. Proses ini dilakukan dengan menggunakan 16 thread.

Selama periode pengujian satu jam, program berhasil untuk mencoba sebanyak 7.985.600.000 (hampir 8 miliar) kombinasi kunci. Ruang kunci dibuat secara berurutan mulai dari nilai terendah hingga tertinggi dalam ruang kunci 256-bit. Dengan kemajuan seperti ini, diperkirakan masih dibutuhkan waktu sekitar $1.4 * 10^{67}$ tahun.

## 5. KESIMPULAN

Berdasarkan hasil dan pembahasan yang telah diuraikan, dapat disimpulkan bahwa implementasi algoritma enkripsi Speck untuk pengamanan frasa mnemonik pada *cryptocurrency wallet* telah berhasil dilakukan dengan efektif. Pengujian fungsional menunjukkan kesesuaian implementasi dengan spesifikasi yang ditetapkan, pengujian kinerja menunjukkan kinerja yang sangat baik dalam kecepatan eksekusi dan penggunaan memori, sementara analisis properti statistik memberikan hasil yang menjanjikan. Pengujian kinerja menunjukkan waktu rata-rata eksekusi di bawah 2 mikro detik dengan penggunaan memori di bawah 3 MB. *Avalanche effect* menunjukkan sensitivitas yang baik terhadap perubahan input, dengan nilai konsisten antara 0,47-0,53 untuk kasus yang signifikan. Pengujian entropi mendemonstrasikan peningkatan signifikan dari rata-rata 4 bit per *byte* pada *plaintext* menjadi sekitar 6 bit per *byte* pada *ciphertext*, mengindikasikan efektivitas algoritma dalam meningkatkan keacakan data. Uji *uniformity* memperlihatkan bahwa 95% hasil enkripsi mnemonik memiliki distribusi yang seragam, menegaskan kemampuan algoritma dalam menghasilkan *ciphertext* yang terdistribusi merata. Terakhir, pengujian keamanan melalui serangan *brute force* menggunakan 16 vCPU menunjukkan ketahanan yang baik dengan waktu pemecahan kunci mencapai sekitar $10^{67}$ tahun.

Implementasi algoritma Speck terbukti menjadi solusi yang layak dan efektif untuk mengamankan frasa mnemonik pada *cryptocurrency wallet*, menawarkan keseimbangan yang baik antara keamanan dan efisiensi. Penelitian ini memberikan kontribusi penting dalam pengembangan solusi keamanan yang efisien untuk cryptocurrency wallet, terutama pada perangkat dengan sumber daya terbatas. Hasil ini membuka jalan bagi pengembangan lebih lanjut dalam mengoptimalkan keamanan *cryptocurrency wallet* dan berpotensi mendorong adopsi yang lebih luas dari teknologi *blockchain* dan *cryptocurrency*. Dengan demikian, penelitian ini tidak hanya menyediakan solusi praktis untuk masalah keamanan saat ini, tetapi juga meletakkan dasar untuk inovasi lebih lanjut dalam keamanan aset digital.

## DAFTAR PUSTAKA


ARIAS-OLIVA, M., PELEGR IN-BORONDO, J. & MATIAS-CLAVERO, G., 2019. Variables influencing cryptocurrency use: a technology acceptance model in Spain. Frontiers in psychology, 10, p.438810.

BEAULIEU, R., SHORS, D., SMITH, J., TREATMAN-CLARK, S., WEEKS, B. & WINGERS, L., 2013. The SIMON and SPECK families of lightweight block ciphers (2013). National Security Agency.

BEAULIEU, R., SHORS, D., SMITH, J., TREATMAN-CLARK, S., WEEKS, B. & WINGERS, L., 2015. The SIMON and SPECK lightweight block ciphers. In: Proceedings of the 52nd annual design automation conference. pp.1–6.

CHAHOOKI, M.A.Z. & KJ, T.S., 2023. Cryptocurrencies investment framework using sentiment analysis of Twitter influencers. Indonesian Journal of Electrical Engineering and Computer Science, 30(2), p.7.

FANG, F., VENTRE, C., BASIOS, M., KANTHAN, L., MARTINEZ-REGO, D., WU, F. & LI, L., 2022. Cryptocurrency trading: a comprehensive survey. Financial Innovation, 8(1), p.13.

LESAVRE, L., VARIN, P., MELL, P., DAVIDSON, M. & SHOOK, J., 2019. A taxonomic approach to understanding emerging blockchain identity management systems. arXiv preprint arXiv:1908.00929.

PALATINUS, M. & RUSNAK, P., 2014. BIP 44: Multi-Account Hierarchy for Deterministic Wallets. Available at: <https://github.com/bitcoin/bips/blob/master/bip-0044.mediawiki> [Accessed 4 July 2024].

PALATINUS, M., RUSNAK, P., VOISINE, A. &BOWE, S., 2013. BIP 39: Mnemonic code for generating deterministic keys. Available at: <https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki> [Accessed 4 July 2024].

SANS, T., LIU, Z. & OH, K., 2023. A Decentralized Mnemonic Backup System for Non-custodial Cryptocurrency Wallets. pp.355–370. https://doi.org/10.1007/978-3-031-30122-3_22.

SHAH, A.F.M.S., KARABULUT, M.A., AKHTER, A.F.M.S., MUSTARI, N., PATHAN, A.-S.K., RABIE, K.M. & SHONGWE, T., 2023. On the vital aspects and characteristics of cryptocurrency—A survey. Ieee Access, 11, pp.9451–9468.

SLEEM, L. AND COUTURIER, R., 2021. Speck-R: An ultra light-weight cryptographic scheme for Internet of Things. Multimedia Tools and Applications, 80(11), pp.17067–17102.

UDDIN, M.S., MANNAN, M. & YOUSSEF, A., 2021. Horus: A security assessment framework for android crypto wallets. In: Security and Privacy in Communication Networks: 17th EAI International Conference, SecureComm 2021, Virtual Event, September 6–9, 2021, Proceedings, Part II 17. pp.120–139.

WUILLE, P., 2012. BIP 32: Hierarchical Deterministic Wallets. Available at: <https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki> [Accessed 4 July 2024].

YUSTIARINI, B.Y., DEWANTA, F. & NUHA, H.H., 2022. A Comparative Method for Securing Internet of Things (IoT) Devices: AES vs Simon-Speck Encryptions. In: 2022 1st International Conference on Information System and Information Technology, ICISIT 2022. Institute of Electrical and Electronics Engineers Inc. pp.392–396. https://doi.org/10.1109/ICISIT54091.2022.9872666.