Implementasi RC4 dalam Aplikasi Enkripsi Berkas pada Diska Lepas USB

Aufar Ramadhan 18221163
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
18221163@std.stei.itb.ac.id

***Abstrak*—Pengguna perangkat cerdas semakin sering melakukan pemindahan dan penyimpanan informasi antar pengguna. Proses transfer dan penyimpanan data juga sudah didukung layanan penyimpanan awan dengan kepraktisan dan keamanan yang baik. Untuk mengatasi masalah keamanan pemindahan dan penyimpanan data pada media penyimpanan data fisik khususnya diska lepas USB, perlu dibuat aplikasi enkripsi berkas yang ringan, portabel, dan mendukung sistem operasi komputer umum. Algoritma RC4 dengan optimasi keamanan dan bahasa pemrograman Python dengan rangka kerja Flet digunakan untuk membangun aplikasi ini pada sistem operasi Windows dan MacOS. Aplikasi yang dibuat berhasil menjalankan fungsinya dengan baik tapi masih bisa dioptimasi lebih lanjut khususnya pada aspek keringkasan dan portabilitas.**

Kata kunci—kriptografi; diska lepas; RC4; Flet; aplikasi portabel;

I. PENDAHULUAN

Seiring perkembangan teknologi, pengguna perangkat cerdas semakin sering melakukan pemindahan dan penyimpanan informasi antarpengguna. Banyak cara untuk melakukan hal tersebut, khususnya dengan semakin terjangkaunya akses internet yang cepat. Proses transfer informasi yang awalnya dilakukan menggunakan media berbasis magnet seperti disket (*floppy disc*) beralih ke media lain ang lebih cepat dan praktis, mulai dari media berbasis optik seperti cakram padat (*compact disc*), media berbasis memori kilat (*flash memory*) seperti diska lepas (*portable flash drive*), hingga melalui jaringan internet.

Pemindahan dan penyimpanan informasi melalui jaringan internet memang sudah menjadi standar dengan keamanan, kepraktisan, dan murahnya biaya yang diperlukan. Akan tetapi, akses internet masih belum sebaik dan secepat yang diharapkan [2] dan masih banyak kasus yang memerlukan transfer cepat secara luring.

Untuk menyelesaikan masalah di atas, pengguna perangkat cerdas dapat kembali menggunakan media transfer fisik seperti diska lepas. Diska lepas saat ini memiliki kecepatan transfer yang tinggi didukung teknologi USB 3.0 dan sangat praktis dibawa ke manapun dengan harga yang cukup terjangkau, seperti [3]. Akan tetapi, diska lepas yang terjangkau tidak memiliki mekanisme pengamanan data sehingga keamanan data berada dalam ancaman, khususnya ketika terjadi kehilangan diska lepas yang berisi informasi penting.

Agar data pada diska lepas dapat diamankan dari akses oknum yang tidak bertanggung jawab, sebuah metode enkripsi yang ringan tapi cukup aman perlu diimplementasikan dalam bentuk alat atau aplikasi pengenkripsi berkas yang bersifat portabel dan mudah digunakan.

Aplikasi yang memenuhi kebutuhan di atas dapat dengan mudah dibangun menggunakan rangka kerja (*framework*) Flet yang berbasis bahasa pemrograman Python. Bahasa pemrograman C juga dapat digunakan untuk membuat beberapa optimasi terhadap program yang dibuat. RC4 dipilih sebagai algoritma enkripsi yang diimplementasikan karena memiliki performa yang cukup baik.

II. LANDASAN TEORI

A. Penyimpanan Data Komputer

Data pada komputer disimpan dalam bentuk bit-bit pada berbagai macam media. Terdapat tiga kategori media fisik penyimpanan data digital modern berdasarkan metode penyimpanannya sebagai berikut.

1) Magnetis

Media penyimpanan magnetis seperti *hard disk* dan disket menyimpan data digital pada medium magnetis melalui pembubuhan pola magnetis. [4]

2) Optik

Media penyimpanan optik seperti cakram padat menyimpan data digital melalui pembubuhan sinar laser daya rendah pada sebuah piringan. Media penyimpanan optik menyimpan data yang lebih banyak daripada media penyimpanan magnetis. [5]

3) Memori Kilat

Memori kilat adalah media penyimpanan data digital yang menyimpan data pada sel-sel memori yang terbuat dari transistor-transistor. Memori kilat hemat ruang dan memiliki kapasitas tinggi tapi memiliki batas penggunaan sebelum tidak dapat ditulis kembali. [6]

B. Diska Lepas USB

Diska lepas USB atau biasa disebut *flash disk* adalah media penyimpanan data digital yang menggabungkan memori kilat yang hemat tempat dengan antarmuka USB (*Universal Serial Bus*). Diska lepas USB dapat dilepas-pasang, ditulis ulang, dan

dibawa dengan mudah karena ukurannya yang kecil dan antarmuka USB yang fleksibel. [7]

Harga jual diska lepas USB semakin murah seiring waktu dengan kapasitas penyimpanan mencapai satuan Terabyte. Sayangnya, fitur enkripsi diska lepas USB masih jarang ditemukan. Seluruh diska lepas USB dengan fitur enkripsi masih memiliki harga yang tinggi.

C. *Kriptografi*

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan [8][9]. Kriptografi dapat menjamin empat aspek dari keamanan pesan:

1) Kerahasiaan

Isi pesan tidak dapat diketahui oleh pihak yang tidak berwenang.

2) Keutuhan dan keaslian isi pesan

Isi pesan dapat dijamin keutuhan dan integritasnya.

3) Keaslian pengirim dan penerima pesan

Pengirim dan penerima pesan dapat dibuktikan dengan benar. Pihak lain tidak dapat menyerupai pihak pengirim pesan yang sebenarnya.

4) Kenirsangkalan

Pengirim pesan tidak dapat menyangkal telah mengirim pesan yang ia kirim.

Berdasarkan objek kriptografi dan zaman pengembangannya, kriptografi terbagi menjadi kriptografi klasik dan modern. Kriptografi klasik mengenkripsi huruf dan angka menggunakan media tradisional seperti kertas dan pena. Contoh kriptografi klasik adalah sandi Caesar dan Vigenere. Kriptografi modern melalukan enkripsi dan dekripsi pesan dalam bentuk digital dan dilakukan oleh komputer. Contoh algoritma kriptografi modern adalah RC4, DSA, AES, RSA, dan SHA-3.

Algoritma kriptografi juga dapat diklasifikasikan berdasarkan kesimetrisan kuncinya. Pada algoritma kriptografi kunci simetris, kunci yang digunakan untuk dekripsi dan enkripsi sama. Pada algoritma kriptografi kunci nirsimetri atau umum disebut kriptografi kunci publik, kunci yang digunakan untuk enkripsi dan dekripsi tidak sama.

Algoritma kriptografi dapat juga dibagi berdasarkan satuan unit yang diproses sekaligus. Terdapat algoritma sandi alir dan sandi blok. Algoritma sandi alir beroperasi pada unit (huruf, karakter, bit, atau byte) tunggal. Algoritma sandi blok beroperasi pada sebuah blok yang terdiri dari beberapa unit sekaligus.

Selain ilmu untuk menjaga keamanan pesan, terdapat juga ilmu memecahkan kerahasiaan pesan. Ilmu ini disebut kriptanalisis. Perkembangan kriptanalisis mendorong ilmu kriptografi lebih jauh untuk menjaga keamanan pesan.

D. *RC4*

RC4 adalah algoritma sandi alir yang paling populer. RC4 dibuat oleh Ronald (Ron) Rivest dari Laboratorium RSA pada 1987 . RC4 digunakan dalam beberapa sistem keamanan lawas seperti SSL (Secure Socket Layer), WEP (Wired Equivalent Privacy), dan WPA (Wi-Fi Protocol Access). [10]

Kunci rahasia RC4 memiliki panjang maksimal 256 byte. Jika diasumsikan 1 byte setara 1 karakter, panjang kunci rahasia RC4 maksimal 256 karakter. RC4 memroses data dalam satuan byte.

Berikut dua subproses yang menjadi dua tahap pemrosesan pada algoritma RC4.

1) Key-Scheduling Algorithm (KSA)

Pada tahap ini, dilakukan pembangkitan kunci alir sepanjang 256 byte melalui proses permutasi terhadap himpunan (*array*) byte.

2) Pseudo-random Generation Algorithm (PRGA)

Pada tahap ini, kunci alir yang telah dibangkitkan digunakan untuk melakukan enkripsi maupun dekripsi pada byte pesan masukan. Permutasi terhadap kunci alir terus dilakukan selama proses enkripsi/dekripsi berlangsung. Enkripsi maupun dekripsi dilakukan dengan langkah-langkah yang sama karena operasi XOR bersifat bolak-balik.

Algoritma RC4 telah berhasil dipecahkan oleh metode kriptanalisis namun masih cukup aman digunakan dengan beberapa optimasi. [11]

E. *Bahasa Pemrograman Python*

Python adalah bahasa pemrograman tingkat tinggi multiparadigma yang dapat digunakan untuk berbagai keperluan umum. Tipe pada Python bersifat dinamis dan memiliki mekanisme pengumpulan sampah. Salah dua ciri khas Python adalah penekanan penggunaan indentasi kode dan kumpulan modul *library* yang sangat banyak, baik yang bersifat bawaan maupun tambahan.

Bahasa Python sangat sering digunakan dalam pemelajaran mesin (*machine learning*) dan menjadi salah satu bahas pemrograman terpopuler [12]. Bahasa Python memiliki lebih dari 548000 modul tambahan (*packages*) pada PyPI (Python Package Index) [13]. Salah satu *package* yang membantu pengembangan aplikasi banyak *platform* adalah Flet.

Flet adalah *package* yang membantu membuat aplikasi web, komputer (Linux, MacOS, dan Windows), dan *mobile* (Android dan iOS) dengan cepat menggunakan Python. Flet membangun aplikasi dengan basis Flutter dan dapat disambungkan dengan *platform* pengembangan aplikasi *mobile* dan komputer seperti Xcode, Android SDK, dan Visual Studio untuk melakukan kompilasi aplikasi.

F. *Bahasa Pemrograman C*

C adalah bahasa pemrograman umum tingkat menengah yang bersifat imperatif-prosedural, bertipe statis, dan tergolong bebas secara posisi pengetikan kode. Bahasa C sangat menyesuaikan kondisi CPU dan sangat cepat sehingga banyak digunakan dalam program-program tingkat rendah dan sistem operasi. Akan tetapi, kesulitan optimasi keamanan dan keterbatasan fitur bahasa ini menyebabkan C jarang digunakan dalam membuat aplikasi. [14]

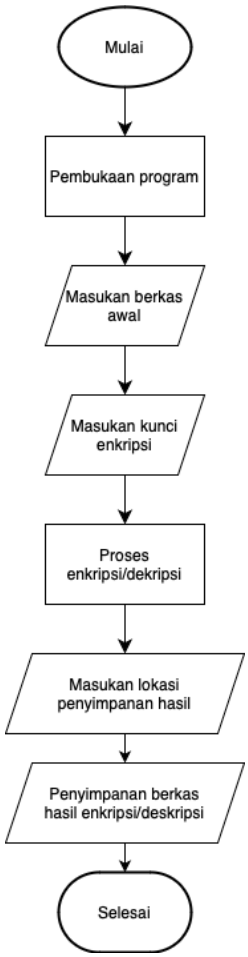
III. Rancangan dan Implementasi

Aplikasi pengenkripsi berkas ini hampir sepenuhnya diimplementasikan dengan bahasa pemrograman Python, memanfaatkan Flet sebagai rangka kerja pembuatan aplikasi GUI. Untuk memudahkan peluncuran program pada sistem operasi Windows, digunakan program pembantu peluncur program yang diimplementasikan dengan bahasa pemrograman C.

A. Rancangan Aplikasi

Aplikasi pengenkripsi berkas ini dirancang agar mudah dan cepat digunakan. Pengguna dapat langsung memasukkan kunci enkripsi dan berkas yang hendak dienkripsi atau didekripsi setelah aplikasi terbuka. Berkas hasil pemrosesan akan disimpan langsung di tempat aplikasi dibuka, umumnya di direktori dasar sebuah diska lepas.

Berikut diagram alur penggunaan aplikasi enkripsi berkas berbasis RC4.



Gbr. 1. Diagram alur aplikasi pengenkripsi berkas

Untuk membedakan berkas yang sudah atau belum dienkripsi, program akan mendeteksi format ekstensi berkas. Berkas dengan format ekstensi .enc diasumsikan sebagai berkas yang terenkripsi.

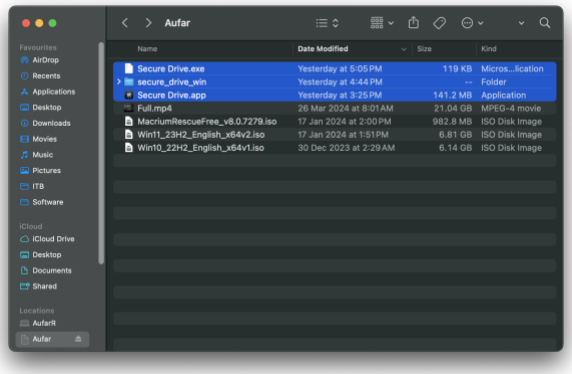
Untuk meningkatkan keamanan RC4, 768 byte awal kunci dibuang sebelum melakukan proses enkripsi pesan. [15]

B. Implementasi

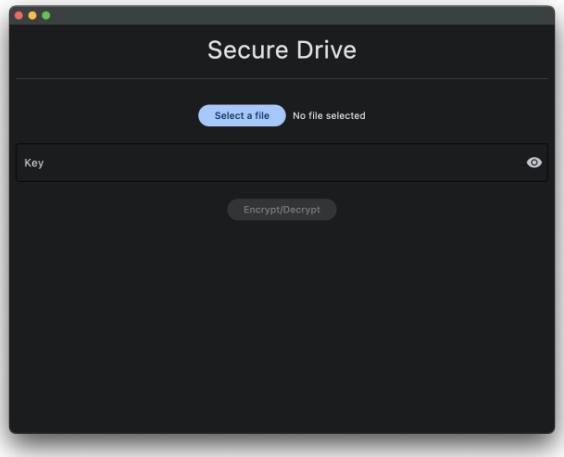
Implementasi aplikasi disusun sesuai struktur projek Flet sederhana. Berikut struktur projek implementasi aplikasi enkripsi berkas berbasis RC4.

```
secure-drive/
|_assets/
||_icon.png
|_.gitignore
|_LICENSE
|_README.md
|_main.py
|_rc4.py
|_runner.c
```

Kode program dikompilasi menjadi aplikasi Windows dan MacOS yang dapat langsung disimpan di sebuah direktori tanpa melakukan instalasi. Berikut contoh hasil program yang telah dikompilasi dan diletakkan di sebuah diska lepas. Repositori projek ini dapat diakses di [16].



Gbr. 2. Berkas-berkas *executable* hasil kompilasi pada sebuah diska lepas



Gbr. 3. Tampilan antarmuka aplikasi saat dibuka

IV. PENGUJIAN

A. *Rancangan Pengujian*

Pengujian dilakukan untuk berbagai kondisi masukan awal. Berikut daftar kasus pengujian aplikasi.

TABEL I. DAFTAR KASUS PENGUJIAN

No	Kondisi awal	Ekspektasi keluaran
1	Tidak ada berkas maupun kunci yang dimasukkan	Tombol enkripsi/dekripsi tidak dapat ditekan
2	Tidak ada berkas yang dipilih, kunci telah dimasukkan	Tombol enkripsi/dekripsi tidak dapat ditekan
3	Ada berkas yang dipilih, masukan kunci dapat kosong atau terisi, tombol enkripsi/dekripsi ditekan, pengguna membatalkan pemilihan lokasi penyimpanan	Tombol enkripsi/dekripsi dapat ditekan. Setelah pemilihan lokasi dibatalkan, muncul peringatan galat, kondisi masukan berkas tidak berubah
4	Ada berkas-tidak-terenkripsi yang dipilih, tidak ada kunci yang dimasukkan, tombol enkripsi ditekan, pengguna memilih lokasi penyimpanan	Tombol enkripsi dapat ditekan. Setelah konfirmasi, muncul peringatan galat, kondisi masukan berkas tidak berubah
5	Ada berkas-terenkripsi yang dipilih, tidak ada kunci yang dimasukkan, tombol dekripsi ditekan, pengguna memilih lokasi penyimpanan	Tombol dekripsi dapat ditekan. Setelah konfirmasi, muncul peringatan galat, kondisi masukan berkas tidak berubah
6	Ada berkas-tidak-terenkripsi yang dipilih, kunci telah dimasukkan, tombol enkripsi ditekan, pengguna memilih lokasi penyimpanan	Tombol enkripsi dapat ditekan. Setelah konfirmasi, berkas terenkripsi berhasil tersimpan, kondisi masukan berkas kembali kosong dan tombol enkripsi/dekripsi kembali nonaktif
7	Ada berkas-terenkripsi yang dipilih, kunci yang benar telah dimasukkan, tombol dekripsi ditekan, pengguna memilih lokasi penyimpanan	Tombol dekripsi dapat ditekan. Setelah konfirmasi, berkas terdekripsi berhasil tersimpan, kondisi masukan berkas kembali kosong dan tombol enkripsi/dekripsi kembali nonaktif, berkas terdekripsi sesuai dengan berkas uji
8	Ada berkas-terenkripsi yang dipilih, kunci yang salah telah dimasukkan, tombol dekripsi ditekan, pengguna memilih lokasi penyimpanan	Tombol dekripsi dapat ditekan. Setelah konfirmasi, berkas terdekripsi berhasil tersimpan, kondisi masukan berkas kembali kosong dan tombol enkripsi/dekripsi kembali nonaktif, berkas terdekripsi tidak sesuai dengan berkas uji

Berikut tangkapan layar berkas yang dipakai untuk melakukan pengujian.

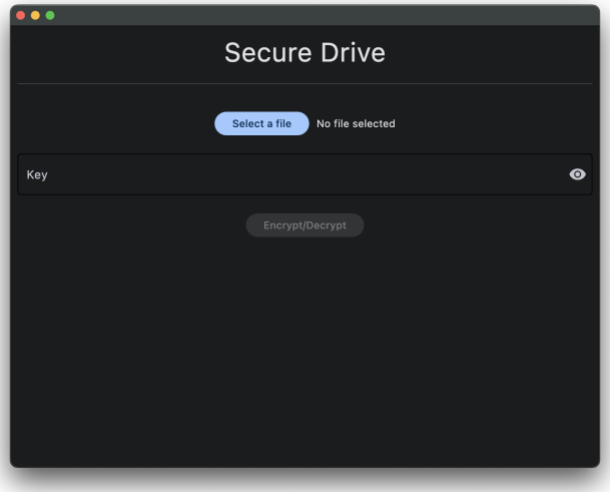
Gbr. 4. Berkas uji dengan MD5 260b61c025442909f4f7afe77d8a0c78

B. *Hasil Pengujian*

Berikut hasil pengujian aplikasi untuk seluruh kasus kondisi awal yang terdefinisi pada tabel 1.

1) Tidak ada berkas maupun kunci yang dimasukkan

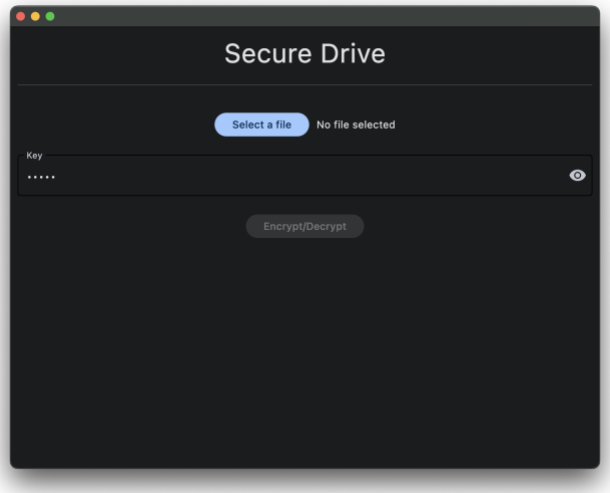
Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol enkripsi/dekripsi tidak dapat ditekan.



Gbr. 5. Kondisi akhir kasus uji 1

2) Tidak ada berkas yang dipilih, kunci telah dimasukkan

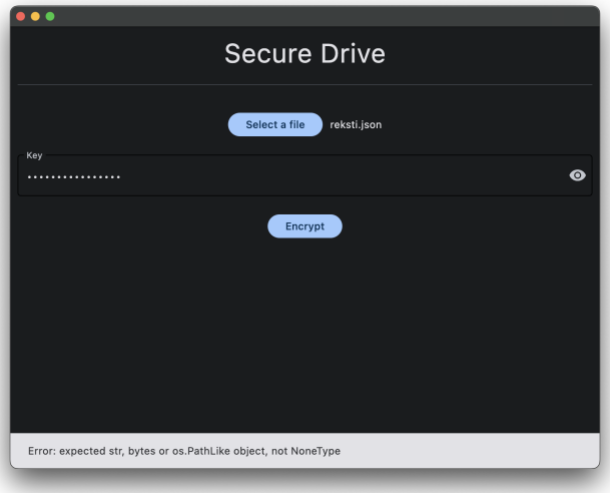
Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol enkripsi/dekripsi tidak dapat ditekan.



Gbr. 6. Kondisi akhir kasus uji 2

3) Ada berkas yang dipilih, masukan kunci dapat kosong atau terisi, tombol enkripsi/dekripsi ditekan, pengguna membatalkan pemilihan lokasi penyimpanan

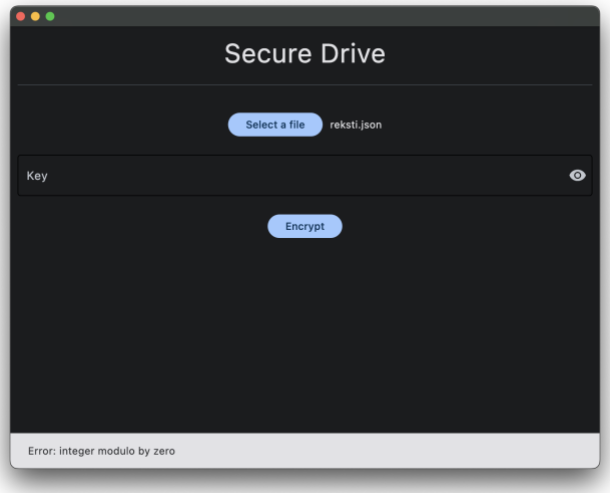
Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol enkripsi/dekripsi dapat ditekan. Setelah pemilihan lokasi dibatalkan, muncul peringatan galat, kondisi masukan berkas tidak berubah



Gbr. 7. Kondisi akhir kasus uji 3

4) Ada berkas-tidak-terenkripsi yang dipilih, tidak ada kunci yang dimasukkan, tombol enkripsi ditekan

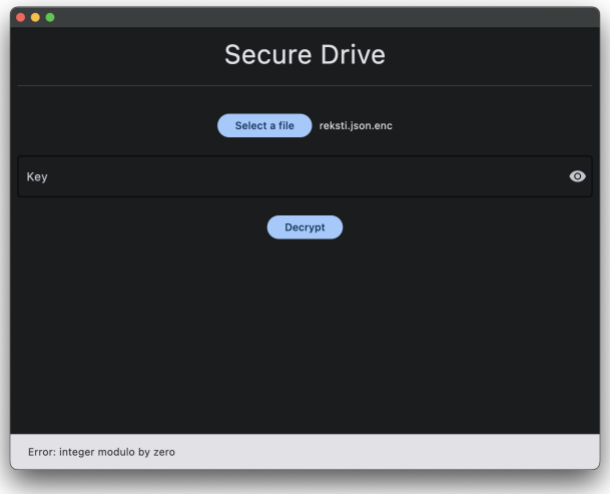
Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol enkripsi dapat ditekan. Setelah konfirmasi, muncul peringatan galat, kondisi masukan berkas tidak berubah.



Gbr. 8. Kondisi akhir kasus uji 4

5) Ada berkas-terenkripsi yang dipilih, tidak ada kunci yang dimasukkan, tombol dekripsi ditekan

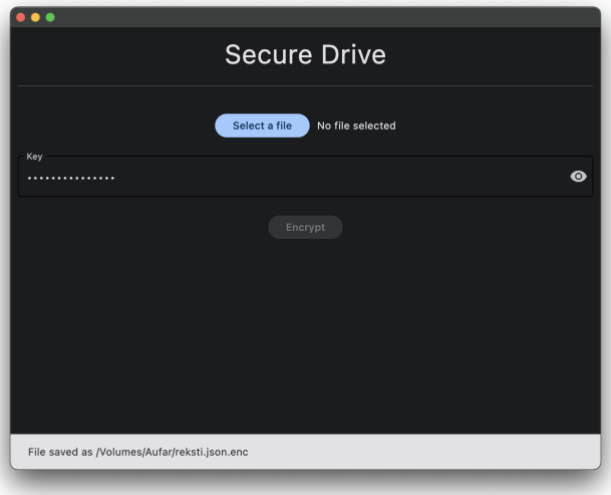
Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol dekripsi dapat ditekan. Setelah konfirmasi, muncul peringatan galat, kondisi masukan berkas tidak berubah.



Gbr. 9. Kondisi akhir kasus uji 5

6) Ada berkas-tidak-terenkripsi yang dipilih, kunci telah dimasukkan, tombol enkripsi ditekan

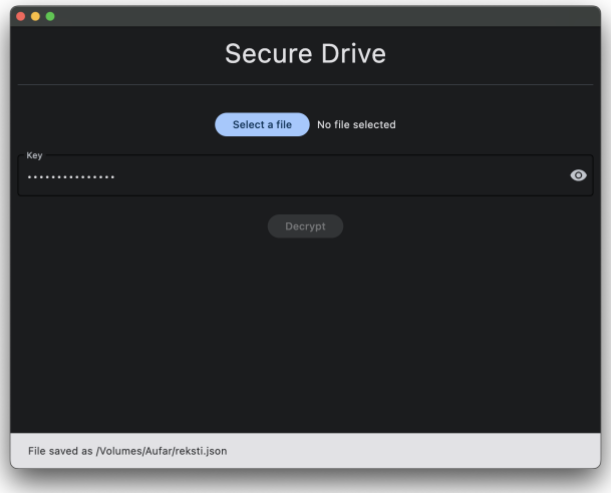
Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol enkripsi dapat ditekan. Setelah konfirmasi, berkas terenkripsi berhasil tersimpan, kondisi masukan berkas kembali kosong dan tombol enkripsi/dekripsi kembali nonaktif.



Gbr. 10. Kondisi akhir kasus uji 6

7) Ada berkas-terenkripsi yang dipilih, kunci yang benar telah dimasukkan, tombol dekripsi ditekan

Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol dekripsi dapat ditekan. Setelah konfirmasi, berkas terdekripsi berhasil tersimpan, kondisi masukan berkas kembali kosong dan tombol enkripsi/dekripsi kembali nonaktif, berkas terdekripsi dapat dibuka dan sesuai dengan berkas uji.

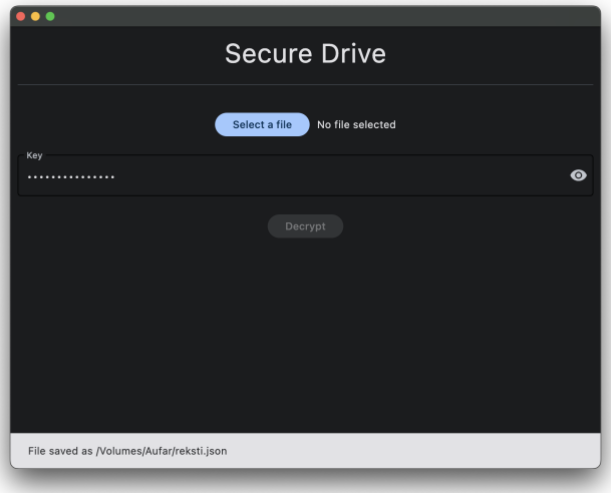


Gbr. 11. Kondisi akhir aplikasi pada kasus uji 7

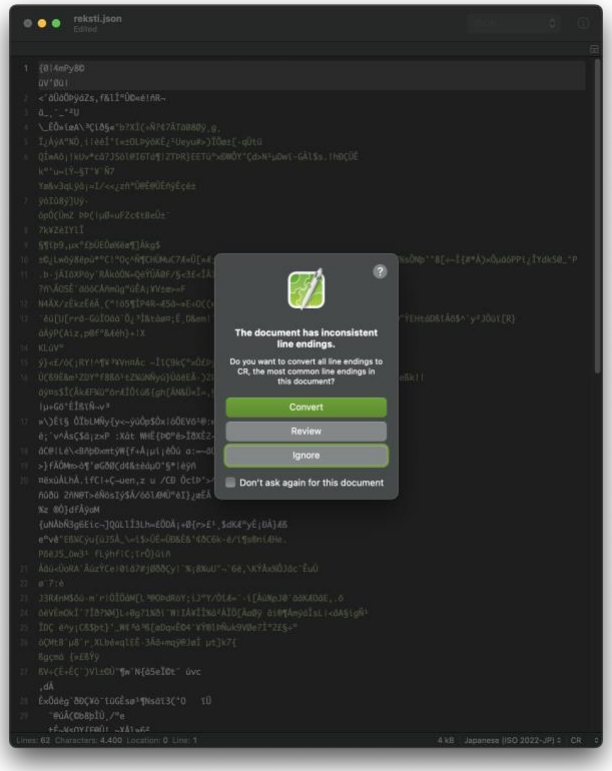
Gbr. 12. Kondisi berkas terdekripsi dengan MD5 260b61c025442909f4f7afe77d8a0c78

8) Ada berkas-terenkripsi yang dipilih, kunci yang salah telah dimasukkan, tombol dekripsi ditekan

Pada kasus ini, kondisi akhir aplikasi sesuai dengan ekspektasi. Tombol dekripsi dapat ditekan. Setelah konfirmasi, berkas terdekripsi berhasil tersimpan, kondisi masukan berkas kembali kosong dan tombol enkripsi/dekripsi kembali nonaktif, namun berkas terdekripsi tidak sesuai dengan berkas uji.



Gbr. 13. Kondisi akhir aplikasi pada kasus uji 8



Gbr. 14. Kondisi berkas terdekripsi dengan MD5 0545f4bcb27089cd49405ea90049140e

V. PENUTUP

Keamanan informasi pada diska lepas USB dapat ditingkatkan melalui aplikasi enkripsi berkas yang ringan dan portabel berbasis Flet dan algoritma RC4. Sesuai hasil pengujian, hasil implementasi aplikasi berhasil melakukan enkripsi dan dekripsi berkas pada diska lepas USB dengan kunci yang sesuai.

Untuk pengembangan lebih lanjut, diperlukan optimasi ukuran aplikasi agar semakin ringkas dan portabel. Pembangunan aplikasi secara *native* dapat menjadi opsi walaupun lebih lama dan kompleks. Eksplorasi penggunaan algoritma lainnya yang lebih aman dan lebih cepat juga dapat dilakukan untuk meningkatkan keandalan proses enkripsi dan dekripsi berkas.

VI. TAUTAN VIDEO YOUTUBE

Video pemaparan hasil penelitian ini dapat diakses di tautan https://youtu.be/b5uneJT8txE

REFERENSI

[1] S. Obrutsky, "(PDF) Cloud Storage: Advantages, Disadvantages and Enterprise Solutions for Business," *ResearchGate*, Jul. 2016. https://www.researchgate.net/publication/305508410_Cloud_Storage_Advantages_Disadvantages_and_Enterprise_Solutions_for_Business

[2] F. Amanta, "Unpacking Indonesia's Digital Accessibility," *Center for Indonesia Policy Studies*, Jul. 15, 2022. https://www.cips-indonesia.org/post/unpacking-indonesia-s-digital-accessibility?lang=id

[3] Lexar, "Lexar JumpDrive S47 USB 3.1 Flash Drive," *Lexar*, 2024. https://www.lexar.com/product/lexar-jumpdrive-s47-usb-3-1-flash-drive/

[4] TechTarget, "magnetic storage," *TechTarget*, Aug. 2014. https://www.techtarget.com/whatis/definition/magnetic-storage

[5] Britannica, "optical storage," *Britannica*, Jul. 09, 2020. https://www.britannica.com/technology/optical-storage/

[6] A. Bhosle, "What is Flash Memory?," *GeeksforGeeks*, Jul. 16, 2023. https://www.geeksforgeeks.org/what-is-flash-memory/

[7] M. Buchanan, "Object of Interest: The Flash Drive," *The New Yorker*, Jun. 14, 2013. https://www.newyorker.com/tech/annals-of-technology/object-of-interest-the-flash-drive

[8] B. Schneier, *Applied cryptography, second edition : protocols, algorithms, and source code in C*. Indianapolis, In: Wiley, 2015.

[9] R. Munir, "Pengantar Kriptografi STI," in *II4031 Kriptografi dan Koding 2023/2024*, 2024

[10] R. Munir, "Stream Cipher," in *II4031 Kriptografi dan Koding 2023/2024*, 2024

[11] M. Green, "Attack of the week: RC4 is kind of broken in TLS," *A Few Thoughts on Cryptographic Engineering*, Mar. 12, 2013. https://blog.cryptographyengineering.com/2013/03/12/attack-of-week-rc4-is-kind-of-broken-in/

[12] Stack Overflow, "2022 Developer Survey," *Stack Overflow Annual Developer Survey*, 2022. https://survey.stackoverflow.co/2022/

[13] Python Software Foundation, "PyPI," *The Python Package Index*, 2024. https://pypi.org

[14] B. W. Kernighan and D. M. Ritchie, *The C programming language*. Englewood Cliffs, Nj Prentice-Hall, 1991.

[15] D. Hopwood, "Standard Cryptographic Algorithm Naming," *Zetnet*, Oct. 22, 2002. http://www.users.zetnet.co.uk/hopwood/crypto/scan/cs.html

[16] A. Ramadhan, "AufarR/secure-drive," *GitHub*, Jun. 2024. https://github.com/AufarR/secure-drive

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024
Ttd

Aufar Ramadhan 18221163