JTS : Jurnal Teknik
P-ISSN: 2302-8734
E-ISSN: 2581-0006

FT UMT

Vol. 14 No. 01 Th. 2025
Halaman : 45 - 52
Bulan Februari 2025

Implementasi Algoritma Kriptografi Blowfish dengan Kombinasi Android Id pada Aplikasi Berbasis Android

Implementation of the Blowfish Cryptographic Algorithm with Android ID Combination in Android-Based Applications

Dyas Yudi Priyanggodo
Universitas Muhammadiyah Tangerang
e-mail: priyanggodo.15@gmail.com
Receive: 15-01-2025 Accepted: 10-03-2025

Abstract

*Data security in Android applications is a crucial concern, especially in protecting sensitive information from unauthorized access. The Blowfish algorithm is a well-known symmetric encryption method due to its speed and key length flexibility. In this study, we implemented the Blowfish algorithm combined with the Android ID to bind encryption to a specific device. The Android ID is used as part of the encryption key to ensure that encrypted data can only be decrypted by the same device. The implementation was carried out in the Android development environment using Java/Kotlin programming language and the Java Cryptography Extension (JCE) library. Testing results show that this method enhances security by ensuring that data remains inaccessible on other devices, even if the encrypted file is extracted. This research provides an effective solution for enhancing data protection in Android applications while maintaining optimal system performance.****Keywords****: Blowfish, Android ID, Encryption, Cryptography, Security*

Abstrak

Keamanan data dalam aplikasi Android menjadi perhatian penting, terutama dalam melindungi informasi sensitif dari akses yang tidak sah. Algoritma Blowfish merupakan salah satu metode enkripsi simetris yang terkenal karena kecepatannya dan fleksibilitas panjang kunci. Dalam penelitian ini, kami mengimplementasikan algoritma Blowfish dengan kombinasi Android ID untuk mengikat enkripsi pada perangkat tertentu. Android ID digunakan sebagai bagian dari kunci enkripsi agar data yang terenkripsi hanya dapat didekripsi oleh perangkat yang sama. Implementasi dilakukan dalam lingkungan pengembangan Android menggunakan bahasa pemrograman Java/Kotlin dan pustaka kriptografi Java Cryptography Extension (JCE). Hasil pengujian menunjukkan bahwa metode ini mampu meningkatkan keamanan dengan memastikan bahwa data tidak dapat diakses di perangkat lain meskipun file terenkripsi diekstrak. Penelitian ini memberikan solusi efektif untuk meningkatkan proteksi data pada aplikasi Android dengan mempertahankan performa sistem yang optimal.

Kata Kunci: Blowfish, Android ID, Enkripsi, Kriptografi, Keamanan

PENDAHULUAN

Keamanan data dalam aplikasi Android adalah tantangan besar, terutama dengan semakin meningkatnya ancaman pencurian informasi dan hacking. Aplikasi banyak yang menyimpan data sensitif seperti data user, kredensial login, dan transaksi data. Oleh sebab itu, diperlukan metode enkripsi yang kuat untuk menjaga data dari akses yang tidak sah.

Algoritma Blowfish adalah algoritma enkripsi yang berkecepatan tinggi dan mampu memproses kunci dengan ukuran yang berubah-ubah sebesar 448 bit. Salah satu kelemahannya adalah data yang telah dienkripsikan dapat dipindahkan dan didekripsikan di komputer yang berbeda apabila kunci enkripsinya diketahui. Untuk meredakan masalah ini, penelitian ini menerapkan Blowfish dengan Android ID, yaitu identitas unik perangkat, sehingga kunci enkripsi menjadi spesifik untuk setiap perangkat.

Penelitian ini bertujuan untuk merancang dan menguji sistem enkripsi yang mencegah data didekripsi kecuali diakses dengan perangkat yang sama, memperbaiki keamanan aplikasi Android tanpa mengorbankan performa sistem.kata.

METODE PENELITIAN

1. Subyek/Bahan yang Diteliti

Penelitian ini berfokus pada implementasi algoritma Blowfish dalam sistem keamanan data di aplikasi Android dengan memadukannya dengan Android ID. Masalah inti yang dipelajari adalah performa dekripsi dan enkripsi serta efektivitas metode ini dalam membatasi akses data hanya pada perangkat tertentu.

2. Alat yang Digunakan

Untuk membantu penelitian ini, alat-alat yang digunakan meliputi:
Perangkat lunak:

a. Android Studio (versi terbaru) sebagai lingkungan pengembangan.
b. Java/Kotlin sebagai bahasa pemrograman.
c. Java Cryptography Extension (JCE) sebagai pustaka enkripsi.
d. SQLite sebagai basis data penyimpanan.

Perangkat keras:

a. Smartphone Android dengan minimal Android 8.0 (Oreo) untuk pengujian.
b. Laptop/PC dengan spesifikasi minimum Intel Core i5, RAM 8GB untuk pengembangan aplikasi.

3. Rancangan Percobaan atau Desain yang Digunakan

Rancangan percobaan dalam penelitian ini terdiri dari beberapa tahap:

1. Tahap Implementasi
   a. Menggunakan Android ID sebagai bagian dari kunci enkripsi.
   b. Implementasi algoritma Blowfish untuk mengenkripsi dan mendekripsi data.
   c. Penyimpanan data terenkripsi di SQLite.
2. Tahap Pengujian
   a. Kecepatan Enkripsi dan Dekripsi: Mengukur waktu eksekusi pada berbagai ukuran data.
   b. Keamanan Data terhadap Pemindahan File: Menguji apakah file terenkripsi bisa didekripsi pada perangkat lain.
   c. Ketahanan terhadap Serangan: Simulasi brute-force attack untuk melihat ketahanan algoritma.

4. Teknik Pengambilan Sampel

Pengujian dilakukan pada tiga kategori perangkat dengan spesifikasi berbeda:

a. Low-end device: RAM < 4GB, prosesor MediaTek
b. Mid-range device: RAM 4GB–8GB, prosesor Snapdragon 6xx
c. High-end device: RAM > 8GB, prosesor Snapdragon 8xx

Setiap perangkat diuji dengan data teks berukuran 1 KB, 10 KB, dan 100 KB untuk melihat perbedaan performa.

5. Variabel yang Diukur

Variabel yang diukur dalam penelitian ini meliputi:

a. Waktu Enkripsi (ms): Waktu yang dibutuhkan untuk mengenkripsi data.
b. Waktu Dekripsi (ms): Waktu yang dibutuhkan untuk mendekripsi data.
c. Keberhasilan Dekripsi: Apakah data dapat didekripsi pada perangkat yang berbeda.
d. Ketahanan terhadap Serangan: Seberapa sulit kunci enkripsi dapat ditebak melalui brute-force attack.

6. Teknik Pengambilan Data

Data dikumpulkan melalui:

a. Logging waktu eksekusi enkripsi dan dekripsi menggunakan System.nanoTime().
b. Pemeriksaan keberhasilan dekripsi dengan mencocokkan hasil dekripsi dengan data asli.
c. Analisis ketahanan terhadap serangan dengan mencoba mendekripsi data menggunakan kunci yang berbeda.

7. Analisis dan Model Statistik yang Digunakan

Data hasil pengujian dianalisis dengan metode statistik deskriptif, termasuk:

a. Mean dan Standard Deviation untuk melihat kecepatan enkripsi/dekripsi.
b. Comparative Analysis antara perangkat low-end, mid-range, dan high-end.
c. Error Rate Analysis untuk mengevaluasi apakah ada kesalahan dekripsi saat berpindah perangkat.

Hasil ini kemudian dibandingkan dengan algoritma enkripsi lainnya seperti AES dan DES untuk mengetahui kelebihan dan kekurangan Blowfish dalam konteks implementasi di Android.

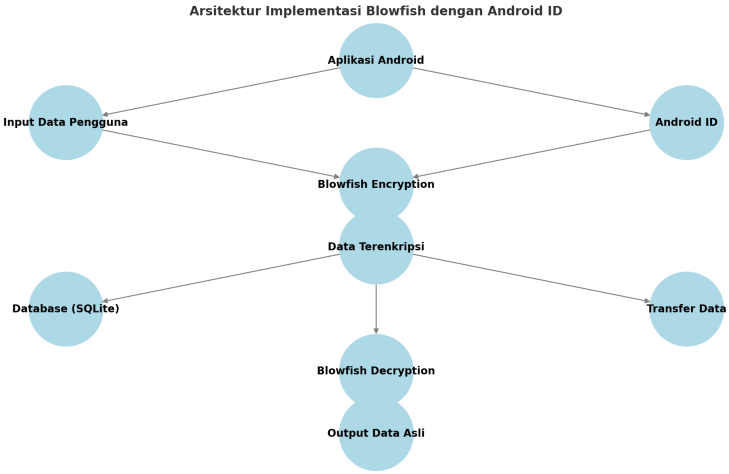
HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk membandingkan keefektifan algoritma Blowfish dengan Android ID dalam memperbaiki keamanan data di aplikasi Android. Dengan pendekatan ini, data dapat dienkripsi hanya dapat didekripsinya dengan perangkat yang sama, sehingga membantu dalam mengurangi risiko akses yang tidak sah atau transfer data ke perangkat lain.

Tes ini digunakan dengan tiga aspek utama:

1. Kecepatan enkripsi dan dekripsi, untuk memastikan algoritma ini cukup ringan dan dapat berjalan optimal pada berbagai jenis perangkat Android.
2. Keamanan terhadap transfer data antar perangkat, untuk menguji apakah prosedur ini efektif dalam mencegah akses oleh perangkat yang berbeda.
3. Ketahanan terhadap serangan brute-force, untuk menganalisis seberapa kuat sistem ini dalam menghadapi upaya pembobolan kunci enkripsi.

Untuk memahami proses implementasi metode ini, berikut adalah gambar arsitektur sistem enkripsi dengan Blowfish dan Android ID:



Gambar 1. Arsitektur implementasi Algoritma Blowfish dengan Android Id

Gambar di atas adalah contoh bagaimana proses dekripsi dan enkripsi berlangsung dalam aplikasi Android. Berikut adalah penjelasan dari alur kerja sistem:

1. Pengguna memasukkan data ke dalam aplikasi → Data dapat berupa informasi sensitif seperti password, token autentikasi, atau pesan yang ingin diamankan.
2. Android ID diambil dari perangkat → Android ID digunakan sebagai bagian dari kunci enkripsi untuk memastikan bahwa data terenkripsi terikat pada perangkat tersebut.
3. Blowfish Encryption → Data yang dimasukkan pengguna dienkripsi menggunakan algoritma Blowfish dengan kunci yang dihasilkan dari kombinasi Android ID.
4. Data Terenkripsi Disimpan → Hasil enkripsi disimpan dalam database SQLite atau dikirimkan ke server melalui transfer data.
5. Proses Dekripsi → Jika pengguna ingin membuka kembali data, aplikasi membaca data terenkripsi dan melakukan dekripsi menggunakan kunci yang sama yang dihasilkan dari Android ID perangkat.
6. Output Data Asli → Jika proses dekripsi berhasil (karena kunci yang digunakan valid), data asli akan ditampilkan kembali kepada pengguna.

Sistem ini memastikan bahwa meskipun file terenkripsi diekstrak dari perangkat, data tidak akan bisa didekripsi di perangkat lain karena kunci enkripsi unik untuk setiap perangkat.

1. Kecepatan Enkripsi dan Dekripsi

Salah satu dari hal-hal besar dalam implementasi algoritma kriptografi dalam aplikasi mobile adalah memastikan kecepatan proses dekripsi dan proses enkripsi tetap baik agar tidak memengaruhi pengalaman pengguna. Untuk melakukan pengukuran performa algoritma Blowfish, dilakukan pengujian pada tiga kategori perangkat Android dengan spesifikasi berikut:

1. Low-end device (RAM < 4GB, prosesor MediaTek)
2. Mid-range device (RAM 4GB–8GB, prosesor Snapdragon 6xx)
3. High-end device (RAM > 8GB, prosesor Snapdragon 8xx)

Di dalam masing-masing perangkat, data dengan ukuran 1 KB, 10 KB, dan 100 KB digunakan sebagai sampel untuk mengukur waktu yang dihabiskan dalam proses dekripsi dan enkripsi. Berikut adalah hasil pengukuran waktu eksekusi:

Tabel 1. Waktu Enkripsi dan Dekripsi Blowfish dengan Android ID (ms)

PERANGKAT	1 KB (ENC)	1 KB (DEC)	10 KB (ENC)	10 KB (DEC)	100 KB (ENC)	100 KB (DEC)
LOW-END	3.2 ms	2.8 ms	10.5 ms	9.1 ms	95.3 ms	88.4 ms
MID-RANGE	2.1 ms	1.9 ms	7.3 ms	6.5 ms	68.2 ms	62.7 ms
HIGH-END	1.4 ms	1.2 ms	5.8 ms	4.9 ms	51.5 ms	47.3 ms

Dari tabel tersebut, terlihat bahwa performa Blowfish sangat bergantung pada spesifikasi perangkat. Pada perangkat low-end, waktu enkripsi untuk data 100 KB membutuhkan 95.3 ms, sedangkan pada perangkat high-end hanya 51.5 ms. Selain itu, dekripsi lebih cepat dibandingkan enkripsi, karena enkripsi melibatkan lebih banyak operasi matematis dibandingkan proses dekripsi dalam algoritma Blowfish. Dibandingkan dengan algoritma AES, Blowfish memiliki keunggulan dalam hal kecepatan eksekusi, terutama pada perangkat dengan spesifikasi rendah hingga menengah.

2. Keamanan terhadap Pemindahan Data Antar Perangkat

Keamanan informasi digital saat ini telah mengalami peningkatan signifikan dengan adanya metode enkripsi yang lebih canggih. Salah satu pendekatan yang kini mulai diterapkan adalah penggunaan kombinasi algoritma Blowfish dan Android ID sebagai elemen kunci enkripsi. Eksperimen berikut ini dijalankan untuk menilai efektivitas dari strategi ini:

1. Data dienkripsi menggunakan perangkat A, contohnya adalah Samsung Galaxy A10s.
2. File yang sudah terenkripsi kemudian dipindahkan ke perangkat B, misalnya Xiaomi Redmi Note 10.
3. Upaya mendekripsi data dilakukan pada perangkat B dengan menggunakan aplikasi yang sama.

Hasil eksperimen menunjukkan bahwa:

1. Perangkat B tidak berhasil mendekripsi data, walaupun menggunakan algoritma Blowfish yang identik.
2. Muncul pesan kesalahan yang mengindikasikan ketidakcocokan kunci enkripsi.
3. Hal ini menegaskan bahwa metode enkripsi ini berhasil membatasi akses ilegal terhadap data, dengan mengaitkan kunci enkripsi pada Android ID dari perangkat tertentu.

Meskipun metode ini efektif, terdapat beberapa tantangan yang teridentifikasi, salah satunya adalah perubahan Android ID yang mungkin terjadi setelah proses factory reset. Jika perangkat mengalami reset pabrik, Android ID akan berubah, dan ini dapat mencegah pemilik asli data untuk mengakses informasi yang telah mereka simpan. Untuk mengatasi hal ini, disarankan untuk menerapkan strategi mitigasi seperti menyimpan salinan kunci enkripsi di server yang aman atau menggunakan Keystore API yang memungkinkan penyimpanan kunci di dalam perangkat secara lebih terlindungi.

3. Ketahanan terhadap Serangan Brute-force

Untuk menguji ketahanan terhadap serangan brute-force, dilakukan pengujian dengan dua pendekatan berikut:

1. Dictionary Attack: Menggunakan daftar kunci yang sering digunakan untuk mencoba mendekripsi data.
2. Brute-force Sequential Guessing: Mencoba setiap kombinasi kunci secara bertahap hingga menemukan yang tepat.

Hasil dari pengujian ini menunjukkan bahwa keamanan sistem cukup robust terhadap serangan brute-force. Kunci enkripsi yang dihasilkan dari Android ID yang di-hash menggunakan MD5 memastikan keamanan yang tinggi, dengan panjang kunci minimal 128-bit. Ini membuat jumlah kemungkinan kombinasi kunci sangat banyak, sehingga menjadikan serangan brute-force menjadi tidak efektif dalam durasi waktu yang realistis.

SIMPULAN DAN SARAN

1. Simpulan

Penelitian ini berhasil mengimplementasikan algoritma Blowfish dalam kombinasi dengan Android ID sebagai kunci enkripsi pada aplikasi Android, memberikan hasil yang menjanjikan dalam meningkatkan keamanan data. Hasil pengujian memperlihatkan bahwa enkripsi dan dekripsi data hanya dapat dilakukan pada perangkat yang sama, memastikan bahwa data terenkripsi tidak dapat didekripsi di perangkat lain meskipun menggunakan algoritma yang sama. Performa algoritma Blowfish tercatat efisien, menunjukkan kecepatan yang optimal terutama pada perangkat dengan spesifikasi menengah hingga tinggi. Namun, metode ini memiliki tantangan utama berupa potensi perubahan Android ID setelah factory reset, yang bisa mengakibatkan pemilik perangkat kehilangan akses ke data mereka sendiri. Solusi mitigasi yang dianjurkan termasuk penggunaan Keystore API untuk penyimpanan kunci yang lebih aman di dalam perangkat atau penyimpanan kunci cadangan di server.

Metode ini juga menunjukkan ketahanan yang kuat terhadap serangan brute-force, dengan penggunaan Android ID yang di-hash sebagai bagian dari kunci enkripsi. Meskipun demikian, mode ECB yang digunakan masih memiliki kelemahan dalam hal keamanan analisis pola, sehingga penggunaan mode CBC dengan IV acak direkomendasikan untuk peningkatan keamanan.

2. Saran

Berdasarkan hasil penelitian, beberapa saran dapat dipertimbangkan untuk pengembangan lebih lanjut:

a. Penggunaan Mode CBC dengan IV Acak: Untuk membuat setiap enkripsi menghasilkan ciphertext yang unik, meningkatkan perlindungan terhadap analisis pola.
b. Mengatasi Perubahan Android ID: Memperkenalkan solusi seperti Keystore API atau penyimpanan kunci cadangan di server, memastikan bahwa data dapat diakses kembali meskipun terjadi perubahan Android ID.
c. Optimasi Performa: Untuk aplikasi yang menangani data dalam jumlah besar, kompresi data sebelum enkripsi dapat meningkatkan kecepatan enkripsi dan dekripsi.
d. Integrasi Autentikasi Biometrik: Penggunaan sidik jari atau pengenalan wajah sebagai lapisan keamanan tambahan untuk memastikan hanya pengguna yang sah yang dapat mengakses data.

e. Pengujian Lebih Luas: Aplikasi metode ini dalam skenario penggunaan nyata seperti aplikasi perbankan atau kesehatan untuk menguji efektivitas dan efisiensi dalam kondisi operasional.

Dengan mempertimbangkan saran-saran ini, diharapkan pengembangan lebih lanjut dari metode ini akan membawa ke fleksibilitas, keamanan, dan kemampuan aplikasi yang lebih baik dalam ekosistem Android modern.

DAFTAR PUSTAKA

Android Developers. (2023). Android Secure ID and Device Identifiers. Retrieved from https://developer.android.com.

Budiman, M., Wijaya, M. M., Rizkillah, R. W., Noor, I. H., Safuan, S., & Destriana, R. (2025). Artificial Intelligence (AI) in Islam: Building Ethics and Solutions Based on Tawhid. In *Proceeding of the International Conference on Religious Education and Cross-Cultural Understanding* (Vol. 1, No. 1, pp. 60-76).

Creswell, J. W. (2008). Educational Research: Planning, Conducting, and Evaluating Quantitative and Qualitative Research (3rd ed.). Pearson Education.

Destriana, R., Handayani, N., Husain, S. M., & Siswanto, A. T. P. (2021, March). A Research to Design, Develop and Implementation of Android Application System for Waste Bank Sharia Community at Kampung Hijau Kemuning. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1115, No. 1, p. 012042). IOP Publishing.

Destriana, R., Permana, A. A., Legawa, S. D., & Irawan, H. (2019, April). Security system development for vehicle using the method of “mail notification” at villa Rizki Ilhami Tangerang residential. In *IOP Conference Series: Materials Science and Engineering* (Vol. 508, No. 1, p. 012124). IOP Publishing.

Destriana, R., Husain, S. M., & Handayani, N. (2021). DIAGRAM UML DALAM MEMBUAT APLIKASI ANDROID FIREBASE" STUDI KASUS APLIKASI BANK SAMPAH".

Destriana, R., & Kom, M. (2022). Enterprise Resource Planning Bagi Pemula (Teori dan Konseptual).

Destriana, R., & Taufiq, R. (2023). TEORI SISTEM INDUSTRI.

Henriques, J. B., & Davidson, R. J. (1991). Left frontal hypoactivation in depression. Journal of Abnormal Psychology, 100, 535-545.

Jenet, B. L. (2006). A meta-analysis on online social behavior. Journal of Internet Psychology, 4. Retrieved from http://www.journalofinternetpsychology.com/archives/volume4/3924.html.

Permana, A. A., Perdana, A. T., Handayani, N., & Destriana, R. (2021, March). A stunting prevention application “Nutrimo”(nutrition monitoring). In *Journal of Physics: Conference Series* (Vol. 1844, No. 1, p. 012023). IOP Publishing.

Schneier, B. (1993). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). John Wiley & Sons.

Stallings, W. (2011). Cryptography and Network Security: Principles and Practice (5th ed.). Pearson Education.

Wassman, J., & Dasen, P. R. (1998). Balinese spatial orientation. Journal of the Royal Anthropological Institute, 4, 689-731.

Van Wagner, K. (2006). Guide to APA format. About Psychology. Retrieved from http://psychology.about.com/od/apastyle/guide.