IMPLEMENTASI SISTEM *DEKRIPSI TOKEN* UNTUK MENCEGAH BERBAGI AKUN PADA APLIKASI GURU PINTAR DI SEKOLAH

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan

Gelar Sarjana Komputer (S.Kom) Program Studi Informatika

MUH. AL IQRAM MARZAH
105841105121

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025

IMPLEMENTASI SISTEM *DEKRIPSI TOKEN* UNTUK MENCEGAH BERBAGI AKUN PADA APLIKASI GURU PINTAR DI SEKOLAH

Diajukan sebagai Salah Satu Syarat untuk Mendapatkan
Gelar Sarjana Komputer (S.Kom) Program Studi Informatika

Disusun dan Diajukan Oleh :

MUH. AL IQRAM MARZAH
105841105121

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MUHAMMADIYAH MAKASSAR

2025

PENGESAHAN

Skripsi atas nama MUH. AL IQRAM MARZAH dengan nomor induk Mahasiswa 105841105121, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

Makassar, 15 Rabi'uLAwal 1447 H / 08 September 2025 M

1. Pengawas Umum
   a. Rektor Universitas Muhammadiyah Makassar
      Dr. Ir. H. Abd. Rakhim Nanda, ST.,MT.,IPU : .....
   b. Dekan Fakultas Teknik Universitas Hasanuddin
      Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T.,ASEAN.,Eng : .....
2. Penguji
   a. Ketua : Dr.Ir Zahir Zainuddin, M.Sc : .....
   b. Sekertaris : Titin Wahyuni, S.Pd., M.T. : .....
3. Anggota : 1. Desi Anggreani, S.Kom., M.T : .....
   2. Ir. Muhammad Faisal, S.SI.,M.T.,Ph.D.,IPM : .....
   3. Darniati, S.Kom., M.T. : .....

Mengetahui :

Pembimbing I	Pembimbing II
Rizki Yusliana Bakti, S.T., M.T.	**Muhyiddin A. M Hayat, S.Kom., M.T.**

Dekan

Ir. Muh. Syafaat S. Kuba, S.T., M.T.
NBM : 795 288

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **IMPLEMENTASI SISTEM DEKRIPSI TOKEN UNTUK MENCEGAH BERBAGI AKUN PADA APLIKASI GURU PINTAR DI SEKOLAH**

Nama : MUH. AL IQRAM MARZAH

Stambuk : 105 84 11051 21

Makassar, 08 September 2025

Telah Diperiksa dan Disetujui
Oleh Dosen Pembimbing;

Pembimbing I	Pembimbing II
Rizki Yusliana Bakti, S.T., M.T.	**Muhyiddin A M Hayat, S.Kom., M.T.**

Mengetahui,
Ketua Prodi Informatika

Rizki Yusliana Bakti, S.T., M.T.
NBM : 1307 284

MOTTO DAN PERSEMBAHAN

Motto

"Jangan bangun saat matahari bersinar, bangunlah lebih awal dan biarkan matahari melihatmu bersinar"

Persembahan

Bismillahirrahmanirrahim. Dengan penuh rasa syukur ke hadirat Allah SWT atas limpahan rahmat, hidayah, dan karunia-Nya, sehingga skripsi ini dapat terselesaikan dengan baik. Karya sederhana ini kupersembahkan kepada ayahanda tercinta, Laode Marhaba, dan ibunda tersayang, Filzah Adabiah, yang dengan doa, kasih sayang, pengorbanan, dan keikhlasannya senantiasa menjadi sumber kekuatan dalam setiap langkah perjuangan. Persembahan ini juga kutujukan untuk diriku sendiri, sebagai bukti bahwa usaha, kesabaran, dan keyakinan akan selalu menghadirkan hasil yang indah pada waktunya. Tidak lupa, ucapan terima kasih dan penghargaan yang tulus kupersembahkan kepada seluruh dosen dan pembimbing di Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Makassar, atas ilmu, bimbingan, dan inspirasi yang telah diberikan, serta kepada seluruh civitas akademika Fakultas Teknik yang telah menjadi bagian penting dalam perjalanan akademik ini. Persembahan ini juga untuk sahabat seperjuangan, teman-teman angkatan 2021 Program Studi Informatika, yang telah berbagi suka duka, tawa, dan semangat hingga akhirnya tiba di titik pencapaian ini. Semoga skripsi ini dapat memberikan manfaat, menjadi langkah kecil menuju perubahan yang lebih baik, dan menjadi ungkapan tulus atas segala dukungan serta kehadiran semua pihak dalam perjalanan hidup dan pendidikan penulis.

ABSTRAK

MUH. AL IQRAM MARZAH, Implementasi Sistem *Dekripsi* Token Untuk Mencegah Berbagi Akun Pada Aplikasi Guru Pintar di Sekolah. (dibimbing oleh Rizki Yusliana Bakti, S.T.,M.T, dan Muhyiddin AM Hayat, S.Kom.,M.T).

Penelitian ini bertujuan untuk mengimplementasikan sistem dekripsi token berbasis perangkat (*device-bound token decryption*) pada aplikasi *Guru Pintar* untuk mencegah praktik berbagi akun. Aplikasi *Guru Pintar* hadir sebagai solusi digital dalam mendukung pembelajaran, namun menghadapi permasalahan serius terkait keamanan, khususnya risiko kebocoran data akibat penggunaan akun secara bersamaan di perangkat berbeda. Sebagai solusi inovatif, sistem yang dikembangkan dirancang dengan framework Electron dan memanfaatkan *device fingerprinting* untuk mengikat token autentikasi pada identitas perangkat keras tertentu. Data penelitian diperoleh melalui proses implementasi dan diuji menggunakan metode *black box testing* dengan berbagai skenario login. Hasil pengujian menunjukkan bahwa sistem mampu menolak login dari perangkat yang berbeda meskipun kredensial valid, serta terbukti aman terhadap serangan umum seperti *SQL injection*, *XSS*, dan *token tampering*. Temuan ini menunjukkan bahwa penerapan dekripsi token berbasis perangkat efektif dalam meningkatkan keamanan autentikasi, menjaga keunikan akses pengguna, dan memperkuat perlindungan data pada aplikasi edukasi digital.

Kata Kunci : *dekripsi token*, keamanan akun, *device-bound token*, *device fingerprinting*, berbagi akun, *Electron JS*.

ABSTRAK

MUH. AL IQRAM MARZAH, *Implementation of a Token Decryption System to Prevent Account Sharing in the Guru Pintar Application in Schools*. (*supervised by* Rizki Yusliana Bakti, S.T., M.T., and Muhyiddin AM Hayat, S.Kom., M.T.).

This research aims to implement a device-bound token decryption system in the Guru Pintar application to prevent account sharing. The Guru Pintar application is presented as a digital solution to support learning, but faces serious security issues, particularly the risk of data leakage due to concurrent account use on different devices. As an innovative solution, the developed system is designed with the Electron framework and utilizes device fingerprinting to bind authentication tokens to specific hardware identities. Research data was obtained through the implementation process and tested using black box testing methods with various login scenarios. The test results show that the system is able to reject logins from different devices even though the credentials are valid, and is proven secure against common attacks such as SQL injection, XSS, and token tampering. These findings indicate that the implementation of device-based token decryption is effective in improving authentication security, maintaining the uniqueness of user access, and strengthening data protection in digital education applications.

***Keywords**: token decryption, account security, device-bound token, device fingerprinting, account sharing, Electron JS.*

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatu

Segala puji bagi Allah Subhanallahu Wa Ta'ala atas limpahan Rahmat dan Karunia-Nya, serta kesabaran dalam mempermudah jalan sehingga penulis dapat menyelesaikan penyusunan skripsi ini yang berjudul **"Implementasi Sistem Dekripsi Token Untuk Mencegah Berbagi Akun Pada Aplikasi Guru Pintar Di Sekolah"** Salawat beserta salam senantiasa penulis panjatkan kepada Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliah menuju zaman yang serba modern seperti yang kita rasakan saat ini.

Dalam penyusunan skripsi ini penulis banyak menerima bimbingan, arahan, motivasi, serta dibantu oleh berbagai pihak, baik langsung maupun tidak langsung. Penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Dr. Ir. Hj. Abd. Rakhim Nanda, S.T., M.T., IPU, selaku Rektor Universitas Muhammadiyah Makassar.
2. Bapak Ir. Muhammad Syafa'at S.Kuba, S.T., M.T, selaku Dekan Fakultas Teknik Universitas Muhammadiyah Makassar.
3. Ibu Rizki Yusliana Bakti, S.T., M.T, selaku Ketua Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.
4. Ibu Rizki Yusliana Bakti, S.T., M.T, juga selaku Dosen Pembimbing 1 yang telah memberikan arahan dan bimbingan serta saran yang sangat berarti dalam penyusunan skripsi ini.
5. Bapak Muhyiddin AM Hayat, S.Kom., M.T, selaku Dosen Pembimbing 2 yang telah memberikan arahan dan bimbingan serta saran yang sangat berarti dalam penyusunan skripsi ini.
6. Seluruh Dosen Fakultas Teknik Program Studi Informatika Universitas Muhammadiyah Makassar yang telah memberikan ilmu dan bantuanya serta doronganya dalam penulisan skripsi ini.

7. Pegawai Tata Usaha Fakultas Teknik Universitas Muhammadiyah Makassar yang telah membantu dalam pengurusan berkas dan administrasi sehingga proses penyusunan skripsi ini berjalan dengan baik.
8. Kedua orang tua sy yakni bapak laode marhaba dan ibu filzah adabiah yang selalu memberikan dukungan baik berupa moral, materi, dan spiritual agar terselesaikannya penulisan skripsi ini.
9. Kakanda Andi Agung, S.Kom, selaku senior yang telah banyak membantu penulis dan memberikan ilmu yang sangat berarti dalam penulisan skripsi ini.
10. Teman-teman angkatan 21 terkhusus kelas B informatika Fakultas Teknik Universitas Muhammadiyah Makassar
11. Kepada semua pihak yang sudah membantu, Penulis mengucapkan banyak terima kasih yang sebesar-besarnya.

Semoga kebaikan menjadi Amal Sholeh dan dibalas dengan kebaikan yang lebih oleh Allah. Demikian laporan skripsi ini, dan penulis sadar bahwa skripsi ini masih banyak kekurangan didalamnya oleh karena itu kritik dan saran yang kostruktif sangat diharapkan demi penyempurnaan skripsi ini. Akhir kata penulis ucapkan terima kasih

Billahi fisabililhaq, fastabiqul khairat.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Makassar, 30 Agustus 2025

MUH. AL IQRAM MARZAH

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

DAFTAR ISTILAH

Autentikasi	Proses verifikasi identitas pengguna untuk memastikan hanya pihak yang sah yang dapat mengakses sistem.
Dekripsi	Proses mengubah data terenkripsi kembali menjadi bentuk aslinya menggunakan kunci tertentu.
Token	Data digital berupa string terenkripsi yang digunakan untuk menjaga sesi autentikasi pengguna tanpa perlu login berulang.
JSON Web Token (JWT)	Standar token berbasis JSON untuk pertukaran informasi secara aman antar sistem dengan tanda tangan digital.
Device Fingerprinting	Teknik identifikasi perangkat berdasarkan parameter unik perangkat keras seperti MAC Address, CPU model, dan hostname.
Device-Bound Token	Token autentikasi yang terikat pada identitas unik perangkat, sehingga tidak bisa digunakan pada perangkat lain.
Electron JS	Framework berbasis JavaScript untuk membangun aplikasi desktop lintas platform dengan teknologi web.
SafeStorage API	Fitur keamanan pada Electron untuk menyimpan data terenkripsi secara aman di perangkat pengguna.

AES (Advanced Encryption Standard)	Algoritma kriptografi simetris untuk enkripsi dan dekripsi data dengan tingkat keamanan tinggi.
RSA (Rivest–Shamir–Adleman)	Algoritma kriptografi asimetris yang menggunakan pasangan kunci publik dan privat dalam proses enkripsi dan dekripsi.
HMAC (Hash-based Message Authentication Code)	Mekanisme autentikasi berbasis hash untuk memastikan integritas dan keaslian data digital.
Black Box Testing	Metode pengujian perangkat lunak yang menilai fungsi sistem dari sisi input-output tanpa melihat kode internal.
Brute Force Attack	Metode serangan dengan mencoba semua kemungkinan kombinasi kata sandi atau kunci enkripsi hingga menemukan yang benar.
SQL Injection	Serangan yang memanfaatkan celah query SQL untuk mengakses atau merusak basis data.
XSS (Cross-Site Scripting)	Serangan keamanan dengan menyisipkan skrip berbahaya ke dalam aplikasi untuk mencuri data atau mengambil alih sesi pengguna.

BAB I
PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi tela'h mendorong digitalisasi di berbagai sektor, termasuk pendidikan. Transformasi digital ini bukan sekadar respons terhadap perkembangan zaman, tetapi menjadi kebutuhan untuk meningkatkan relevansi dan daya saing sumber daya manusia di era global (Kudriani et al., 2023). Salah satu inovasi yang paling menonjol adalah hadirnya aplikasi pembelajaran digital yang memberikan pengalaman belajar interaktif dan fleksibel bagi guru maupun siswa. Aplikasi edukasi seperti *Guru Pintar* hadir untuk mempermudah manajemen materi pelajaran, penilaian, serta personalisasi proses belajar. Fitur-fitur ini dirancang agar setiap pengguna mendapatkan layanan sesuai kebutuhannya. Namun, meningkatnya penggunaan aplikasi ini membawa tantangan baru, terutama terkait keamanan akses dan perlindungan data pribadi.

Salah satu permasalahan yang sering terjadi adalah praktik berbagi akun. Aktivitas ini tidak hanya melanggar kebijakan penggunaan, tetapi juga mengganggu akurasi sistem personalisasi. Data aktivitas belajar menjadi tidak sesuai dengan perilaku asli pengguna, sehingga mengurangi efektivitas pembelajaran. Selain itu, manajemen sistem menjadi lebih sulit karena analisis penggunaan fitur tidak lagi mencerminkan data yang valid. Lebih jauh, praktik berbagi akun dapat menyebabkan kebocoran data dan penyalahgunaan hak akses (Pattinama et al., 2023).

penyalahgunaan data pribadi di media digital kerap terjadi akibat lemahnya perlindungan privasi. Informasi seperti kredensial login yang dibagikan tanpa kendali dapat dimanfaatkan pihak tidak berwenang untuk mengakses data sensitif. Hal ini berisiko menimbulkan kerugin, baik dari sisi teknis, finansial, maupun reputasi penyedia layanan (Pertiwi et al., 2022).

Untuk mengatasi masalah tersebut, diperlukan sistem autentikasi yang mampu memastikan hanya pengguna sah yang dapat mengakses akun. Salah satu solusi yang banyak digunakan adalah *JSON Web Token* (*JWT*). *JWT* memungkinkan autentikasi yang aman, namun pada implementasi konvensional masih terdapat kelemahan jika token disimpan di lokasi yang kurang aman. Token yang disimpan di *session storage* atau lokasi terbuka rentan dicuri dan digunakan di perangkat lain (Darmawan et al, 2023).

Oleh karena itu, penelitian ini mengusulkan penerapan *dekripsi* token berbasis perangkat (*device-bound token decryption*) pada aplikasi *Guru Pintar*. Pendekatan ini mengaitkan token autentikasi dengan identitas unik perangkat keras, sehingga token hanya dapat digunakan pada perangkat asal. Proses *dekripsi* dilakukan secara lokal menggunakan kunci yang dihasilkan dari parameter perangkat, dan token disimpan dalam media yang aman seperti *safeStorage API* milik *Electron* atau file terenkripsi menggunakan *algoritma kriptografi* modern seperti *AES* (Rahman et al, 2024). Pendekatan ini secara signifikan mengurangi risiko pencurian token dibandingkan menyimpannya di penyimpanan peramban web standar, sesuai dengan kebutuhan strategi pengamanan data pada aplikasi yang mengelola informasi sensitif (Astowo and Sujarwo, 2023).

Metode ini memberikan dua keuntungan utama. Pertama, tingkat keamanan meningkat karena token yang dicuri tidak akan berlaku di perangkat lain. Kedua, pengalaman pengguna tetap nyaman karena tidak memerlukan autentikasi tambahan seperti *OTP* atau *TOTP* setiap kali *login* (Sitorus et al., 2020). Selain itu, pendekatan ini sejalan dengan tren keamanan modern yang mengedepankan *multi-layer security* dan *device-based authentication*, yang terbukti efektif mengurangi risiko penyalahgunaan akun di berbagai platform digital.

Dengan demikian, penerapan sistem dekripsi token pada aplikasi *Guru Pintar* diharapkan dapat memberikan perlindungan lebih terhadap data pengguna, menjaga keakuratan personalisasi pembelajaran, dan memastikan

kepercayaan pengguna terhadap platform tetap terjaga. Penelitian ini akan merancang, membangun, dan menguji mekanisme tersebut, sehingga dapat menjadi referensi untuk pengembangan keamanan di aplikasi edukasi sejenis.

B. Rumusan Masalah

Berdasarkan konteks permasalahan yang telah dijelaskan di atas, permasalahan pokok yang dihadapi adalah:

1. Bagaimana sistem dekripsi token dapat diimplementasikan untuk meningkatkan validitas dan keunikan akses pengguna dalam aplikasi Guru Pintar ?
2. Bagaimana memastikan keamanan akses akun pengguna pada aplikasi Guru Pintar agar tidak dapat digunakan oleh pihak yang tidak berwenang ?

C. Tujuan Penelitian

Berdasarkan uraian rumusan masalah tersebut, maka dirumuskan tujuan penelitian sebagai berikut, yakni:

1. Mengimplementasikan sistem *dekripsi* token untuk meningkatkan validitas dan keunikan akses pengguna pada aplikasi Guru Pintar.
2. Memastikan keamanan akses akun pengguna pada aplikasi Guru Pintar agar tidak dapat digunakan oleh pihak yang tidak berwenang.

D. Manfaat Penelitian

1. Terhadap Peneliti

   Penelitian ini bermanfaat untuk mengaplikasikan pengetahuan yang telah diperoleh selama masa studi dan memperluas wawasan mengenai keamanan akses akun pengguna melalui penerapan sistem dekripsi token.
2. Terhadap Institusi Akademik

   Penelitian ini bermanfaat untuk menambah kontribusi ilmiah di bidang keamanan aplikasi digital serta memperkaya referensi penelitian yang dapat

digunakan oleh mahasiswa, dosen, dan peneliti di lingkungan institusi akademik.

3. Terhadap Pengguna

   Mempermudah pengguna dalam mengakses akun mereka dengan aman serta memastikan bahwa hanya pengguna yang sah yang dapat mengakses informasi sensitif.

E. Ruang Lingkup Penelitian

Berdasarkan perumusan masalah, maka batasan masalah yang dibahas meliputi:

1. Ruang lingkup penelitian ini terbatas pada mekanisme pengamanan token di sisi klien, dan tidak mencakup aspek keamanan pada level basis data maupun *server.*
2. Aplikasi yang dikembangkan dalam penelitian ini menggunakan *Electron JS* untuk platform desktop, sehingga implementasinya tidak mencakup platform web atau mobile.

F. Sistematika Penelitian

Bab I Pendahuluan Bab ini memuat uraian mengenai latar belakang permasalahan, perumusan masalah, tujuan dari penelitian, kegunaan serta manfaat penelitian, dan sistematika penulisan

Bab II Tinjauan Pustaka Bab ini menyajikan dasar teori yang relevan, hasil-hasil penelitian sebelumnya, kerangka pemikiran, serta hipotesis yang digunakan dalam penelitian.

Bab III Metodologi Penelitian Bab ini menjelaskan lokasi tempat pengumpulan data, populasi dan sampel yang diteliti, definisi operasional variabel, alat atau instrumen penelitian yang digunakan, pengujian validitas dan reliabilitas, serta metode analisis data yang diterapkan.

BAB IV HASIL DAN PEMBAHASAN Bab ini menjelaskan hasil penelitian dan pembahasannya dalam menerapkan sistem dekripsi token dan pengambilan id device untuk mencegah berbagi akun

BAB V PENUTUP Bab ini menjelaskan kesimpulan dari hasil penelitian serta saran untuk penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

A. Kajian Teori

1. Keamanan Sistem Informasi

Keamanan sistem informasi adalah elemen penting untuk memastikan kelangsungan operasional dan melindungi aset digital organisasi. Ini mencakup langkah-langkah untuk mencegah akses tidak sah, menjaga integritas data, dan memastikan sistem tetap tersedia bagi pengguna yang berwenang. Dalam menghadapi dinamika ancaman siber yang semakin kompleks, berbagai pendekatan keamanan terus dikembangkan. penerapan keamanan sistem informasi di era modern menekankan pentingnya teknologi terkini, seperti enkripsi, autentikasi *multifaktor* (*MFA*), dan pendekatan arsitektur *Zero Trust* untuk menghadapi kompleksitas ancaman siber yang terus berkembang (Hafsah dan Nasution., 2024). Selain itu, kecerdasan buatan (AI) dan teknologi *blockchain* juga berperan dalam memperkuat keamanan data dan informasi.

Selain teknologi, keamanan sistem juga membutuhkan strategi pengelolaan yang efektif. pentingnya penerapan strategi keamanan berlapis (*layered security*) dan pemantauan sistem secara *real-time* dalam infrastruktur teknologi informasi sebagai langkah mitigasi terhadap serangan siber (Hosmand and Ratnawati., 2023). Pendekatan ini memungkinkan sistem tetap tangguh dan adaptif dalam menghadapi berbagai jenis ancaman.

Salah satu komponen teknis yang tak kalah penting dalam pengembangan sistem adalah penerapan teknologi *kriptografi* terkini. Protokol *WPA3* memang menunjukkan peningkatan signifikan dalam hal keamanan dibandingkan versi sebelumnya. Namun, penelitian mereka juga mengidentifikasi sejumlah celah, seperti potensi serangan downgrade dan kelemahan dalam proses *handshake SAE*, yang masih dapat dimanfaatkan

oleh pihak tidak berwenang (Halbouni, Ong, dan Leow., 2023). Oleh karena itu, untuk memperkuat sistem keamanan, sangat disarankan agar protokol ini dilengkapi dengan mekanisme deteksi intrusi yang bersifat adaptif serta penerapan patch keamanan secara berkala guna mengantisipasi eksploitasi terhadap kerentanan yang mungkin muncul.

keamanan data dalam Sistem Informasi Manajemen (SIM) sangat penting untuk melindungi informasi dari ancaman seperti pencurian data dan kerusakan fisik terhadap sistem informasi (Wijoyo, Fatimah, and Widianti., 2023).

2. Teknologi Token dan Dekripsi

   Token merupakan komponen autentikasi yang digunakan untuk mempertahankan sesi pengguna tanpa harus melakukan login secara berulang. Pada sistem yang menggunakan token, biasanya token tersebut dienkripsi guna mencegah modifikasi atau penyalahgunaan oleh pihak yang tidak memiliki izin. Beberapa metode yang digunakan dalam teknologi token antara lain

   a. *JSON Web Token (JWT)*

      Secara umum, *JSON Web Token (JWT)* adalah format token berbasis *JSON* yang ringan dan efisien, serta dapat ditransmisikan melalui protokol *HTTP* tanpa membutuhkan penyimpanan sesi di sisi server. *JWT* banyak digunakan dalam sistem autentikasi modern karena mampu memverifikasi identitas pengguna dan menjaga integritas data secara aman dan praktis.

   b. *Refresh* Token

      Sementara itu, *refresh* token adalah jenis token yang digunakan untuk memperbarui sesi pengguna tanpa harus melalui proses autentikasi dari awal. Token ini biasanya digunakan ketika *access* token telah kedaluwarsa, sehingga pengguna tetap dapat melanjutkan sesi tanpa perlu login kembali. Dengan demikian, refresh token berperan penting

dalam menjaga kenyamanan pengguna sekaligus mempertahankan keamanan sistem.

Dalam sistem keamanan, *dekripsi* token bertujuan untuk mengonversi token yang telah *dienkripsi* menjadi format yang dapat divalidasi. Beberapa algoritma dekripsi yang sering digunakan antara lain:

a. *Advanced Encryption Standard* (*AES*)

   Untuk menjaga kerahasiaan dan keamanan data, terutama token dalam sistem, diperlukan algoritma enkripsi yang kuat dan efisien. *Advanced Encryption Standard* (*AES*) merupakan *algoritma kriptografi* simetris yang secara luas digunakan dalam proses *enkripsi* dan *dekripsi* token karena kemampuannya memberikan tingkat perlindungan yang tinggi. Dengan menggunakan kunci yang sama untuk proses *enkripsi* dan *dekripsi*, *AES* sangat efektif dalam mencegah akses yang tidak sah. Selain itu, performa sistem *AES* dalam menyembunyikan data sangat unggul, menjadikannya pilihan utama untuk aplikasi yang membutuhkan keamanan data yang handal (Adigun et al., 2024).

b. *Rivest-Shamir-Adleman* (*RSA*)

   Dalam sistem keamanan digital, autentikasi berperan penting untuk memastikan bahwa hanya pengguna yang memiliki hak akses yang dapat menggunakan layanan. Untuk mendukung proses ini, digunakan algoritma kriptografi yang terpercaya dan aman. *Rivest-Shamir-Adleman* (*RSA*) merupakan algoritma kriptografi asimetris yang sering digunakan dalam skema autentikasi berbasis token. Algoritma ini memanfaatkan pasangan kunci publik dan privat yang memungkinkan proses enkripsi dan dekripsi dilakukan secara terpisah, sehingga sangat sesuai untuk menjaga kerahasiaan dan validitas token selama proses autentikasi berlangsung.

c. *Hash-based Message Authentication Code* (*HMAC*)

HMAC merupakan teknik *kriptografi* yang berfungsi untuk menjamin integritas dan autentisitas data, termasuk token. Dalam sistem ini, *HMAC* digunakan untuk memverifikasi bahwa token tetap utuh setelah proses *dekripsi*, sehingga dapat dipastikan bahwa token tidak mengalami modifikasi atau pemalsuan. Dengan cara ini, sistem dapat memastikan bahwa token berasal dari sumber yang terpercaya dan belum diubah. Selain itu, token yang dihasilkan hanya berlaku untuk satu perangkat tertentu, sehingga tidak bisa digunakan di perangkat lain. Mekanisme ini meningkatkan tingkat keamanan dengan mencegah praktik berbagi akusn antar pengguna. *HMAC-SHA256* terbukti ampuh dalam menjaga integritas data serta mendeteksi perubahan atau manipulasi yang tidak sah terhadap informasi digital (Uriawan et al., 2024). Pendekatan ini dapat diterapkan dengan berbagai metode seperti :

1) *Device Fingerprinting*
   Memanfaatkan parameter perangkat seperti *MAC address*, tipe sistem operasi, serta *konfigurasi* perangkat keras untuk membuat token unik yang terhubung dengan perangkat spesifik.
2) Binding Token ke Perangkat
   Mengasosiasikan token dengan identitas unik dari suatu perangkat, sehingga token tersebut tidak bisa dipakai pada perangkat yang berbeda.
3) *Stateful* vs. *Stateless* Token
   Token bisa bersifat *stateful*, yaitu bergantung pada penyimpanan sesi di sisi server, atau *stateless*, yang artinya dapat *diverifikasi* tanpa memerlukan penyimpanan tambahan di server.

4. Penerapan Dekripsi Token dalam Sistem Keamanan

Dalam sistem keamanan yang mengatur akses berdasarkan perangkat, proses dekripsi token digunakan untuk memverifikasi bahwa token tersebut hanya dapat digunakan oleh perangkat yang telah terdaftar atau dianggap

valid. Beberapa langkah dalam penerapan dekripsi token meliputi Pembuatan token asli, enkripsi token, dekripsi token, validasi perangkat, penolakan akses jika perangkat berbeda.

Dalam penerapan dekripsi token untuk sistem keamanan berbasis perangkat, validasi perangkat yang sah sangat penting untuk mencegah akses tidak sah. Penggunaan algoritma kriptografi dalam pengamanan data pelanggan dan penjualan dapat meningkatkan keamanan sistem secara signifikan (Vivi Wahdini et al.,2021). Penelitian tersebut menunjukkan bahwa implementasi *algoritma kriptografi* yang tepat dapat mengamankan data sensitif dari akses yang tidak diinginkan. Hal ini menekankan pentingnya proses *dekripsi* token yang efektif untuk memastikan bahwa hanya perangkat yang telah terdaftar atau dianggap valid yang dapat mengakses sistem. Dengan demikian, integrasi dekripsi token yang kuat menjadi kunci dalam menjaga integritas dan keamanan sistem secara keseluruhan.

5. Token binding untuk Mencegah Penyalahgunaan Token

   Token binding adalah metode keamanan yang mengaitkan token autentikasi dengan perangkat atau klien tertentu melalui penggunaan *kriptografi*. Dengan teknik ini, token hanya dapat diakses oleh perangkat yang memiliki kunci privat yang sesuai, sehingga mencegah penggunaan oleh pihak yang tidak berwenang. Teknik ini mengurangi risiko pencurian token dan serangan replay dengan memastikan bahwa hanya aplikasi yang mengajukan permintaan token yang dapat menggunakannya untuk mengakses sumber daya terkait. Dengan demikian, keamanan akses terhadap sumber daya menjadi lebih terjamin.

6. Manajemen Identitas dan Akses

   Manajemen identitas dan akses (IAM) adalah proses mengelola identitas digital dan mengontrol akses ke sumber daya dalam sistem informasi. penerapan IAM yang efektif melibatkan otentikasi, otorisasi, audit akses, dan kebijakan manajemen identitas untuk memastikan bahwa hanya pengguna yang berwenang yang dapat mengakses informasi sensitif

(Almalki, 2024). Teknologi seperti *Single Sign-On (SSO)* dan *Federated Identity Management* semakin banyak digunakan untuk meningkatkan kenyamanan dan keamanan dalam pengelolaan identitas.

7. Penyimpanan Aman pada Aplikasi Desktop

   Dalam aplikasi desktop berbasis *Electron*, keamanan penyimpanan token sangat penting. Berdasarkan dokumentasi *Electron*, penyimpanan aman dapat dilakukan dengan menggunakan modul *safeStorage*, *enkripsi* file lokal, serta *enkripsi* berbasis sistem operasi seperti *Keychain* di *macOS* atau *Credential Locker* di Windows. Penyimpanan seperti *localStorage* dianggap tidak aman karena dapat diakses oleh pihak ketiga melalui *DevTools*.

8. Perbandingan Sistem *Konvensional* vs Token Terikat Perangkat

   Sistem autentikasi *konvensional* yang hanya mengandalkan *username* dan *password* rentan terhadap pembajakan akun, terutama jika informasi kredensial tersebar luas. Sistem token *konvensional* juga dapat disalahgunakan jika token berpindah perangkat. Sebaliknya, token yang terikat pada perangkat memastikan bahwa token hanya valid jika digunakan pada perangkat asalnya, karena token tersebut di*enkripsi* dengan parameter unik dari perangkat tersebut. Pendekatan ini meningkatkan keamanan dengan memastikan bahwa hanya perangkat yang sah yang dapat menggunakan token autentikasi tersebut.

B. Penelitian Terkait

Tabel 1. Penelitian terkait

Penelitian	Tujuan/Kasus	Metode	Hasil
Evaluasi Keamanan *Privilege* Terintegerasi *JSON Web Token* pada Sistem Informasi Akademik	Mengevaluasi keamanan otorisasi dan privilege pada sistem informasi akademik berbasis JWT di Universitas	Metode *eksperimental*	JWT telah digunakan namun implementasinya belum optimal. Terdapat celah seperti token lama yang masih aktif dan validasi otorisasi yang

Penelitian	Tujuan/Kasus	Metode	Hasil
(Darmawan et al., 2023)	Madura. Fokus pada token logout, penyimpanan token, dan audit logging.		lemah pada beberapa endpoint. Disarankan perbaikan sistem seperti pemblokiran token kadaluarsa dan penerapan audit logging.
Implementasi JSON Web Token Authentication pada Aplikasi Pembayaran Berbasis Mobile (Sopingi, Nastiti, and Majid, 2021)	Penelitian ini bertujuan untuk meningkatkan keamanan proses autentikasi dan pengiriman data pada aplikasi pembayaran berbasis mobile di Universitas Duta Bangsa. Fokus utama adalah melindungi data mahasiswa dan memastikan transaksi yang dikirim dari aplikasi ke web service sistem keuangan aman dan tervalidasi.	Metode *Rapid Application Development* (RAD)	Hasil penelitian menunjukkan bahwa sistem berhasil melakukan autentikasi menggunakan JWT dan mengamankan data dengan hash token. Pengujian menggunakan teknik User Acceptance Testing (UAT) membuktikan bahwa sistem berjalan sesuai kebutuhan, bebas bug, dan memberikan keamanan serta kenyamanan dalam proses pembayaran.
Utilization of JWT Tokens as an Authenticity Validation Method for Correspondence at Muhammadiyah University of East	Penelitian ini bertujuan untuk meningkatkan keamanan dan keakuratan dalam proses validasi surat elektronik di Universitas Muhammadiyah	Metode implementasi sistem menggunakan *framework Django* dan pengujian *blackbox*	Hasil penelitian menunjukkan bahwa sistem berhasil menyisipkan JWT token ke dalam surat dan memverifikasinya secara otomatis melalui UUID.

Penelitian	Tujuan/Kasus	Metode	Hasil
Kalimantan (Hendra Saputra, Sayekti Harits Suryawan, Faldi, Bulan Suci Cahayawati, dan Ririn Wahyuni, 2023)	Ka limantan Timur. Sistem sebelumnya menggunakan QR code yang mudah dipalsukan. Oleh karena itu, JWT diterapkan sebagai metode tambahan untuk memastikan keaslian surat melalui penyisipan token ke dalam dokumen PDF.		Verifikasi keaslian surat berbasis JWT lebih efisien dan akurat dibandingkan dengan QR code. Sistem mampu membedakan surat yang valid dan tidak valid berdasarkan kecocokan UUID dan isi surat, serta mendukung pengelolaan dokumen yang lebih aman dan terpercaya.
Penerapan JSON Web Token sebagai Strategi Pengamanan Data pada Aplikasi MultiMasjid (Astowo and Sujarwo 2023)	Penelitian ini bertujuan untuk merancang dan menerapkan JSON Web Token (JWT) dengan algoritma hash SHA-512 dalam aplikasi Multimasjid untuk meningkatkan keamanan otorisasi dan autentikasi pengguna, khususnya dalam pengelolaan data jamaah yang bersifat privat.	Metode *Waterfall*	Hasil penelitian menunjukkan bahwa penerapan JWT dengan algoritma SHA-512 dapat meningkatkan keamanan aplikasi Multimasjid. Sistem berhasil melakukan autentikasi dan otorisasi secara efisien serta mencegah akses tidak sah. Pengujian menggunakan Postman menunjukkan bahwa hanya permintaan dengan token valid yang dapat mengakses data,

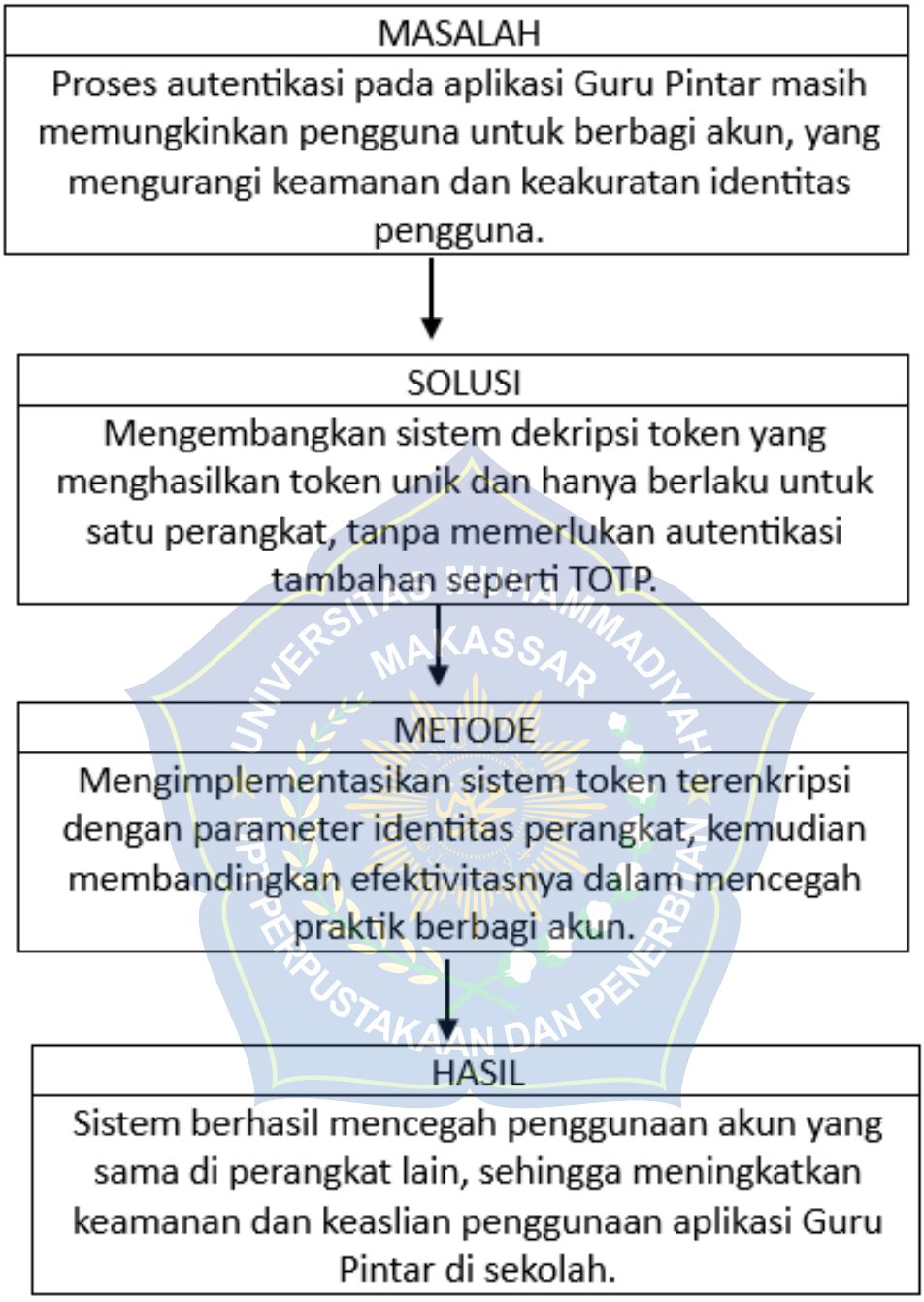
Penelitian	Tujuan/Kasus	Metode	Hasil
			sedangkan permintaan tanpa token ditolak. Ini membuktikan efektivitas JWT dalam menjaga keamanan data dan hak akses pengguna.
Implementasi *Rest API Web Service* Dengan *Otentifikasi JSON* Web Token Untuk Aplikasi Properti (Loui Pattinama et al. 2023)	Penelitian ini bertujuan untuk mengatasi permasalahan keterlambatan informasi dan kurangnya kepercayaan konsumen di sektor properti PT. Arifindo Adiputra Ariaguna. Solusi yang ditawarkan adalah membangun aplikasi berbasis web dan mobile yang terintegrasi menggunakan RESTful API dan dilindungi oleh JSON Web Token (JWT) agar proses transaksi dan koordinasi antar divisi menjadi lebih efisien dan aman.	Metode rekayasa perangkat lunak berorientasi implementasi (*waterfall/implementatif*) dan pengujian *blackbox*	Hasil penelitian menunjukkan bahwa penerapan RESTful API dengan otentikasi JWT berhasil meningkatkan efisiensi pertukaran data antara aplikasi web dan mobile. Sistem mampu memberikan keamanan akses dengan validasi token JWT serta menghasilkan respon sesuai hak akses pengguna. Aplikasi terbukti berjalan lancar pada dua platform, mudah digunakan, dan mempercepat proses transaksi serta koordinasi tim internal perusahaan.
Implementasi Sistem *Dekripsi* Token untuk	Mencegah praktik berbagi akun dengan	Metode rekayasa perangkat lunak menggunakan *Electron*	Hasil menunjukkan sistem dekripsi

Penelitian	Tujuan/Kasus	Metode	Hasil
Mencegah Berbagi Akun pada Aplikasi Guru Pintar di Sekolah (Muh. Al Iqram Marzah, 2025	menerapkan sistem dekripsi token yang hanya dapat digunakan pada satu perangkat.		token berhasil membatasi akses hanya pada perangkat terdaftar. Autentikasi menjadi lebih aman dan tidak dapat digunakan di perangkat lain tanpa proses registrasi ulang.

C. Kerangka Berpikir

Kerangka berpikir adalah model konseptual yang menjelaskan hubungan antara teori dan berbagai faktor yang telah diidentifikasi sebagai masalah penting dalam penelitian ini. Secara teoritis, sistem *dekripsi* token dapat diterapkan untuk meningkatkan keamanan autentikasi dengan menghasilkan token yang unik dan hanya berlaku untuk satu perangkat. Pendekatan ini bertujuan untuk meminimalkan potensi penyalahgunaan akun melalui praktik berbagi akun. Implementasi sistem *dekripsi* token tanpa autentikasi tambahan seperti *TOTP* bertujuan memberikan solusi yang efisien namun tetap aman dalam penggunaan aplikasi Guru Pintar di lingkungan sekolah.

Kerangka berpikir ini akan menggambarkan hubungan teoritis antara variabel seperti token, identitas perangkat, dan sistem autentikasi, serta bagaimana interaksi antar variabel tersebut dapat mencegah berbagi akun



Gambar 1. Kerangka berpikir

BAB III
METODE PENILITIAN

A. Tempat dan Waktu Penelitian

1. Lokasi Penelitian

   Tempat penelitian merupakan lokasi atau objek yang dijadikan sebagai fokus pelaksanaan penelitian guna mempermudah proses pengumpulan data. Dalam hal ini, peneliti memilih Universitas Muhammadiyah Makassar, tepatnya di laboratorium komputer Fakultas Teknik, sebagai lokasi pelaksanaan penelitian.

2. Waktu Penelitian

   Waktu penelitian akan di lakukan dalam jangka kurang lebih 2 bulan, yang akan di mulai pada Mei 2025 hingga seluruh prosedur pengumpulan data selesai

B. Alat dan Bahan

Adapun alat dan bahan yang akan digunakan dalam penelitian ini, yaitu :

1. Kebutuhan *Hardware* (Perangkat Keras)
   a. Laptop *Asus Vivobook*
   b. Besar Memory Ram 16 GB
   c. Kapasitas SSD 512 GB
2. Kebutuhan *Software* (Perangkat Lunak)
   a. Visual Studio Code
   b. Windows – electron
   c. MySQL
   d. Javascript

C. Perancangan Sistem

Tahapan perancangan sistem dalam penelitian ini berfokus pada pengembangan autentikasi berbasis token yang terikat pada identitas unik perangkat keras pengguna. Sistem ini diimplementasikan pada aplikasi klien

"Guru Pintar" yang dibangun menggunakan kerangka kerja Electron. Tujuan utama perancangan ini adalah untuk memastikan bahwa token autentikasi yang diterbitkan hanya dapat didekripsi dan digunakan pada perangkat asal, sehingga dapat mencegah potensi penyalahgunaan akun melalui mekanisme berbagi token antar perangkat. Adapun proses system yakni:

a. Aplikasi Klien *Electron*

Komponen ini bertanggung jawab atas pengelolaan alur autentikasi pengguna, permintaan token ke server, penyimpanan token secara aman dan persisten, ekstraksi pengidentifikasi unik perangkat (menggunakan teknik seperti pustaka node-machine-id atau kombinasi data dari modul os Node.js), serta proses dekripsi atau validasi token di sisi klien sebelum mengirim permintaan ke sumber data server.

b. Autentikasi Server

Komponen server autentikasi bertanggung jawab untuk memverifikasi kredensial pengguna saat login, menerbitkan token autentikasi (seperti JWT) yang dienkripsi dan diikat dengan informasi identitas perangkat yang diterima dari klien, serta memvalidasi token pada setiap permintaan yang masuk

c. Proses Penerbitan dan Enkripsi Token

Setelah kredensial pengguna berhasil diverifikasi, aplikasi klien memulai proses identifikasi perangkat untuk menghasilkan ID unik (misalnya, menggunakan node-machine-id). ID ini kemudian dikirimkan ke server bersama dengan permintaan penerbitan token. Server memproses permintaan tersebut dan menghasilkan token autentikasi. Token ini kemudian dienkripsi menggunakan algoritma kriptografi yang telah ditentukan (seperti AES atau RSA), di mana kunci enkripsi dapat diderivasi atau dikaitkan secara kriptografis dengan ID perangkat unik yang diterima.

ID perangkat unik klien diintegrasikan ke dalam payload token sebelum proses enkripsi atau digunakan sebagai parameter dalam derivasi kunci enkripsi di sisi server, untuk memastikan token hanya dapat digunakan pada perangkat asal.

d. Rencana Penyimpanan Token

Token yang diterima oleh aplikasi klien disimpan dalam mekanisme penyimpanan persisten yang aman dan terpisah dari penyimpanan standar peramban web (seperti *localStorage* atau *sessionStorage*) yang rentan terhadap akses tidak sah. Alternatif penyimpanan dalam lingkungan Electron meliputi penggunaan API safeStorage, penyimpanan dalam berkas terenkripsi dengan akses terbatas pada proses aplikasi, atau pengelolaan token melalui Inter-Process Communication (IPC) di proses utama Electron

e. Proses dekripsi dan validasi token di sisi klien

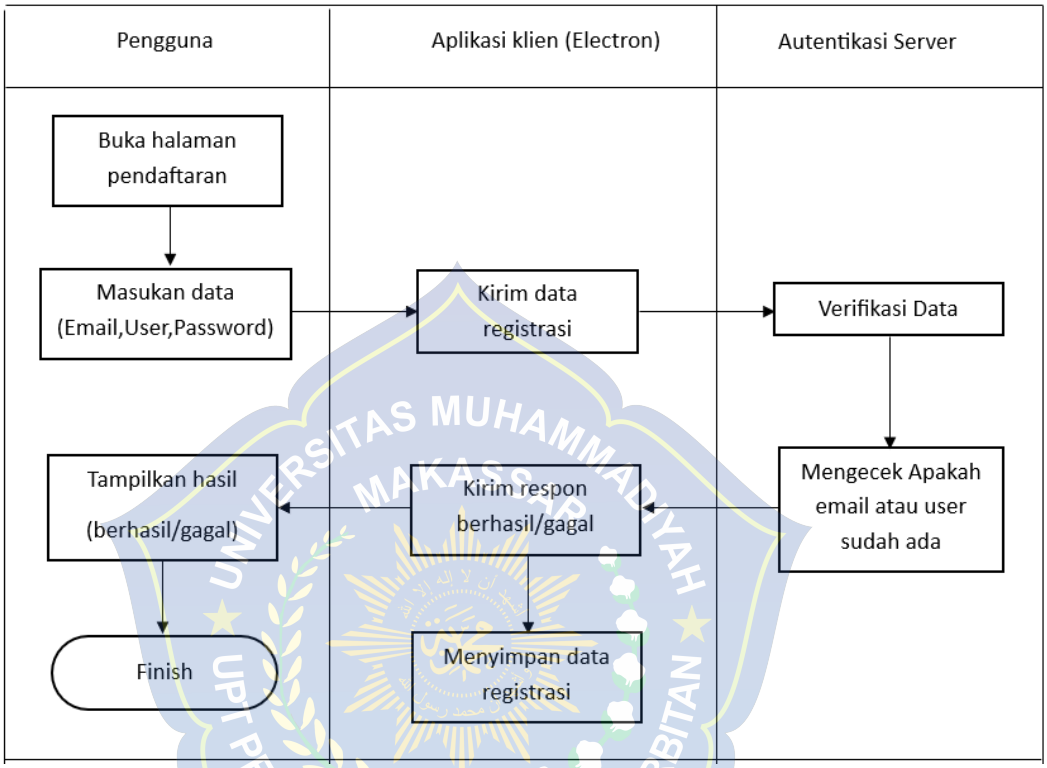
Setiap kali ada permintaan akses ke sumber daya server yang memerlukan autentikasi, aplikasi klien akan mengambil token yang tersimpan, mengekstraksi kembali ID perangkat saat ini menggunakan metode identifikasi yang konsisten seperti mengeksekusi ulang node-machine-id, dan kemudian melakukan upaya dekripsi atau validasi token dengan menggunakan ID perangkat yang baru diekstraksi sebagai parameter kunci atau validasi.

Jika proses dekripsi atau validasi berhasil (menunjukkan bahwa ID perangkat saat ini cocok dengan ID yang terikat pada token), token tersebut dianggap valid untuk digunakan pada perangkat tersebut dan akan dikirimkan ke server. Namun, jika proses dekripsi atau validasi gagal (karena ketidakcocokan ID perangkat), ini menunjukkan kemungkinan penggunaan token pada perangkat yang tidak sah, sehingga permintaan akses akan ditolak oleh klien sebelum mencapai server.

Sebelum masuk pada tahapan teknis pengujian sistem, terlebih dahulu diperlukan penjelasan alur proses utama yang menjadi dasar pengembangan, yaitu login dan registrasi pengguna. Kedua aktivitas ini digambarkan melalui activity diagram yang menunjukkan bagaimana data pengguna, kredensial, serta perangkat diolah dan diverifikasi oleh sistem. Dari diagram tersebut, dapat dipahami alur interaksi antara aplikasi klien, server autentikasi, serta mekanisme pengelolaan token. Dengan pemahaman ini, pembahasan kemudian dilanjutkan pada tahap perancangan sistem yang secara lebih rinci menjelaskan komponen,

proses, serta implementasi teknis dari autentikasi berbasis token yang terikat pada perangkat.

Diagram activity register:



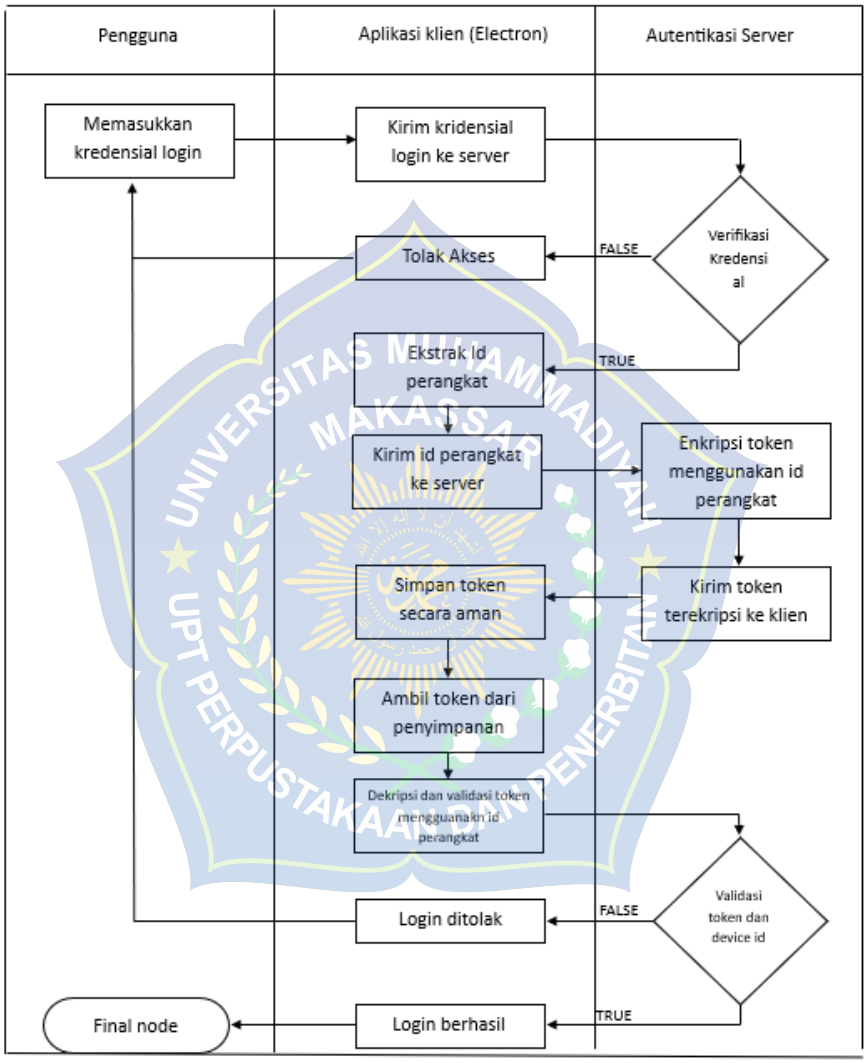
Gambar 2. *Activity diagram register*

Alur register :

Alur pendaftaran diawali ketika pengguna mengakses halaman formulir registrasi melalui aplikasi klien berbasis Electron, di mana mereka perlu memasukkan informasi esensial seperti alamat email, nama pengguna, dan kata sandi. Setelah data tersebut diisi, aplikasi klien akan meneruskannya ke server autentikasi untuk proses validasi. Di sisi server, dilakukan pemeriksaan terhadap format data sekaligus pengecekan ke dalam basis data untuk memastikan bahwa email dan nama pengguna yang diajukan belum pernah terdaftar. Selanjutnya, server mengirimkan umpan balik ke aplikasi klien mengenai status keberhasilan registrasi. Jika proses registrasi sukses, aplikasi klien akan melakukan penyimpanan data pengguna secara lokal.

Sebagai langkah terakhir, aplikasi akan menampilkan notifikasi kepada pengguna yang mengonfirmasi apakah pendaftaran mereka berhasil atau gagal.

Activity diagram login :



Gambar 3. *Activity diagram login*

Alur Activity :

Proses dimulai ketika pengguna melakukan *login* dan memasukkan kredensial (email dan kata sandi). Aplikasi klien (Electron) mengirimkan kredensial ini ke *server autentikasi* untuk diverifikasi. Jika verifikasi gagal, server mengirimkan respons penolakan dan akses ditolak. Jika verifikasi

berhasil, aplikasi klien akan mengekstrak ID perangkat dan mengirimkannya ke server. Server kemudian mengenkripsi token menggunakan ID perangkat tersebut, lalu mengirimkan token terenkripsi kembali ke klien. Aplikasi klien menyimpan token ini secara aman.

Ketika pengguna meminta akses, token diambil dari penyimpanan dan didekripsi serta divalidasi menggunakan ID perangkat. Jika token tidak valid pada tahap ini, akses ditolak. Jika valid, token tersebut dikirim kembali ke server untuk divalidasi akhir.

Server akan memeriksa token. Jika token valid, akses diizinkan. Jika tidak, akses ditolak.

D Teknik Pengujian Sistem

Dalam penelitian ini, sistem diuji menggunakan metode Black Box Testing, yaitu pendekatan pengujian perangkat lunak yang berfokus pada fungsi sistem dari sisi pengguna tanpa memperhatikan struktur internal atau kode program. Teknik ini dilakukan dengan memberikan berbagai skenario input pada sistem dan mengamati output yang dihasilkan, kemudian membandingkannya dengan spesifikasi yang telah ditentukan.

Pengujian dimulai pada fitur registrasi akun. Pada skenario registrasi dengan data valid (username, email, password benar, serta device baru), sistem diharapkan dapat menyimpan akun pada database dan mengikatnya dengan device ID pengguna. Sebaliknya, ketika input registrasi tidak valid seperti username kosong, format email yang salah, atau password yang lemah, sistem harus menolak pendaftaran dengan menampilkan pesan error sesuai validasi. Selain itu, pengujian juga dilakukan untuk memastikan bahwa sistem menolak registrasi apabila device ID yang digunakan sudah terdaftar pada akun lain, dengan menampilkan pesan error “Device is already registered to another account”.

Selanjutnya, pengujian dilakukan pada fitur login. Apabila pengguna memasukkan username dan password yang benar pada perangkat yang sesuai,

sistem harus menerima login, menghasilkan token terenkripsi, dan menyimpannya secara aman di sisi klien. Namun, ketika password yang dimasukkan salah, sistem harus menolak akses dan menampilkan pesan error “Invalid username or password.” Pengujian juga mencakup login menggunakan device berbeda meskipun kredensial benar. Pada skenario ini, sistem wajib menolak akses dengan menampilkan pesan error “This account is registered to a different device”. Selain itu, dilakukan pengujian terhadap kondisi login tanpa input. Jika pengguna tidak mengisi username dan password, sistem harus menolak login, menandai field kosong dengan warna merah, dan mencegah proses autentikasi dilanjutkan.

Berdasarkan hasil pengujian, semua skenario tersebut berjalan sesuai dengan requirement. Sistem mampu menerima data yang valid, menolak input tidak valid, membatasi penggunaan akun hanya pada perangkat yang terdaftar, serta memberikan pesan kesalahan yang sesuai pada setiap kondisi. Hal ini menunjukkan bahwa sistem autentikasi berbasis dekripsi token telah berfungsi dengan baik sesuai perspektif pengguna akhir.

E Teknik Analisis Data

Langkah-langkah analisis data untuk penelitian ini adalah sebagai berikut:

a. Pendaftaran Akun

Tahap pendaftaran akun dianalisis untuk menilai efisiensi dan keamanan proses registrasi pengguna. Proses ini melibatkan input data seperti nama, email, dan kata sandi, yang kemudian dikirim ke server untuk divalidasi. Validasi mencakup format data, keunikan email, dan kekuatan kata sandi. Jika data valid, akun disimpan di basis data dan sistem mengirimkan respons sukses. Jika tidak, pengguna menerima pesan kesalahan seperti "Email sudah terdaftar" atau "Kata sandi terlalu lemah". Evaluasi dilakukan melalui simulasi pendaftaran pada aplikasi, dengan fokus pada waktu respons, keberhasilan registrasi, dan akurasi validasi. Dokumentasi teknis dan hasil simulasi digunakan untuk menilai keandalan dan efisiensi sistem.

b. Pengujian Authentikasi

Pengujian autentikasi dilakukan secara langsung pada sistem aplikasi melalui simulasi penggunaan. Evaluasi ini berfokus pada proses login dan mekanisme keamanan akun dengan mengamati respons sistem terhadap berbagai skenario login. Informasi terkait sistem diperoleh dari dokumentasi teknis aplikasi dan hasil pengujian fungsional, langsung dengan pengguna. Jika pengujian berhasil, sistem akan memberikan akses sesuai dengan kredensial yang valid dan menolak upaya login dari perangkat yang tidak sah. Sebaliknya, jika pengujian gagal, sistem tidak mengizinkan akses dan memberikan respons yang sesuai, seperti pesan kesalahan atau permintaan verifikasi ulang.

c. *Preprocessing*

Data yang telah dikumpulkan kemudian diproses melalui tahap pra-pemrosesan (*preprocessing*). Dalam konteks penelitian ini, preprocessing mencakup identifikasi struktur data login pengguna, identifikasi parameter token, serta analisis alur login sebelum dan sesudah implementasi sistem dekripsi token.

d. Analisis Implementasi

Pada tahap ini, analisis dilakukan terhadap sistem dekripsi token yang telah diimplementasikan. Pengujian dilakukan dengan berbagai skenario penggunaan, termasuk login di perangkat berbeda untuk memastikan token tidak dapat digunakan lintas perangkat. Hasil pengujian dianalisis berdasarkan efektivitas sistem dalam mencegah berbagi akun.

e. *Display Data*

Data hasil implementasi dan pengujian disajikan dalam bentuk tabel, grafik, dan narasi deskriptif. Penyajian ini bertujuan untuk memudahkan pemahaman mengenai efektivitas sistem dekripsi token yang dikembangkan.

f. Pengambilan Kesimpulan

Berdasarkan hasil analisis, kesimpulan ditarik mengenai efektivitas sistem yang dibangun. Kesimpulan ini mempertimbangkan keberhasilan sistem dalam membatasi penggunaan akun pada satu perangkat serta masukan dari pengguna selama proses pengujian.

BAB IV
HASIL DAN PEMBAHASAN

A. Implentasi Sistem

1. Arsitektur Sistem

Sistem autentikasi berbasis *device ID restriction* yang telah diimplementasikan menggunakan arsitektur *client-server* untuk memisahkan antara logika presentasi dan logika bisnis. Pemisahan ini memungkinkan pengembangan yang lebih terstruktur dan maintenance yang lebih mudah di masa mendatang.

Aplikasi desktop dibangun menggunakan *Electron* karena *framework* ini memberikan akses penuh ke sistem operasi untuk mendapatkan informasi hardware yang diperlukan dalam pembuatan *device ID*. *Electron* juga memungkinkan pengembangan aplikasi desktop *cross-platform* menggunakan teknologi web yang sudah familiar, sehingga mempercepat proses *development*.

Backend API dibangun menggunakan *Express.js* dengan arsitektur yang modular. Setiap komponen memiliki tanggung jawab yang *spesifik* - *controllers* menangani *request/response*, services mengimplementasikan *business logic*, dan *middleware* menangani *cross-cutting concerns* seperti autentikasi dan validasi. Pendekatan ini membuat kode lebih mudah ditest dan dimaintain.

2. Struktur *Database*

Database dirancang dengan mempertimbangkan normalisasi untuk menghindari redundansi data. *MySQL* dipilih sebagai *database management system* karena reliabilitas dan performanya yang telah teruji. *Prisma ORM* digunakan untuk mempermudah interaksi dengan *database* dan memberikan *type safety* pada level aplikasi, sehingga dapat mengurangi potensi kesalahan dalam pemanggilan data.

Struktur database pada penelitian ini disusun untuk mendukung proses autentikasi, pengelolaan token, serta pencatatan aktivitas pengguna. Salah

satu tabel utama yang digunakan adalah tabel User, yang menyimpan informasi dasar pengguna beserta metadata terkait perangkat dan aktivitas login. Rincian struktur tabel User ditunjukkan pada Tabel berikut

Tabel 2. *Schema database user*

Field	***Type***	***Constraint***	***Deskripsi***
id	*INT*	*PRIMARY KEY, AUTO_INCREMENT*	ID unik untuk setiap user
username	*VARCHAR*	*UNIQUE, NOT NULL*	Username untuk login
email	*VARCHAR*	*UNIQUE, NOT NULL*	Email user
password	*VARCHAR*	*NOT NULL*	Password terenkripsi
deviceId	*VARCHAR*	*NULL*	Device ID yang terdaftar
lastLogin	*DATETIME*	*NULL*	Waktu login terakhir
createdAt	*DATETIME*	*DEFAULT NOW()*	Waktu pembuatan akun
updatedAt	*DATETIME*	*AUTO UPDATE*	Waktu update terakhir

Password disimpan dalam bentuk hash menggunakan bcrypt dengan *cost factor* 10. Pemilihan bcrypt didasarkan pada ketahanannya terhadap *brute force attack* dan kemampuannya untuk menyesuaikan *computational cost* seiring dengan peningkatan kecepatan *hardware*.

Selain tabel User, sistem juga membutuhkan tabel Token untuk mengelola autentikasi pengguna secara aman. Tabel ini menyimpan informasi terkait token yang dihasilkan setelah proses login, termasuk keterkaitannya dengan perangkat tertentu, status validitas, serta waktu

kedaluwarsa token. Dengan adanya tabel ini, sistem dapat memverifikasi dan mengontrol akses pengguna agar hanya dapat digunakan pada perangkat yang sah. Rincian struktur tabel Token ditunjukkan pada Tabel berikut

Tabel 3. *Schema Database* Token

Field	***Type***	***Constraint***	**Deskripsi**
id	*INT*	*PRIMARY KEY, AUTO_INCREMENT*	ID unik untuk setiap user
Token	*VARCHAR*	*UNIQUE, NOT NULL*	JWT token string
UserId	*INT*	*FOREIGN KEY*	Referensi ke tabel User
deviceId	*VARCHAR*	*NOT NULL*	Device ID yang terkait
isValid	*BOOLEAN*	*DEFAULT TRUE*	Status validitas token
expiresAt	*DATATIME*	*NOT NULL*	Waktu kadaluarsa token
createdAt	*DATATIME*	*DEFAULT NOW()*	Waktu pembuatan token

Meskipun *JWT* bersifat stateless, implementasi ini menyimpan token dalam database untuk memungkinkan revokasi token secara *real-time* dan memberikan *audit trail* yang lengkap untuk keperluan *security monitoring*.

3. Implementasi token management

Sistem keamanan aplikasi ini menerapkan enkripsi berlapis untuk melindungi token autentikasi. Token yang digunakan adalah *JSON Web Token (JWT)* yang tidak hanya ditandatangani secara digital, tetapi juga dienkripsi pada berbagai tahap penyimpanan. Implementasi *enkripsi* ini

bertujuan untuk mencegah token disalahgunakan meskipun penyerang berhasil mengakses *database* atau *local storage* pengguna. Berikut adalah implementasi detail dari proses *enkripsi* dan *dekripsi* pada berbagai komponen sistem.

a. *Enkripsi* sebelum

   Proses enkripsi dilakukan sebelum token dikirimkan dari server ke aplikasi klien. Setelah kredensial pengguna berhasil diverifikasi, server membuat JSON Web Token (JWT) yang berisi userId, username, dan deviceId. Token ini kemudian dienkripsi dengan algoritma HS256 menggunakan secret key yang hanya diketahui server. Dengan demikian, sebelum token berpindah ke klien, data sudah dalam bentuk terenkripsi sehingga tidak bisa dimodifikasi oleh pihak ketiga selama proses transmisi.

b. *Enkripsi* sesudah

   Proses enkripsi juga diterapkan sesudah token diterima dan akan disimpan di sisi klien. Pada aplikasi berbasis Electron, token yang sudah valid tidak langsung ditaruh di localStorage (karena rawan diakses), tetapi disimpan dengan mekanisme aman seperti safeStorage atau file terenkripsi. Hal ini memastikan bahwa meskipun penyerang berhasil membuka penyimpanan lokal, token tetap tidak bisa digunakan tanpa proses dekripsi.

c. *Dekripsi* sebelum

   Dekripsi dilakukan sebelum token digunakan untuk mengakses resource server. Saat aplikasi klien hendak mengirim permintaan, token diambil dari penyimpanan aman dan didekripsi menggunakan kunci yang sesuai dengan deviceId. Proses ini memastikan bahwa token yang dipakai benar-benar berasal dari perangkat yang terdaftar, bukan hasil duplikasi dari perangkat lain.

d. *Dekripsi* sesudah

   Dekripsi terakhir dilakukan sesudah token diterima kembali oleh server pada setiap request. Server menjalankan fungsi jwt.verify untuk

memeriksa tanda tangan digital token, memastikan masa berlaku, dan mencocokkan deviceId yang terikat. Jika hasil dekripsi sesuai, server memberikan akses; jika tidak, akses ditolak. Dengan tahapan ini, keamanan terjamin baik dari sisi klien maupun server.

Untuk memperjelas proses enkripsi dan dekripsi yang telah dijelaskan pada poin a–d di atas, berikut disajikan tabel yang menampilkan perbandingan data sebelum enkripsi, setelah enkripsi, dan setelah dekripsi. Tabel ini bertujuan untuk memberikan visualisasi bagaimana data yang awalnya masih dalam bentuk asli (*plaintext*) diubah menjadi bentuk terenkripsi (*ciphertext*), kemudian dikembalikan lagi ke bentuk semula setelah proses dekripsi dilakukan.

Tabel 4. Proses sebelum *enkripsi* dan sesudah *dekripsi*

Sebelum Enkripsi	**Data Terenkripsi**	**Data Setelah Dekripsi**
userId: 101	8d7f3a9b2c…	userId: 101
username: iqram	4a1b9f0e7d…	username: iqram
email: iqram@gmail.com	9c6e1d4b7a…	email: iqram@gmail.com
deviceId: 2f4a7b9c1d	1d7e9c3b2a…	deviceId: 2f4a7b9c1d

Berdasarkan tabel di atas, dapat dilihat bahwa data seperti *userId*, *username*, *email*, *deviceId*, dan *loginTime* pada awalnya berada dalam bentuk asli (*plaintext*). Setelah melalui proses enkripsi, data tersebut diubah menjadi *ciphertext* berupa rangkaian karakter acak yang tidak dapat dibaca oleh pihak yang tidak berwenang. Hal ini memastikan bahwa meskipun data berhasil diakses oleh pihak ketiga, isinya tetap tidak dapat dipahami tanpa kunci dekripsi yang benar. Selanjutnya, melalui proses dekripsi, data yang terenkripsi dikembalikan ke bentuk semula tanpa mengalami perubahan sedikit pun. Hal ini membuktikan bahwa mekanisme enkripsi–dekripsi yang

diterapkan tidak hanya menjaga kerahasiaan data, tetapi juga mempertahankan integritas dan keasliannya.

4. Implementasi *frontend*

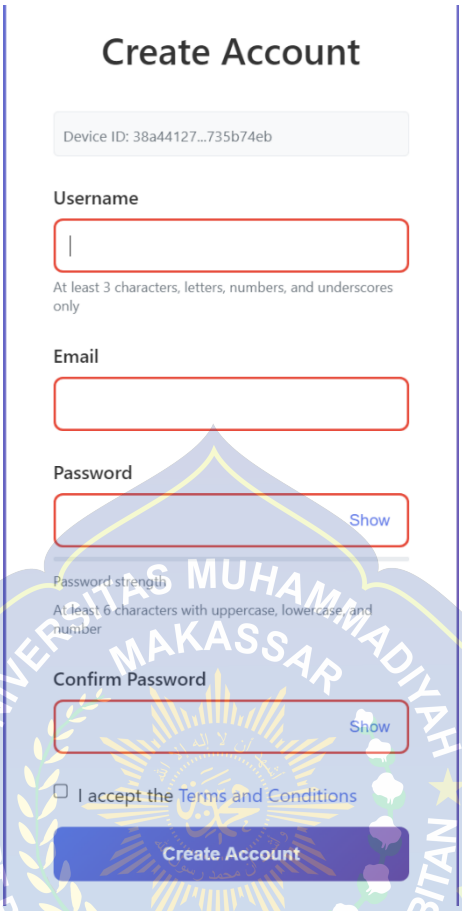
Frontend aplikasi dirancang dengan pendekatan modular dimana setiap halaman dan fungsi dipisahkan ke dalam file-file terpisah. Hal ini memudahkan pengembangan dan debugging karena setiap modul memiliki tanggung jawab yang jelas.

Struktur proyek *frontend* ini terdiri dari beberapa direktori utama yang tertata rapi untuk memisahkan fungsi sesuai prinsip *single responsibility*. Di dalam *folder pages* terdapat tiga halaman utama, yaitu *login.html*, *register.html*, dan *dashboard.html*. Folder css menyimpan file gaya terpisah untuk setiap halaman, seperti *common.css* untuk gaya umum, serta *login.css*, *register.css*, dan *dashboard.css* untuk kebutuhan spesifik masing-masing halaman. Folder js berisi modul *JavaScript* yang masing-masing memiliki tanggung jawab tunggal: *api.js* menangani komunikasi dengan *backend*, *device.js* bertugas menghasilkan *device ID*, *validators.js* berisi seluruh logika validasi input, sementara file lain seperti *auth.js*, *register.js*, *storage.js*, dan *dashboard.js* mengelola fungsi-fungsi khusus sesuai konteksnya.

B. Hasil Implementasi

1. Halaman Registrasi

Halaman registrasi merupakan salah satu komponen utama dari sistem, karena menjadi pintu masuk pertama bagi pengguna untuk membuat akun baru. Pada tahap ini, sistem tidak hanya berfokus pada kemudahan penggunaan (*usability*), tetapi juga mengutamakan aspek keamanan, khususnya dalam proses validasi data dan pengenalan perangkat. Tampilan antarmuka halaman registrasi ditunjukkan pada Gambar berikut



Gambar 4. Tampilan *register*

Proses registrasi dirancang untuk memberikan pengalaman yang smooth bagi user namun tetap mempertahankan keamanan yang tinggi. Validasi dilakukan secara *real-time* untuk memberikan *feedback immediate* kepada *user* saat mengisi form.

Saat *user* membuka halaman registrasi, sistem secara otomatis melakukan *generate device ID* di *background*. *Device ID* ini dibuat dengan mengkombinasikan berbagai informasi *hardware* seperti *MAC address*, *CPU model*, *hostname*, dan informasi sistem lainnya. Kombinasi ini kemudian di-hash menggunakan *SHA-256* untuk menghasilkan identifier yang unik.

Fitur-fitur yang diimplementasikan pada halaman registrasi:

a. *Real-time username availability check*
b. *Email format validation*

c. *Password strength indicator*

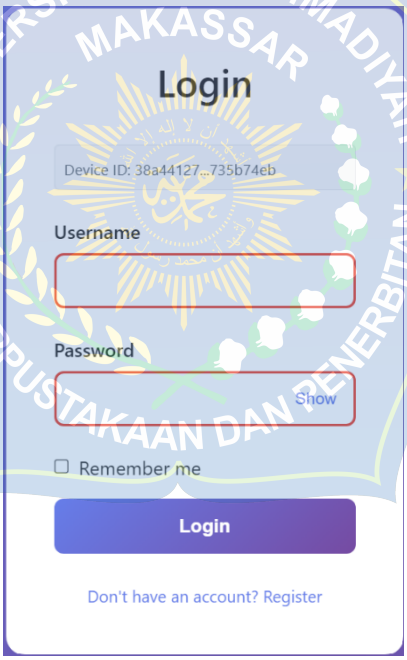
d. *Automatic device ID generation*

e. *Terms and conditions checkbox*

f. *Loading state saat submit*

2. Halaman *Login*

Halaman login merupakan bagian penting dari sistem yang berfungsi sebagai gerbang autentikasi bagi pengguna yang telah terdaftar. Pada tahap ini, sistem mengombinasikan dua mekanisme utama yaitu validasi kredensial (*username* dan *password*) serta verifikasi perangkat melalui *device ID*. Pendekatan ini memberikan lapisan keamanan tambahan tanpa menambah kompleksitas bagi pengguna. Tampilan antarmuka halaman login ditunjukkan pada Gambar berikut



Gambar 5. Tampilan *login*

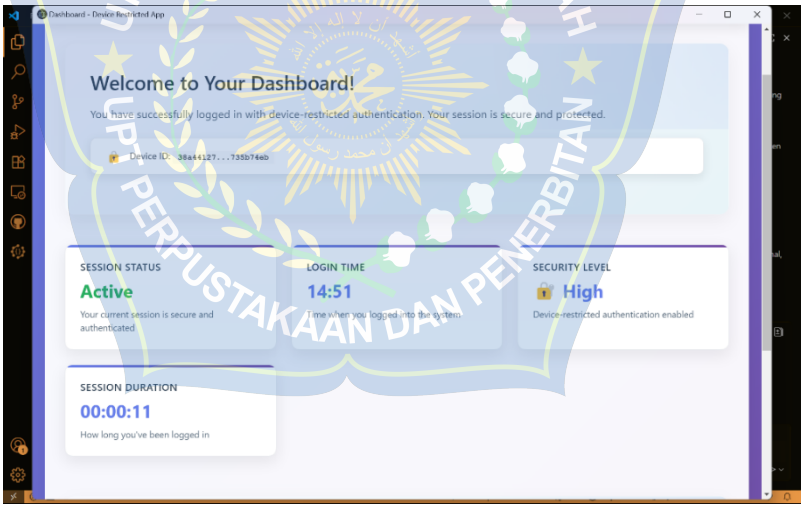
Saat *user* melakukan login, sistem tidak hanya memverifikasi *username* dan *password*, tetapi juga memastikan bahwa login dilakukan dari *device* yang telah terdaftar. Jika *user* mencoba *login* dari *device* yang berbeda, sistem akan menolak akses meskipun *credentials* yang dimasukkan benar.

Fitur keamanan pada halaman *login*:

a. *Rate limiting* untuk *prevent brute force*
b. *Device ID verification*
c. *Remember me functionality*
d. *Clear error messages*
e. *Auto redirect* jika sudah login

3. Halaman *Dashboard*

Setelah berhasil login, pengguna diarahkan ke halaman *Dashboard* yang menjadi pusat informasi terkait status sesi dan keamanan akun. Halaman ini tidak hanya berfungsi sebagai tampilan utama setelah autentikasi, tetapi juga sebagai sarana transparansi yang menunjukkan bagaimana sistem menjaga keamanan akses pengguna. Dengan menampilkan informasi *real-time*, pengguna dapat memantau sesi *login* sekaligus memastikan bahwa autentikasi berbasis perangkat berjalan sebagaimana mestinya.Tampilan halaman *Dashboard* ditunjukkan pada Gambar berikut



Gambar 6. Tampilan *dashboard*

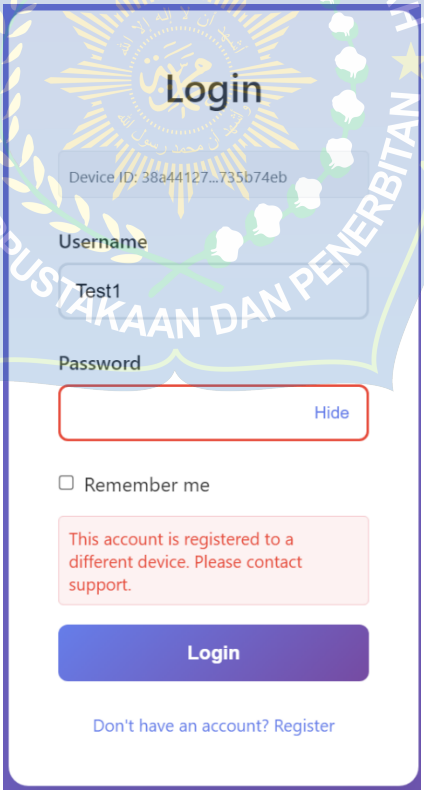
Session management menggunakan *JWT* token dengan *expiration time* 24 jam. Pemilihan durasi ini berdasarkan pertimbangan balance antara *security* dan *user convenience*. Token yang terlalu *short-lived* akan mengganggu *user experience*, sementara token yang terlalu *long-lived* meningkatkan *security risk*

Informasi yang ditampilkan:

a. *Username* dan *email*
b. *Device ID*
c. *Login time*
d. *Session duration (real-time)*
e. *Security status*
f. *Logout button*

4. *Device Restriction Implementation*

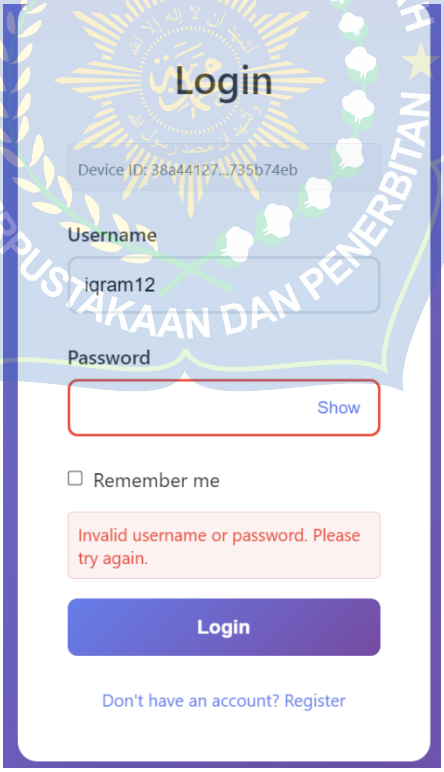
Untuk menguji efektivitas mekanisme *device restriction*, dilakukan simulasi *login* menggunakan perangkat yang berbeda dari yang telah terdaftar. Hasil pengujian menunjukkan bahwa sistem menolak akses meskipun kredensial yang dimasukkan benar. Hal ini menegaskan bahwa hanya perangkat yang sah yang dapat digunakan untuk mengakses akun. Tampilan ketika pengguna mencoba login dari perangkat yang berbeda ditunjukkan pada Gambar 5 berikut.



Gambar 7. *Login device* berbeda

Device ID generation menggunakan kombinasi dari berbagai *hardware identifier* untuk memastikan uniqueness. Proses ini melibatkan pengambilan informasi seperti *MAC address* dari *network interface* utama, *CPU* model, total *memory*, *hostname*, dan *platform information*. Semua informasi ini dikombinasikan dan di-hash menggunakan *SHA-256*.

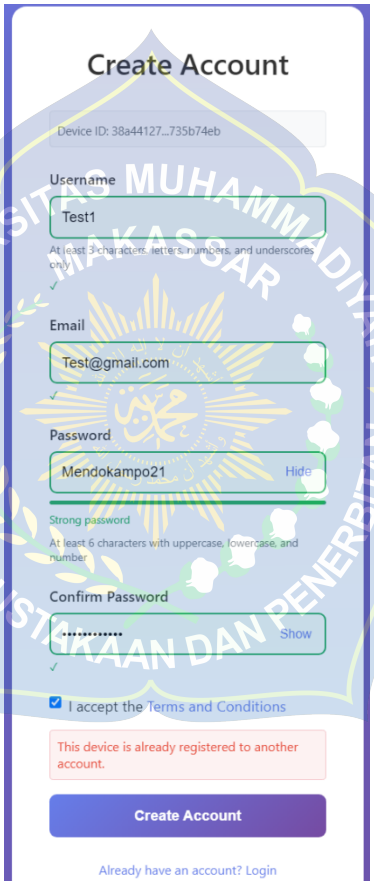
Selain proses identifikasi perangkat, sistem juga memperhatikan aspek keamanan pada tahap autentikasi pengguna. Untuk menguji keamanan pada proses autentikasi, sistem memberikan respon yang sesuai ketika pengguna memasukkan kredensial yang tidak valid. Hal ini bertujuan untuk memastikan bahwa login hanya bisa dilakukan dengan kombinasi *username* dan *password* yang benar, serta mencegah kebocoran informasi yang dapat dimanfaatkan oleh pihak yang tidak berwenang. Gambar berikut menampilkan kondisi saat pengguna salah memasukkan *username* atau *password*.



Gambar 8. *Kredensial* salah

Ketika user memasukan password dan user salah maka terlihat pesan error yang muncul. Pesan error yang generic ("Invalid username or password") digunakan untuk alasan keamanan agar tidak memberikan informasi spesifik kepada potential attacker.

Berikut adalah tampilan antarmuka proses registrasi pada aplikasi, yang menampilkan form untuk mengisi *username*, *email*, serta *password*. Pada bagian atas, sistem juga menampilkan *Device ID* yang secara otomatis dihasilkan dan terhubung dengan perangkat pengguna.



Gambar 9. *Error device already register*

Ketika user mencoba mendaftarkan akun baru dari device yang sudah terdaftar untuk akun lain, sistem menampilkan pesan error "This device is already registered to another account." seperti yang terlihat pada gambar di atas. Hal ini mencegah satu device digunakan untuk multiple accounts, meningkatkan accountability dan security.

C. Pengujian Sistem

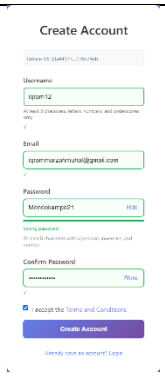
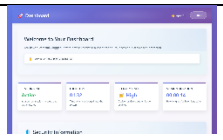
1. Pengujian *black box*

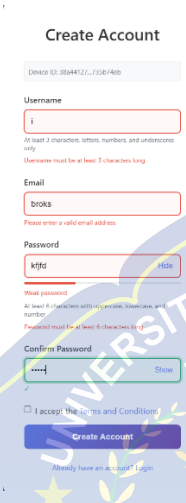
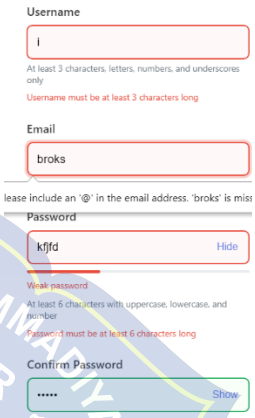
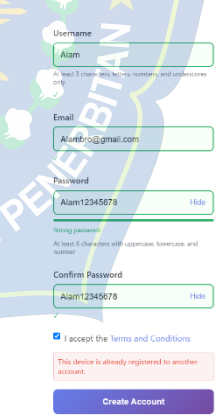
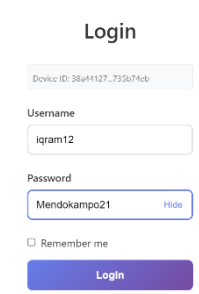
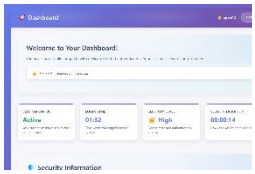
Pengujian *Black Box Testing* dilakukan untuk memastikan bahwa fungsionalitas utama sistem autentikasi berjalan sesuai dengan *requirement* yang telah ditentukan. Metode ini berfokus pada input dan output sistem tanpa memperhatikan proses internal atau kode program. Skenario pengujian dirancang untuk mencakup kondisi normal maupun kondisi kesalahan sehingga dapat menggambarkan perilaku sistem dalam penggunaan nyata.

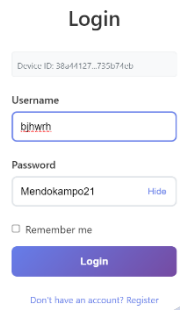
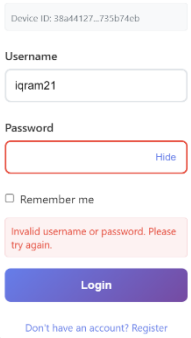
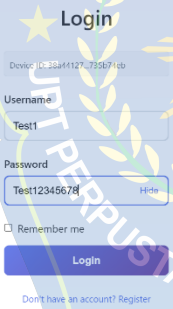
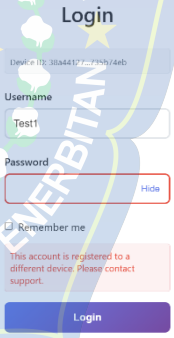
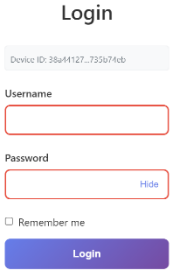
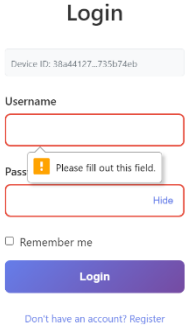
Pengujian difokuskan pada proses registrasi dan *login*. Pada tahap registrasi, diuji apakah sistem mampu menerima data yang valid dan menolak data yang tidak valid, termasuk menolak registrasi apabila perangkat yang digunakan sudah terdaftar untuk akun lain. Pada tahap *login*, sistem diuji untuk memastikan bahwa akses hanya diberikan jika kredensial dan perangkat sesuai, serta menolak *login* ketika *password* salah, perangkat berbeda, atau *field input* dikosongkan.

Berikut hasil pengujian *Black Box Testing*:

Tabel 5. Hasil pengujian *black box testing*

Fitur yang diuji	***Input* pengujian**	**Hasil yang di harapkan**	**Hasil pengujian**	**Status**
Registrasi akun baru		Registrasi berhasil, akun tersimpan di *database, device ID*		Berhasil

Fitur yang diuji	*Input* pengujian	Hasil yang di harapkan	Hasil pengujian	Status
		terikat pada akun		
Registrasi dengan data tidak valid		Sistem menolak registrasi, muncul pesan *error* sesuai validasi		Berhasil
Registrasi dengan *device* sudah terdaftar		Sistem menolak, pesan *error* *"Device is already registered to another account"*		Berhasil
Login valid		Login berhasil		Berhasil

Fitur yang diuji	*Input* pengujian	Hasil yang di harapkan	Hasil pengujian	Status
Login dengan *kredensial* salah		Sistem menolak, muncul pesan *error* *"Invalid username or password"*		Berhasil
Login dengan *device* berbeda		Sistem menolak, pesan *error* *"This account is registered to a different device"*		Berhasil
Login tanpa input (kosong)		Sistem menolak, field ditandai merah, login gagal		Berhasil

Hasil pengujian menunjukkan bahwa seluruh skenario Black Box Testing telah berjalan sesuai dengan yang diharapkan. Sistem mampu menerima input yang valid, menolak input yang tidak valid, membatasi penggunaan akun hanya pada perangkat terdaftar, serta memberikan pesan kesalahan yang sesuai untuk setiap kondisi error. Hal ini membuktikan bahwa requirement fungsional sistem autentikasi berbasis dekripsi token telah berhasil diimplementasikan dengan baik.

2. Pengujian keamanan

Pengujian keamanan dilakukan untuk memastikan bahwa sistem yang dibangun mampu bertahan terhadap berbagai *common attack vectors* yang sering digunakan penyerang. Fokus utama pengujian ini adalah memvalidasi efektivitas kontrol keamanan yang telah diterapkan, seperti proteksi terhadap serangan injeksi, validasi input, serta mekanisme perlindungan autentikasi. Setiap *security control* diuji secara langsung dengan mensimulasikan skenario serangan guna melihat sejauh mana sistem dapat memberikan respon dan menjaga integritas data.

Hasil pengujian keamanan yang diperoleh dirangkum dalam Tabel 5 berikut

Tabel 6. Hasil pengujian keamanan

Security Test	***Method***	***Result***	***Status***
SQL Injection	*Input malicious SQL*	*Prevented by Prisma ORM*	*Secure*
XSS Attack	*Input script tags*	*Input sanitized*	*Secure*
Brute Force	*Multiple login attempts*	*Rate limiting active*	*Secure*
Token Tampering	*Modify JWT token*	*Invalid signature detected*	*Secure*
Password Storage	*Check database*	*Bcrypt hashed*	*Secure*

Penggunaan *Prisma ORM* secara otomatis melindungi aplikasi dari *SQL injection* dengan menggunakan *parameterized queries*. Input validation dan sanitization di frontend dan backend mencegah *XSS attacks*. *Rate limiting effectively* mencegah *brute force attacks* tanpa mengganggu *legitimate users*.

3. Pengujian *cross – device*

Pengujian *cross-device* secara spesifik menvalidasi *core feature* dari sistem yaitu *device restriction*. Berbagai skenario ditest untuk memastikan tidak ada loophole yang memungkinkan *unauthorized access*. Beberapa skenario disusun untuk menguji perilaku sistem dalam berbagai kondisi penggunaan, mulai dari registrasi hingga percobaan login dari perangkat yang berbeda.

Rincian skenario pengujian *cross-device* ditunjukkan pada Tabel berikut:

Tabel 7. Skenario pengujian

Skenario	***Device A***	***Device B***	**Hasil**
Register on A, Login on A	Register berhasil	-	*Login Success*
Register on A, Login on B	Register berhasil	Login gagal	Ketidakcocokan perangkat
Register Different Account	Register berhasil	Register berhasil	*Both Active*

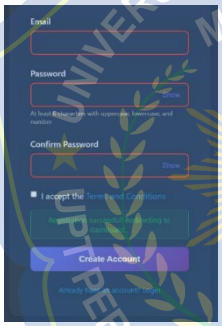
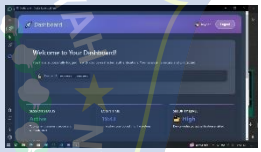
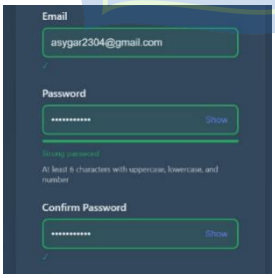
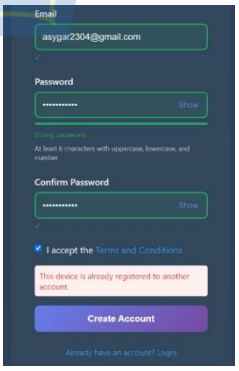
Hasil pengujian menunjukkan bahwa device restriction berfungsi dengan sempurna. Tidak ada *false positive* (*legitimate user* yang terblokir) maupun *false negative* (*unauthorized access* yang lolos) dalam semua skenario yang ditest.

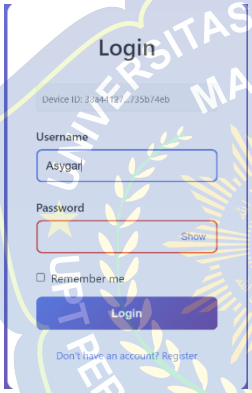
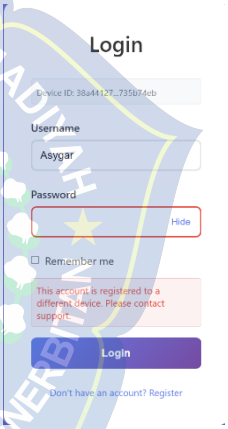
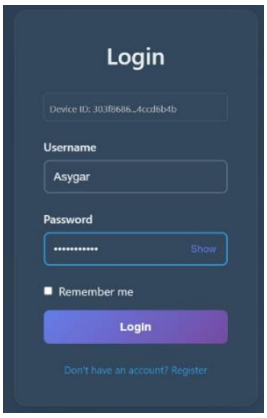
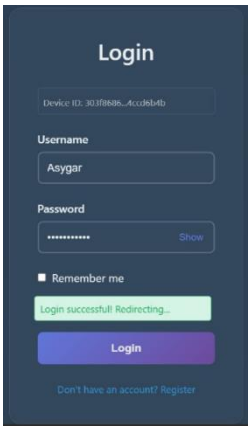
4. Pengujian pada Perangkat Berbeda

Untuk memperkuat hasil pengujian sistem autentikasi, dilakukan juga pengujian tambahan pada perangkat berbeda. Tujuan pengujian ini adalah

untuk membuktikan bahwa mekanisme device restriction benar-benar membatasi penggunaan akun hanya pada perangkat yang telah terdaftar. Pengujian dilakukan dengan beberapa skenario, yaitu registrasi akun baru, registrasi dengan perangkat yang sudah terdaftar, login dengan kredensial salah, login dari perangkat berbeda, serta login dengan kredensial valid pada perangkat yang sama. Dokumentasi hasil pengujian ditampilkan dalam tabel berikut.

Tabel 8. Pengujian perangkat berbeda

Skenario uji	**Input pengujian**	**Hasil yang Diharapkan**	**Hasil Pengujian**	**Status**
Registrasi akun baru		Registrasi berhasil, akun tersimpan, device ID terikat		berhasil
Registrasi akun dengan device yang sudah terdaftar		Sistem menolak, muncul pesan *"This device is already registered to another account"*		Berhasil

Skenario uji	Input pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
Login dengan kredensial salah		Sistem menolak, muncul pesan *"Invalid username or password"*		berhasil
Mencoba Login dari perangkat berbeda		Sistem menolak, muncul pesan *"This account is registered to a different device"*		Berhasil
Login dengan kredensial valid pada perangkat terdaftar		Login berhasil, diarahkan ke dashboard		Berhasil

Berdasarkan hasil pengujian, seluruh skenario berjalan sesuai harapan. Sistem menerima input valid saat registrasi akun baru, menolak registrasi pada perangkat yang sudah terdaftar, serta menghalangi login dengan password salah maupun dari perangkat berbeda. Sebaliknya, login hanya berhasil ketika kredensial valid digunakan pada perangkat yang benar.

5. Pengujian *System Usability Scale*

Selain pengujian teknis seperti black box testing, keamanan, dan cross-device, penelitian ini juga melakukan pengujian usability menggunakan metode *System Usability Scale (SUS)*. Kuesioner SUS terdiri dari 10 pernyataan yang difokuskan pada aspek keamanan login dan registrasi, meliputi kemudahan login dengan username dan password, pemahaman proses validasi perangkat (device binding), serta kepercayaan pengguna terhadap keamanan sistem.

Kuesioner diberikan kepada 5 responden, dengan skala Likert 1–5 (1 = Sangat Tidak Setuju, 5 = Sangat Setuju). Hasil pengisian dapat dilihat pada Tabel 4.xx berikut:

Tabel 9. Pengujian S.U.S

Pernyataan (SUS – Keamanan Login & Registrasi)	**R1**	**R2**	**R3**	**R4**	**R5**	**Skor**
Saya merasa proses login dengan username dan password pada aplikasi ini aman digunakan.	4	5	4	5	4	22
Saya merasa mekanisme login terlalu rumit karena adanya validasi perangkat.	2	1	2	1	2	8

Pernyataan (SUS – Keamanan Login & Registrasi)	R1	R2	R3	R4	R5	Skor
Saya merasa registrasi akun baru cukup mudah dilakukan.	5	4	5	4	5	23
Saya merasa memerlukan bantuan teknis untuk memahami proses registrasi dengan validasi perangkat.	2	2	1	2	1	8
Saya merasa fitur keamanan login terintegrasi dengan baik.	4	5	4	5	4	22
Saya merasa terdapat ketidakkonsistenan pada sistem keamanan login aplikasi ini.	1	2	1	2	1	7
Saya membayangkan pengguna lain akan cepat memahami cara registrasi dengan sistem keamanan ini.	5	4	5	4	5	23
Saya merasa mekanisme validasi perangkat saat login membingungkan.	2	1	2	1	2	8
Saya merasa percaya diri dan terlindungi saat login dengan sistem keamanan ini.	5	5	4	5	5	24
Saya merasa perlu belajar banyak sebelum bisa memahami cara kerja sistem login aman ini	2	2	1	2	1	8

Berdasarkan hasil pada Tabel 8, sistem login dan registrasi dinilai aman, mudah digunakan, serta mudah dipahami oleh pengguna, sehingga dapat disimpulkan memiliki tingkat usability yang baik.

BAB V
KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini dilakukan untuk menjawab tantangan keamanan terkait praktik berbagi akun pada aplikasi Guru Pintar, dengan merancang dan mengimplementasikan sebuah sistem dekripsi token yang mengikat sesi autentikasi secara eksklusif pada perangkat keras pengguna. Dari hasil implementasi dan pengujian sistem yang telah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Penelitian ini berhasil mengimplementasikan sistem dekripsi token dengan memanfaatkan mekanisme *device fingerprinting* pada aplikasi klien berbasis Electron. Sistem ini secara efektif menggabungkan berbagai parameter perangkat keras seperti *Machine ID*, informasi CPU, dan alamat MAC untuk menghasilkan sebuah identitas unik. Identitas unik ini kemudian menjadi kunci dalam proses enkripsi dan dekripsi token, yang memastikan bahwa token autentikasi yang diterbitkan hanya dapat divalidasi dan digunakan pada perangkat asal yang telah terdaftar. Kombinasi ini berhasil menciptakan sebuah lapisan keamanan yang secara fundamental mencegah token digunakan di perangkat lain.
2. Sistem yang dibangun terbukti efektif dalam menjamin keamanan dan keunikan akses akun pengguna, sehingga secara langsung mengatasi risiko penyalahgunaan yang timbul dari praktik berbagi akun. Melalui serangkaian skenario pengujian, sistem menunjukkan kemampuannya untuk menolak upaya akses dari perangkat yang tidak sah, bahkan ketika kredensial login yang benar digunakan. Dengan demikian, implementasi ini tidak hanya meningkatkan perlindungan data pengguna, tetapi juga menjaga integritas data personalisasi pembelajaran, sekaligus mempertahankan pengalaman pengguna yang nyaman dengan tidak memerlukan autentikasi tambahan berulang seperti OTP pada setiap sesi.

B. Saran

Penelitian ini berfokus pada implementasi device binding untuk mengamankan akses akun dalam lingkungan aplikasi desktop. Ke depan, sistem yang telah dibangun ini memiliki potensi besar untuk disempurnakan, salah satunya dengan merancang mekanisme migrasi perangkat (device migration) yang aman bagi pengguna. Fitur ini akan memungkinkan pengguna sah untuk mendaftarkan perangkat baru misalnya saat mengganti laptop tanpa harus kehilangan akses ke akun mereka. Proses ini dapat diimplementasikan melalui alur verifikasi multi-tahap, seperti konfirmasi melalui email yang terdaftar, untuk memastikan bahwa permintaan migrasi benar-benar berasal dari pemilik akun yang sah. Dengan adanya fitur ini, sistem tidak hanya akan menjadi lebih aman, tetapi juga lebih fleksibel dan adaptif terhadap kebutuhan nyata pengguna di lingkungan pendidikan.

DAFTAR PUSTAKA

Adigun, A A, O I Adigun, M O Abolarinwa, and O S Bakare. 2024. "Review of the Advanced Encryption Standard System Performance on Hidden Data." 10(2): 197–209.

Astowo, Unggul Budi, and Ari Sujarwo. 2023. "Penerapan JSON Web Token Sebagai Strategi Pengamanan Data Pada Aplikasi MultiMasjid." *Innovative: Journal Of Social Science Research* 3(6): 5279–92.

Darmawan, Irwan, Muhammad Umar Mansyur, Khana Zulfana Imam, Moh. Syahdan, and Ahmad Fawaid. 2023. "Evaluasi Keamanan Privilege Terintegerasi JSON Web Token Pada Sistem Informasi Akademik." *Jurnal Informasi dan Teknologi* 5(2): 120–28. doi:10.37034/jidt.v5i2.368.

Hafsah, Annisa, Muhammad Irwan, Padli Nasution, Fakultas Ekonomi, Dan Bisnis, Prodi Manajemen, Universitas Islam, and Negeri Sumatera. 2024. "Issn : 3025-9495." 10(9).

Halbouni, Asmaa, Lee Yeng Ong, and Meng Chew Leow. 2023. "Wireless Security Protocols WPA3: A Systematic Literature Review." *IEEE Access* 11: 112438–50. doi:10.1109/ACCESS.2023.3322931.

Hoshmand, Mohammad Omer, Suci Ratnawati, and Email Penulis Korespondensi. 2023. "Analisis Keamanan Infrastruktur Teknologi Informasi Dalam Menghadapi Ancaman Cybersecurity." *Jurnal Sains dan Teknologi* 5(2): 679–86. https://doi.org/10.55338/saintek.v5i2.2347.

Kudriani, Nadira, Fikri Murdana, and Larasati Muriati. 2023. "Transformasi Digital Dalam Pendidikan: Tantangan Dan Peluang Penerapan Kecerdasan Buatan Dalam Proses Pembelajaran." *Jurnal Literasi Digital* 3(3): 129–39. doi:10.54065/jld.3.3.2023.596.

Liu, Qinyi, and Mohammad Khalil. 2023. "Understanding Privacy and Data Protection Issues in Learning Analytics Using a Systematic Review." *British Journal of Educational Technology* 54(6): 1715–47. doi:10.1111/bjet.13388.

Loui Pattinama, Yohannes, FERDIANSYAH, Ika Susanti, and Painem. 2023. "Implementasi Rest API Web Service Dengan Otentifikasi JSON Web Token Untuk Aplikasi Properti." *Informatik : Jurnal Ilmu Komputer* 19(1): 81–89. doi:10.52958/iftk.v19i1.5724.

Olabanji, Samuel Oladiipo, Oluwaseun Oladeji Olaniyi, Chinasa Susan Adigwe, Olalekan Jamiu Okunleye, and Tunbosun Oyewale Oladoyinbo. 2024. "AI for Identity and Access Management (IAM) in the Cloud: Exploring the Potential of Artificial Intelligence to Improve User Authentication, Authorization, and Access Control within Cloud-Based Systems." *Asian Journal of Research in Computer Science* 17(3): 38–56. doi:10.9734/ajrcos/2024/v17i3423.

Pertiwi, Endah, Dzikra Delvina Nuraldini, Gilang Tri Buana, and Amos Arthacerses. 2022. "Analisis Yuridis Terhadap Penyalahgunaan Data Pribadi Pengguna Media Sosial." *Jurnal Rechten : Riset Hukum dan Hak Asasi Manusia* 3(3): 10–16. doi:10.52005/rechten.v3i3.65.

Rafli, Helmi, Raychan Zen, Ilyas Nuryasin, Prodi Informatika, Universitas Muhammadiyah Malang, Jl Raya, Tlogomas No, Kecamatan 246, and Kota Malang Lowokwaru. 2024. "JOISIE Licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0) PENERAPAN WHITEBOX TESTING PADA PENGUJIAN SISTEM MENGGUNAKAN TEKNIK BASIS PATH." *Journal Of Information Systems And Informatics Engineering* 8(1): 101–11. https://doi.org/10.35145/joisie.v8i1.4229.

Rahman, Rakhmadi, Nugrah Surya Pratama, Jurusan Sains, Program Studi, Sistem Informaso, Bumi Harapan, and Kota Parepare. 2024. "Peningkatan Keamanan Data Dengan Kriptografi Modern Pada Sistem Operasi." (4): 2–9.

Saputra, Hendra. 2023. "Utilization of JWT Tokens as an Authenticity Validation Method for Correspondence at Muhammadiyah University of East Kalimantan." 4(3).

Sitorus, Nico F, Ari Kusyanti, and Adhitya Bhawiyuga. 2020. "Implementasi

Autentikasi Berbasis Token Menggunakan Platform-Agnostic Security Tokens (PASETO) Sebagai Mekanisme Autentikansi RESTful API." *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer* 4(11): 3947–55. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/8184.

Sopingi, Faulinda Ely Nastiti, and Arvian Sofyan Majid. 2021. "Implementasi JSON Web Token Authentication Pada Aplikasi Pembayaran Berbasis Mobile." *Seminar Nasional & Call for Paper Hubisintek*: 343–51.

Uriawan, Wisnu, Ray Ramadita, Rizky Dwi Putra, Rizqi Ilham Siregar, and Risyad Addiva. 2024. "Authenticate and Verification Source Files Using SHA256 and HMAC Algorithms Authenticate and Verification Source Files Using." doi:10.20944/preprints202407.0075.v1.

Vivi Wahdini, Sri, Dedy Hartama, Ika Okta Kirana, Poningsih, and Sumarno. 2021. "Pengamanan Data Pelanggan Dan Penjualan Menggunakan Implementasi Algoritma Kriptografi." *Journal of Informatics Management and Information Technology* 1(3): 101–7.

Wijoyo, A, S Fatimah, and Y Widianti. 2023. "Keamanan Data Dalam Sistem Informasi Manajemen: Risiko Dan Strategi Perlindungan." *... : Jurnal Teknologi, Bisnis ...* 1(2): 1–7. http://jurnalmahasiswa.com/index.php/teknobis/article/view/441.

LAMPIRAN

Lampiran 1. Source Code

Auth.js

```
require('dotenv').config();

const authConfig = {
  jwt: {
    secret: process.env.JWT_SECRET || 'your-secret-key',
    expiresIn: process.env.JWT_EXPIRES_IN || '24h',
  },
  bcrypt: {
    rounds: parseInt(process.env.BCRYPT_ROUNDS) || 10,
  },
  rateLimit: {
   windowMs: parseInt(process.env.RATE_LIMIT_WINDOW_MS)
|| 15 * 60 * 1000,
    max: parseInt(process.env.RATE_LIMIT_MAX_REQUESTS)
|| 100,
  },
};

module.exports = authConfig;
```

authController.js

```
const authService = require('../services/authService');

const { validateRegistration, validateLogin,
handleValidationErrors } =
require('../utils/validators');

class AuthController {
  async register(req, res, next) {
    try {
```

```
      const { username, email, password, deviceId } =
req.body;
      const result = await authService.register({
        username,
        email,
        password,
        deviceId,
      });
      res.status(201).json({
        message: 'User registered successfully',
        user: result.user,
        token: result.token,
      });
    } catch (error) {
      next(error);
    }
  }
  async login(req, res, next) {
    try {
      const { username, password, deviceId } = req.body;

      const result = await authService.login({
        username,
        password,
        deviceId,
      });
      res.status(200).json({
        message: 'Login successful',
        user: result.user,
      token: result.token,
```

```
      });
    } catch (error) {
      next(error);
    }
  }
  async logout(req, res, next) {
    try {
       const token = req.headers.authorization?.split(' ')[1];
      if (!token) {
          return res.status(400).json({ error: 'Token required' });
      }
      await authService.logout(token);
      res.status(200).json({
        message: 'Logout successful',
      });
    } catch (error) {
      next(error);
    }
  }
  async forceLogout(req, res, next) {
    try {
      const userId = req.user.id;
      await authService.forceLogout(userId);
      res.status(200).json({
        message: 'Force logout successful',
      });
    } catch (error) {
      next(error);
```

```
    }
  }
  async verify(req, res, next) {
    try {
       const token = req.headers.authorization?.split('
')[1];
      if (!token) {
          return res.status(400).json({ error: 'Token
required' });
      }
                    const      result      =      await
authService.verifyToken(token);
      res.status(200).json(result);
    } catch (error) {
      next(error);
    }
  }
}
module.exports = new AuthController();
```

authMiddleware.js

```
const jwt = require('jsonwebtoken');
const { prisma } = require('../config/database');
const { verifyToken } = require('../utils/jwt');
const authenticateToken = async (req, res, next) => {
  try {
    const authHeader = req.headers['authorization'];
    const token = authHeader && authHeader.split(' ')[1];
    if (!token) {
      return res.status(401).json({ error: 'Access token
required' });
```

```
    }
    // Verify JWT token
    const decoded = verifyToken(token);
    // Check if token exists in database and is valid
    const tokenRecord = await prisma.token.findUnique({
      where: { token },
      include: { user: true },
    });
    if (!tokenRecord || !tokenRecord.isValid) {
      return res.status(401).json({ error: 'Invalid or expired token' });
    }
    // Check if token is expired
    if (new Date() > tokenRecord.expiresAt) {
      // Mark token as invalid
      await prisma.token.update({
        where: { id: tokenRecord.id },
        data: { isValid: false },
      });
      return res.status(401).json({ error: 'Token expired' });
    }
    req.user = tokenRecord.user;
    req.token = tokenRecord;
    next();
  } catch (error) {
    console.error('Authentication error:', error);
    return res.status(403).json({ error: 'Invalid token' });
```

```
  }
};
module.exports = { authenticateToken };
```

deviceMiddleware.js

```
const { prisma } = require('../config/database');
const validateDevice = async (req, res, next) => {
  try {
    const { deviceId } = req.body;
    const user = req.user;
    if (!deviceId) {
      return res.status(400).json({ error: 'Device ID is
required' });
    }
    // Check if user has a registered device
    if (user.deviceId && user.deviceId !== deviceId) {
      return res.status(403).json({
          error: 'Device mismatch. This account is
registered to a different device.',
        code: 'DEVICE_MISMATCH'
      });
    }
    next();
  } catch (error) {
    console.error('Device validation error:', error);
      return res.status(500).json({ error: 'Internal
server error' });
  }
};
```

```
const checkDeviceRegistration = async (req, res, next) => {
  try {
    const { deviceId } = req.body;
    // Check if device is already registered to another user
    const existingUser = await prisma.user.findFirst({
      where: {
        deviceId,
        NOT: { id: req.user ? req.user.id : undefined }
      }
    });
    if (existingUser) {
      return res.status(409).json({
        error: 'Device is already registered to another account',
        code: 'DEVICE_ALREADY_REGISTERED'
      });
    }
    next();
  } catch (error) {
     console.error('Device registration check error:', error);
      return res.status(500).json({ error: 'Internal server error' });
  }
};
module.exports = { validateDevice, checkDeviceRegistration };
```

deviceService.js

```
const { prisma } = require('../config/database');

class DeviceService {
  async registerDevice(userId, deviceId) {
    // Check if device is already registered to another
user
    const existingDevice = await prisma.user.findFirst({
      where: {
        deviceId,
        NOT: { id: userId }
      }
    });
    if (existingDevice) {
      throw new Error('Device is already registered to
another account');
    }
    // Update user's device ID
    const updatedUser = await prisma.user.update({
      where: { id: userId },
      data: { deviceId },
      select: {
        id: true,
        username: true,
        email: true,
        deviceId: true,
      }
    });
    return updatedUser;
  }
```

```
  async changeDevice(userId, newDeviceId) {
    // Check if new device is already registered to another user
    const existingDevice = await prisma.user.findFirst({
      where: {
        deviceId: newDeviceId,
        NOT: { id: userId }
      }
    });
    if (existingDevice) {
      throw new Error('Device is already registered to another account');
    }
    // Invalidate all existing tokens for this user
    await prisma.token.updateMany({
      where: { userId },
      data: { isValid: false }
    });
    // Update user's device ID
    const updatedUser = await prisma.user.update({
      where: { id: userId },
      data: { deviceId: newDeviceId },
      select: {
        id: true,
        username: true,
        email: true,
        deviceId: true,
      }
    });
```

```
    return updatedUser;
  }
  async validateDevice(userId, deviceId) {
    const user = await prisma.user.findUnique({
      where: { id: userId },
      select: { deviceId: true }
    });
    if (!user) {
      throw new Error('User not found');
    }
    if (user.deviceId && user.deviceId !== deviceId) {
      throw new Error('Device mismatch');
    }
    return true;
  }
  async getDeviceInfo(userId) {
    const user = await prisma.user.findUnique({
      where: { id: userId },
      select: {
        deviceId: true,
        lastLogin: true,
        tokens: {
          where: { isValid: true },
          select: {
            deviceId: true,
            createdAt: true,
            expiresAt: true,
          }
```

```
        }
      }
    });
    return user;
  }
}
module.exports = new DeviceService();
```

tokenService.js

```
const { prisma } = require('../config/database');
const { generateToken } = require('../utils/jwt');
class TokenService {
  async createToken(userId, deviceId) {
    const token = generateToken({
      userId,
      deviceId,
    });
    const expiresAt = new Date(Date.now() + 24 * 60 * 60
* 1000); // 24 hours
    await prisma.token.create({
      data: {
        token,
        userId,
        deviceId,
        expiresAt,
      }
    });
    return token;
  }
```

```
async invalidateToken(token) {
  await prisma.token.updateMany({
    where: { token },
    data: { isValid: false }
  });
}
async invalidateAllUserTokens(userId) {
  await prisma.token.updateMany({
    where: { userId },
    data: { isValid: false }
  });
}
async cleanupExpiredTokens() {
  const result = await prisma.token.deleteMany({
    where: {
      expiresAt: {
        lt: new Date()
      }
    }
  });
  return result.count;
}
async getActiveTokens(userId) {
  return await prisma.token.findMany({
    where: {
      userId,
      isValid: true,
      expiresAt: {
```

```
            gt: new Date()
          }
        },
        select: {
          id: true,
          deviceId: true,
          createdAt: true,
          expiresAt: true,
        }
      });
    }
}
module.exports = new TokenService();
```

JWT.js

```
const jwt = require('jsonwebtoken');
const authConfig = require('../config/auth');
const generateToken = (payload) => {
  return jwt.sign(payload, authConfig.jwt.secret, {
    expiresIn: authConfig.jwt.expiresIn,
  });
};
const verifyToken = (token) => {
  try {
    return jwt.verify(token, authConfig.jwt.secret);
  } catch (error) {
    throw new Error('Invalid token');
  }
```

```
};
const decodeToken = (token) => {
  return jwt.decode(token);
};
module.exports = {
  generateToken,
  verifyToken,
  decodeToken,
};
```

App.js

```
const express = require('express');
const cors = require('cors');
const helmet = require('helmet');
const rateLimit = require('express-rate-limit');
require('dotenv').config();
const authRoutes = require('./routes/authRoutes');
const userRoutes = require('./routes/userRoutes');
const errorMiddleware = require('./middleware/errorMiddleware');
const app = express();
// Security middleware
app.use(helmet());
app.use(cors());

// Rate limiting
const limiter = rateLimit({
  windowMs: parseInt(process.env.RATE_LIMIT_WINDOW_MS) || 15 * 60 * 1000, // 15 minutes
```

```
  max: parseInt(process.env.RATE_LIMIT_MAX_REQUESTS) ||
100, // limit each IP to 100 requests per windowMs
  message: 'Too many requests from this IP, please try
again later.',
});
app.use(limiter);
// Body parsing middleware
app.use(express.json({ limit: '10mb' }));
app.use(express.urlencoded({ extended: true }));
// Routes
app.use('/api/auth', authRoutes);
app.use('/api/user', userRoutes);
// Health check endpoint
app.get('/health', (req, res) => {
   res.status(200).json({ status: 'OK', timestamp: new
Date().toISOString() });
});
// Error handling middleware
app.use(errorMiddleware);
// Handle 404
app.use('*', (req, res) => {
  res.status(404).json({ error: 'Route not found' });
});
module.exports = app;
```

Server.js

```
const app = require('./src/app');
const { PrismaClient } = require('@prisma/client');
const prisma = new PrismaClient();
```

```
const PORT = process.env.PORT || 3000;
async function startServer() {
  try {
    // Test database connection
    await prisma.$connect();
    console.log('Database connected successfully');
    // Start server
    app.listen(PORT, () => {
      console.log(`Server is running on port ${PORT}`);
                                  console.log(`Environment:
${process.env.NODE_ENV}`);
    });
  } catch (error) {
    console.error('Failed to start server:', error);
    process.exit(1);
  }
}
// Graceful shutdown
process.on('SIGINT', async () => {
  console.log('\nShutting down gracefully...');
  await prisma.$disconnect();
  process.exit(0);
});
startServer();
```

Lampiran 2. Surat keterangan bebas plagiat

MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN
Alamat kantor: Jl.Sultan Alauddin NO.259 Makassar 90221 Tlp.(0411) 866972,881593, Fax.(0411) 865588

بسم الله الرحمن الرحيم

SURAT KETERANGAN BEBAS PLAGIAT

UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:

Nama : Muh Al Iqram Marzah
Nim : 105841105121
Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9%	10 %
2	Bab 2	11%	25 %
3	Bab 3	7%	10 %
4	Bab 4	5%	10 %
5	Bab 5	4%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan seperlunya.

Makassar, 26 Agustus 2025
Mengetahui.
Kepala UPT- Perpustakaan dan Pernerbitan,

Nursinah, S.Hum., M.I.P
NBM. 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972,881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Lampiran 3. Hasil Turnitin

Bab I MUH. AL IQRAM MARZAH 105841105121

by Tahap Tutup

Submission date: 25-Aug-2025 03:03PM (UTC+0700)
Submission ID: 2734871434
File name: BAB_1_MUH._AL_IQRAM_MARZAH.docx (34.7K)
Word count: 891
Character count: 5962

Bab I MUH. AL IQRAM MARZAH 105841105121

ORIGINALITY REPORT

9%	9%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.ums.ac.id Internet Source	4%
2	text-id.123dok.com Internet Source	2%
3	repository.pnj.ac.id Internet Source	2%
4	repo.stimata.ac.id Internet Source	2%

Exclude quotes	Off	Exclude matches	<2%
Exclude bibliography	Off		

Bab II MUH. AL IQRAM MARZAH 105841105121

by Tahap Tutup

Submission date: 25-Aug-2025 03:04PM (UTC+0700)
Submission ID: 2734871860
File name: BAB_2_MUH._AL_IQRAM_MARZAH.docx (73.92K)
Word count: 2020
Character count: 13673

Bab II MUH. AL IQRAM MARZAH 105841105121

ORIGINALITY REPORT

11%	10%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	j-innovative.org Internet Source	2%
2	Submitted to IAIN Bengkulu Student Paper	1%
3	ojs.udb.ac.id Internet Source	1%
4	hadfex.com Internet Source	1%
5	jurnalmahasiswa.com Internet Source	1%
6	ejournal.upnvj.ac.id Internet Source	1%
7	docplayer.info Internet Source	1%
8	zephyrnet.com Internet Source	1%
9	journal.widyatama.ac.id Internet Source	1%
10	Irham Mu'alimin Arrijal, Rusdi Efendi, Boko Susilo. "PENERAPAN ALGORITMA KRIPTOGRAFI KUNCI SIMETRIS DENGAN MODIFIKASI VIGENERE CIPHER DALAM APLIKASI KRIPTOGRAFI TEKS", Pseudocode, 2016	<1%

	Publication	
11	jtiik.ub.ac.id Internet Source	<1 %
12	indobot.co.id Internet Source	<1 %
13	jwcn-eurasipjournals.springeropen.com Internet Source	<1 %
14	must-august.blogspot.com Internet Source	<1 %
15	www.slideshare.net Internet Source	<1 %
16	Agustinus Bayu Prasetya, Imam Suharjo. "BACKEND API DATA PROTECTION MENGGUNAKAN JWT TOKEN DAN ALGORITMA AES 256-BIT DENGAN BAHASA PEMROGRAMAN GOLANG", Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Bab III MUH. AL IQRAM MARZAH 105841105121

by Tahap Tutup

Submission date: 25-Aug-2025 03:05PM (UTC+0700)
Submission ID: 2734872146
File name: BAB_3_MUH._AL_IQRAM_MARZAH.docx (79.16K)
Word count: 1480
Character count: 9906

Bab III MUH. AL IQRAM MARZAH 105841105121

ORIGINALITY REPORT

7%	6%	2%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universitas Muhammadiyah Makassar Student Paper	3%
2	Submitted to STKIP Sumatera Barat Student Paper	1%
3	docplayer.info Internet Source	1%
4	repositori.uin-alauddin.ac.id Internet Source	1%
5	www.globeindia.org Internet Source	1%
6	www.slideshare.net Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Bab IV MUH. AL IQRAM MARZAH 105841105121

by Tahap Tutup

Submission date: 25-Aug-2025 03:06PM (UTC+0700)
Submission ID: 2734872388
File name: BAB_4_MUH._AL_IQRAM_MARZAH.docx (1.04M)
Word count: 2114
Character count: 14172

Bab IV MUH. AL IQRAM MARZAH 105841105121

ORIGINALITY REPORT

5%	2%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.slideshare.net Internet Source	1%
2	Submitted to SOS Hermann Gmainer International College Student Paper	1%
3	Submitted to UNIVERSITAS BUDI LUHUR Student Paper	1%
4	Robis Fahma Yoga, Yovi Litanianda, Ghulam Asrofi Buntoro. "Pengembangan Sistem Pendukung Keputusan Berbasis SAW untuk Rekomendasi Pemilihan Motor Bekas", bit-Tech, 2025 Publication	1%
5	www.journal.unpas.ac.id Internet Source	1%
6	Agim Aljanata Azwar, Alda Cendekia Siregar, Barry Ceasar Octariadi. "IMPLEMENTASI PROMETHEE DALAM MENENTUKAN PRIORITAS PERBAIKAN SARANA DAN PRASARANA SEKOLAH DASAR DI KOTA PONTIANAK", VOX EDUKASI: Jurnal Ilmiah Ilmu Pendidikan, 2025 Publication	<1%
7	mafiadoc.com	

	Internet Source	<1%
8	Muhamad Nasihin, Rabiatul Adwiya. "SISTEM INFORMASI PENGADUAN MASYARAKAT KECAMATAN PONTIANAK SELATAN", CYBERNETICS, 2018 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Bab V MUH. AL IQRAM MARZAH 105841105121

by Tahap Tutup

Submission date: 25-Aug-2025 03:07PM (UTC+0700)
Submission ID: 2734872556
File name: BAB_5_MUH._AL_IQRAM_MARZAH.docx (27.08K)
Word count: 336
Character count: 2270

Bab V MUH. AL IQRAM MARZAH 105841105121

ORIGINALITY REPORT

4%	4%	2%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	id.scribd.com Internet Source	4%

Exclude quotes	Off	Exclude matches	< 2%
Exclude bibliography	Off		