Implementasi E2EE Kriptografi dalam Glutara: *IoT* Manajemen Diabetes

Austin Gabriel Pardosi - 13521084
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): gabrielpardosi26@gmail.com

***Abstract*—Makalah ini membahas penerapan kriptografi dalam Glutara, sebuah solusi IoT inovatif untuk manajemen diabetes yang menggabungkan *continuous glucose monitoring* (CGM) non-invasif dengan visualisasi data *real-time* dan analisis AI untuk memberikan wawasan nutrisi yang mendalam. Dalam sistem ini, data pengguna yang sangat sensitif dienkripsi menggunakan berbagai teknik kriptografi untuk memastikan keamanan dan privasi informasi dari perangkat IoT hingga ke *backend* dan aplikasi *mobile*. Glutara juga memanfaatkan enkripsi *end-to-end* (E2EE) untuk melindungi komunikasi antara *backend* dan modul AI, yang bertugas melakukan analisis data glukosa dan pengenalan makanan. Selain itu, makalah ini memberikan rekomendasi penerapan enkripsi di berbagai titik kritis dalam alur data, termasuk dari Arduino ke *database*, dari *database* ke *backend*, dan dari *backend* ke modul AI, menggunakan teknologi seperti *Elliptic Curve Cryptography* (ECC), fungsi *hash*, dan tanda tangan digital. Dengan penerapan kriptografi yang komprehensif ini, Glutara memastikan bahwa data kesehatan pengguna terlindungi secara optimal, memberikan rasa aman dalam manajemen diabetes sehari-hari.**

Keywords—Kriptografi; Encryption; IoT; Manajemen Diabetes; Keamanan Data; Glutara

I. PENDAHULUAN

Diabetes adalah penyakit kronis yang mempengaruhi jutaan orang di seluruh dunia. Manajemen diabetes memerlukan pemantauan kadar glukosa darah secara rutin, yang sering kali menjadi beban bagi penderita. Metode tradisional pemantauan glukosa, seperti *finger-pricking*, tidak hanya menyakitkan tetapi juga menghambat pemantauan yang sering dan konsisten. Oleh karena itu, kebutuhan akan solusi non-invasif yang dapat memberikan pemantauan glukosa *real-time* menjadi sangat mendesak.

Glutara hadir sebagai solusi inovatif yang menggabungkan teknologi IoT dan AI untuk manajemen diabetes yang lebih efektif dan nyaman. Sistem ini menggunakan *continuous glucose monitoring* (CGM) non-invasif yang terhubung dengan perangkat Arduino dan ESP32 untuk mengukur kadar glukosa darah secara *real-time*. Data yang dikumpulkan kemudian dikirim ke *backend* untuk penyimpanan dan analisis lebih lanjut. Selain itu, Glutara menyediakan aplikasi *mobile* yang memungkinkan pengguna untuk memantau kadar glukosa mereka dengan mudah dan menerima rekomendasi nutrisi berdasarkan data yang diolah oleh AI.

Namun, dengan meningkatnya jumlah data sensitif yang dikumpulkan dan diproses, keamanan dan privasi data menjadi perhatian utama. Penerapan kriptografi dalam Glutara bertujuan untuk melindungi data pengguna dari akses tidak sah dan memastikan integritas informasi. Enkripsi *end-to-end* (E2EE) dan teknik kriptografi lainnya digunakan untuk mengamankan data dari perangkat IoT hingga ke aplikasi mobile. Ini memastikan bahwa hanya pihak yang berwenang yang dapat mengakses dan memproses data glukosa yang dikumpulkan.

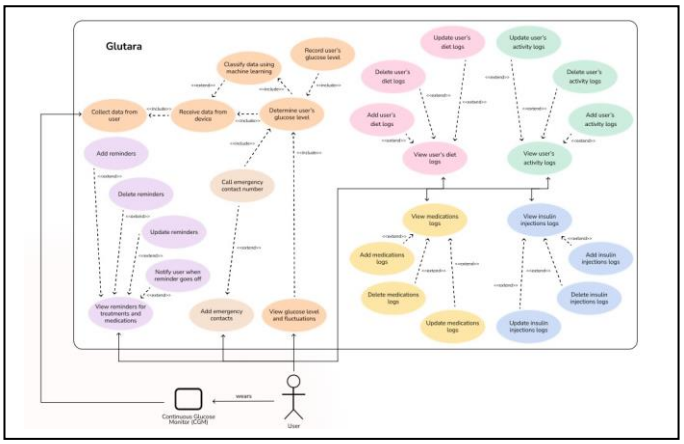
Penerapan kriptografi dalam Glutara tidak hanya terbatas pada transmisi data tetapi juga mencakup penyimpanan dan analisis data. Data yang disimpan di *database* dienkripsi untuk mencegah akses tidak sah. Selain itu, data yang dikirim ke modul AI untuk analisis lebih lanjut juga dienkripsi untuk memastikan keamanan selama proses pemrosesan. Implementasi teknik seperti *Elliptic Curve Cryptography* (ECC), fungsi *hash*, dan *digital signature* membantu dalam mencapai tujuan ini.

Dalam makalah ini akan dijelaskan lebih lanjut mengenai arsitektur sistem Glutara dan bagaimana kriptografi diterapkan pada berbagai level untuk meningkatkan keamanan dan privasi data. Kami juga akan memberikan rekomendasi spesifik untuk penerapan kriptografi pada setiap tahap alur data, mulai dari perangkat Arduino hingga modul AI. Dengan penerapan kriptografi yang tepat, Glutara dapat menjadi solusi yang aman dan andal untuk manajemen diabetes yang efektif.

II. LANDASAN TEORI

A. Glutara

Glutara adalah sebuah solusi IoT (*Internet of Things*) yang dirancang untuk manajemen diabetes secara efektif dan efisien. Sistem ini menggabungkan teknologi *Continuous Glucose Monitor* (CGM) non-invasif dengan perangkat Arduino dan ESP32 untuk mengumpulkan data glukosa dari pengguna secara *real-time*. Data yang dikumpulkan kemudian dikirim ke *backend* untuk penyimpanan, analisis, dan penyediaan rekomendasi nutrisi serta pengingat perawatan. Glutara memanfaatkan *Artificial Intelligence* (AI) untuk menganalisis data dan memberikan wawasan yang berguna bagi pengguna, termasuk klasifikasi data menggunakan *machine learning* dan pemberian informasi nutrisi berdasarkan pengenalan makanan.

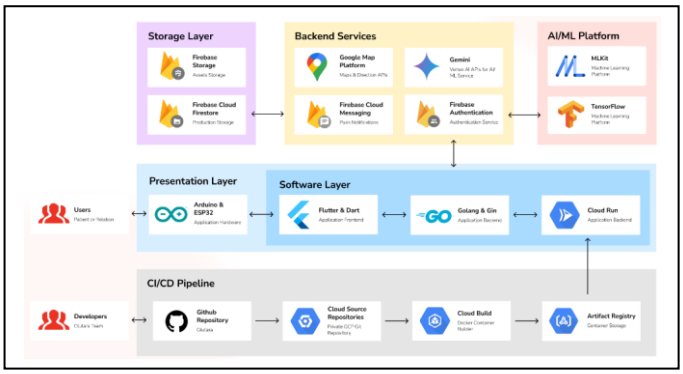


Gambar 2.1. Diagram *use-case* Glutara.

Diagram *use-case* di atas menggambarkan berbagai fungsi dan alur kerja yang dapat dilakukan oleh Glutara. Diagram ini menunjukkan bagaimana data dikumpulkan dari pengguna, diproses, dan digunakan untuk memberikan berbagai layanan kepada pengguna. Glutara mengumpulkan data glukosa dari pengguna menggunakan perangkat CGM, yang kemudian diterima oleh perangkat dan dikirim ke *backend* untuk diproses lebih lanjut. Data glukosa yang diterima diproses untuk menentukan level glukosa pengguna, dan machine learning digunakan untuk mengklasifikasikan data serta memberikan analisis yang lebih mendalam.

Selain itu, Glutara juga mengelola pengingat terkait perawatan dan pengobatan, di mana pengguna dapat menambahkan, menghapus, dan memperbarui pengingat. Sistem akan memberi tahu pengguna ketika pengingat aktif. Manajemen kontak darurat juga menjadi bagian dari fungsi Glutara, memungkinkan pengguna untuk menambahkan kontak darurat yang akan dihubungi dalam situasi darurat berdasarkan data glukosa pengguna. Pengguna juga dapat menambahkan, menghapus, dan memperbarui log aktivitas dan diet mereka, yang membantu dalam memantau aktivitas fisik dan asupan makanan yang berpengaruh terhadap level glukosa.

Dalam hal pengelolaan pengobatan, pengguna dapat menambahkan, menghapus, dan memperbarui log pengobatan mereka. Sistem membantu dalam memantau jadwal dan dosis pengobatan yang diambil oleh pengguna. Selain itu, pengguna dapat mencatat suntikan insulin, termasuk dosis dan waktu, yang membantu dalam memantau penggunaan insulin dan menyesuaikan perawatan sesuai kebutuhan. Dengan demikian, Glutara menyediakan berbagai layanan yang mendukung manajemen diabetes secara komprehensif dan terintegrasi, memastikan pengguna dapat mengelola kondisi mereka dengan lebih efektif dan efisien.



Gambar 2.2. Diagram *system architecture* Glutara.

Arsitektur sistem Glutara dirancang untuk mengintegrasikan berbagai komponen teknologi yang bekerja bersama untuk memberikan solusi manajemen diabetes yang komprehensif. Pada lapisan penyimpanan, *Firebase Storage* digunakan untuk menyimpan aset-aset aplikasi seperti gambar dan file lainnya, sementara *Firebase Cloud Firestore* digunakan untuk penyimpanan data produksi, termasuk data glukosa pengguna. Ini memastikan bahwa semua data pengguna disimpan dengan aman dan dapat diakses dengan cepat saat diperlukan.

Layanan *backend* mencakup beberapa komponen penting. *Google Map Platform* menyediakan layanan peta dan arah untuk membantu pengguna menemukan lokasi pengguna dikala keadaan darurat terjadi. *Gemini*, layanan *Vertex AI* dari *Google*, digunakan untuk analisis data AI/ML, yang memungkinkan sistem untuk memproses gambar makanan dan memberikan wawasan nutrisi terkandung di dalamnya. *Firebase Cloud Messaging* digunakan untuk mengirim *push notification* kepada pengguna, sehingga mereka selalu mendapatkan informasi terbaru mengenai kondisi kesehatan mereka. *Firebase Authentication* memastikan bahwa proses autentikasi pengguna berjalan dengan aman, melindungi data pribadi dan kesehatan pengguna dari akses yang tidak sah.

Lapisan AI/ML menggunakan *MLKit* dan *TensorFlow* untuk menjalankan model *machine learning* yang menganalisis data glukosa dan menyediakan berbagai fungsionalitas berbasis AI. *MLKit* digunakan untuk tugas-tugas machine learning yang lebih sederhana dan cepat, seperti scan QR untuk berbagi kontak antar pengguna di dalam aplikasi. Sementara itu, *TensorFlow* digunakan untuk tugas-tugas yang lebih kompleks seperti analisis tren glukosa jangka panjang dan prediksi kondisi kesehatan pengguna berdasarkan data yang dikumpulkan.

Pada *presentation layer*, perangkat Arduino dan ESP32 digunakan sebagai *hardware* aplikasi yang mengumpulkan data glukosa dari pengguna secara real-time. Aplikasi *mobile* dikembangkan menggunakan Flutter dengan Dart, yang menyediakan antarmuka pengguna yang intuitif dan *user-friendly*. *Backend* aplikasi dikembangkan menggunakan *Golang* dengan *framework Gin*, yang memastikan aplikasi berjalan dengan efisien dan dapat diandalkan. *Cloud Run*

digunakan untuk menyediakan lingkungan eksekusi yang *scalable* bagi aplikasi *backend.*

Pipeline CI/CD memastikan bahwa proses pengembangan dan pengiriman kode berjalan dengan lancar dan aman. *Repositori GitHub* digunakan untuk manajemen versi kode sumber, sementara *Cloud Source Repositories* dari *GCP* menyediakan *repositori Git* privat yang aman untuk kode sumber aplikasi. *Cloud Build* digunakan untuk membangun *container Docker* secara otomatis, yang kemudian disimpan di *Artifact Registry* sebelum di-*deploy* ke *Cloud Run*. Proses ini memastikan bahwa setiap perubahan kode dapat diintegrasikan, diuji, dan di-*deploy* dengan cepat dan tanpa risiko terhadap keamanan dan integritas sistem.

B. Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) adalah salah satu teknik kriptografi kunci publik yang menawarkan keamanan tinggi dengan ukuran kunci yang lebih kecil dibandingkan dengan algoritma tradisional seperti RSA. ECC bekerja berdasarkan sifat matematika dari kurva eliptik untuk menghasilkan pasangan kunci publik dan privat. Ukuran kunci yang lebih kecil ini berarti ECC dapat menyediakan tingkat keamanan yang sama dengan RSA tetapi menggunakan kunci yang jauh lebih pendek, sehingga mengurangi beban komputasi dan kebutuhan penyimpanan. Keuntungan ini sangat penting dalam lingkungan yang memiliki sumber daya terbatas seperti perangkat IoT, termasuk Arduino yang digunakan dalam sistem Glutara. ECC memungkinkan perangkat untuk melakukan operasi kriptografi dengan efisien tanpa mengorbankan keamanan, membuatnya ideal untuk aplikasi yang memerlukan enkripsi data secara *real-time*..

ECC dapat digunakan untuk mengenkripsi data glukosa yang dikirim dari perangkat Arduino ke *database*. Dengan menggunakan ECC, data glukosa yang dikumpulkan oleh sensor CGM dapat dienkripsi sebelum dikirim melalui jaringan. Ini memastikan bahwa data tersebut terlindungi dari penyadapan dan akses tidak sah selama transmisi. Enkripsi menggunakan ECC tidak hanya melindungi privasi pengguna tetapi juga memastikan integritas data sehingga data yang sampai di *database* adalah data asli yang tidak dimodifikasi. Implementasi ECC juga mendukung autentikasi perangkat, memastikan bahwa data yang diterima oleh *backend* memang berasal dari perangkat yang sah. Dengan demikian, ECC memainkan peran penting dalam menjaga keamanan dan keandalan komunikasi data dalam ekosistem Glutara.

C. Digital Signature

Digital Signature adalah metode kriptografi yang digunakan untuk memverifikasi keaslian dan integritas pesan atau dokumen digital. Metode ini memanfaatkan pasangan kunci kriptografi: kunci privat untuk menandatangani data dan kunci publik untuk memverifikasinya. Ketika data ditandatangani secara digital, *hash* dari data tersebut dienkripsi menggunakan kunci privat pengirim, menciptakan tanda tangan digital yang unik untuk setiap pesan. Penerima kemudian dapat menggunakan kunci publik pengirim untuk mendekripsi *hash* tersebut dan membandingkannya dengan *hash* data yang diterima. Jika *hash* cocok, keaslian dan integritas data terjamin, membuktikan bahwa data tidak diubah selama transmisi dan benar-benar berasal dari sumber yang diklaim.

Tanda tangan digital diterapkan untuk memastikan bahwa data yang diterima oleh *backend* berasal dari sumber yang valid dan tidak dimanipulasi. Setiap kali perangkat Arduino mengirimkan data glukosa ke *backend*, data tersebut ditandatangani dengan kunci privat perangkat. *Backend* kemudian menggunakan kunci publik yang terkait untuk memverifikasi tanda tangan digital ini. Dengan demikian, kita dapat mencegah serangan seperti *man-in-the-middle*, di mana data dapat disadap dan dimodifikasi oleh pihak ketiga yang tidak berwenang. Implementasi tanda tangan digital juga digunakan dalam komunikasi antara *backend* dan modul AI untuk menjaga integritas data selama proses analisis. Ini memastikan bahwa semua data yang dianalisis oleh modul AI adalah data asli dan tidak mengalami manipulasi, memungkinkan hasil analisis yang lebih akurat dan andal.

D. Hash Function

Fungsi *hash* adalah algoritma yang mengubah *input* data menjadi string *output* tetap yang unik, yang dikenal sebagai *hash value*. *Hash value* ini berfungsi sebagai sidik jari digital dari data asli, di mana setiap perubahan kecil pada *input* data akan menghasilkan *hash value* yang berbeda secara signifikan. Algoritma *hash* seperti SHA-256 (*Secure Hash Algorithm 256-bit*) sangat populer karena tingkat keamanannya yang tinggi, menghasilkan *output hash* sepanjang 256 bit yang sangat sulit dipalsukan atau direkayasa ulang. Fungsi hash sangat penting dalam kriptografi karena mereka menyediakan cara untuk memastikan integritas data, mendeteksi perubahan atau manipulasi yang tidak sah, dan mengamankan data dalam berbagai aplikasi digital.

Fungsi *hash* seperti SHA-256 dapat digunakan untuk membuat *checksum* atau *hash value* dari data glukosa sebelum data tersebut dikirim ke *backend*. Proses ini melibatkan penghitungan *hash value* dari data glukosa yang akan dikirim dan menyertakan *hash value* tersebut bersama data yang dikirim. Ketika *backend* menerima data tersebut, *backend* dapat menghitung ulang *hash value* dari data yang diterima dan membandingkannya dengan *hash value* yang dikirimkan. Jika kedua *hash value* tersebut cocok, maka data tersebut dianggap tidak berubah selama transmisi, memastikan integritas data. Selain itu, fungsi *hash* juga dapat digunakan dalam kombinasi dengan tanda tangan digital untuk memberikan lapisan keamanan tambahan dalam proses verifikasi data. Dengan cara ini, tidak hanya keaslian data yang terjamin tetapi juga integritasnya, mencegah kemungkinan manipulasi data oleh pihak yang tidak berwenang selama transmisi.

E. Protokol Kriptografi dan Infrastruktur Kunci-Publik (PKI)

Protokol kriptografi adalah seperangkat aturan yang menentukan bagaimana algoritma kriptografi digunakan untuk mencapai keamanan komunikasi. Dalam konteks sistem Glutara, protokol seperti TLS (*Transport Layer Security*) memainkan peran penting dalam mengamankan komunikasi antara berbagai komponen sistem, termasuk perangkat IoT, *backend*, dan modul AI. TLS bekerja dengan mengenkripsi

data yang ditransmisikan sehingga hanya pihak yang memiliki kunci dekripsi yang benar yang dapat membaca data tersebut. Ini memastikan bahwa data yang dikirimkan antara perangkat Arduino dan *backend*, serta antara *backend* dan modul AI, tetap aman dari penyadapan dan manipulasi oleh pihak yang tidak berwenang. Dengan menggunakan TLS, Glutara dapat memberikan jaminan bahwa semua komunikasi antar komponen sistem dilakukan dengan cara yang aman dan terenkripsi, melindungi data sensitif pengguna dari ancaman keamanan siber.

Infrastruktur Kunci-Publik (PKI) mendukung implementasi protokol kriptografi ini dengan menyediakan cara untuk mengelola kunci publik dan privat yang digunakan dalam proses enkripsi dan dekripsi. PKI memungkinkan Glutara untuk menerbitkan, mendistribusikan, dan mencabut sertifikat digital yang digunakan untuk autentikasi dan enkripsi. Sertifikat digital ini berfungsi sebagai identitas digital yang dapat diverifikasi, memastikan bahwa hanya entitas yang terverifikasi yang dapat berkomunikasi dalam sistem. PKI juga memfasilitasi pengelolaan kunci kriptografi, termasuk pembuatan, penyimpanan, dan distribusi kunci publik dan privat, serta menyediakan mekanisme untuk pencabutan kunci jika terjadi pelanggaran keamanan. Dengan memanfaatkan PKI, Glutara dapat memastikan bahwa semua entitas dalam sistem adalah sah dan bahwa komunikasi antara mereka aman dan terlindungi, memberikan lapisan keamanan tambahan yang sangat penting dalam ekosistem IoT dan AI yang kompleks.

III. Rencana Penyelesaian Masalah

Rencana penyelesaian masalah yang akan diterapkan digunakan untuk meningkatkan keamanan sistem dengan menggunakan teknologi kriptografi modern dan metode enkripsi yang kuat. Tujuan utama dari rencana ini adalah memastikan bahwa data sensitif pengguna tetap terlindungi selama proses pengumpulan, transmisi, penyimpanan, dan analisis data.

A. Rancangan Solusi dan Arsitektur

Solusi yang diusulkan untuk sistem manajemen diabetes Glutara menggabungkan teknologi IoT, kriptografi, dan machine learning untuk menciptakan sebuah ekosistem yang aman dan efektif. Arsitektur sistem terdiri dari beberapa komponen utama: perangkat IoT (Arduino), database, layanan backend, dan aplikasi mobile. Perangkat IoT bertugas mengumpulkan data glukosa secara real-time dari pengguna menggunakan sensor CGM (*Continuous Glucose Monitor*). Data ini kemudian dienkripsi dan dikirimkan ke database melalui jaringan yang aman.

Di sisi server, layanan backend yang dikembangkan menggunakan *Golang* dan *Gin* Framework berfungsi untuk menerima, menyimpan, dan memproses data. Backend juga menyediakan API untuk aplikasi mobile yang digunakan oleh pengguna untuk memantau level glukosa mereka. Layanan AI/ML yang berjalan di backend menggunakan TensorFlow dan MLKit untuk menganalisis data glukosa dan memberikan wawasan nutrisi berdasarkan pola yang teridentifikasi. Semua komunikasi antara komponen ini dienkripsi menggunakan protokol TLS dan kunci publik/privat yang dikelola melalui infrastruktur kunci publik (PKI).

Sistem ini juga mengimplementasikan mekanisme autentikasi yang kuat menggunakan Firebase Authentication dan enkripsi *end-to-end* (E2EE) untuk memastikan bahwa hanya pengguna yang sah yang dapat mengakses data sensitif mereka. Dengan menggunakan teknik kriptografi seperti *Elliptic Curve Cryptography* (ECC) dan tanda tangan digital, sistem memastikan integritas dan keaslian data dari saat data dikumpulkan hingga diproses dan disajikan kepada pengguna. Arsitektur ini dirancang untuk memberikan keamanan maksimum tanpa mengorbankan kinerja atau kenyamanan pengguna.

B. Implementasi Perangkat IoT (Arduino) -> Database

Langkah pertama dalam sistem Glutara adalah pengumpulan data glukosa menggunakan perangkat IoT berbasis Arduino. Sensor CGM yang terpasang pada Arduino mengukur level glukosa pengguna dan menghasilkan data yang perlu dienkripsi sebelum dikirim ke database. Algoritma enkripsi yang digunakan adalah *Elliptic Curve Cryptography* (ECC) yang menawarkan keamanan tinggi dengan ukuran kunci yang lebih kecil dibandingkan dengan RSA, membuatnya ideal untuk perangkat dengan sumber daya terbatas seperti Arduino.

Proses enkripsi dimulai dengan pengambilan data glukosa yang dihasilkan oleh sensor CGM. Data ini kemudian dikonversi menjadi format yang sesuai untuk enkripsi. Kunci publik yang disimpan dalam perangkat Arduino digunakan untuk mengenkripsi data tersebut. ECC bekerja dengan menggunakan kurva eliptik untuk menghasilkan pasangan kunci publik dan privat. Keuntungan utama dari ECC adalah bahwa ia dapat menyediakan tingkat keamanan yang setara dengan algoritma kriptografi tradisional seperti RSA tetapi dengan ukuran kunci yang lebih kecil, sehingga mengurangi beban komputasi pada perangkat dengan sumber daya terbatas.

Setelah data dienkripsi, data tersebut dikirim ke database melalui koneksi WiFi yang aman. Dalam proses ini, data yang telah terenkripsi ditransmisikan menggunakan protokol HTTP POST ke endpoint backend yang telah ditentukan. Backend kemudian menerima data terenkripsi ini dan menyimpannya di database untuk analisis lebih lanjut. Selama transmisi, data yang dienkripsi dengan ECC memastikan bahwa informasi sensitif tidak dapat diakses oleh pihak yang tidak berwenang, menjaga privasi dan integritas data pengguna.

Algoritma enkripsi menggunakan ECC pada Arduino melibatkan beberapa langkah penting, termasuk inisialisasi kunci publik, enkripsi data glukosa, dan transmisi data yang aman. Keamanan data dijamin melalui penggunaan kunci publik yang dihasilkan dari kurva eliptik, yang memastikan bahwa hanya penerima yang memiliki kunci privat yang sesuai yang dapat mendekripsi dan mengakses data tersebut. Dengan demikian, penggunaan ECC pada perangkat Arduino tidak hanya melindungi data selama transmisi tetapi juga memastikan bahwa data yang disimpan di database tetap aman dan terlindungi dari ancaman keamanan potensial.

C. *Implementasi Backend -> Mobile*

Di sisi *backend*, data yang diterima dari perangkat IoT dienkripsi kembali sebelum dikirim ke aplikasi *mobile* pengguna. *Backend* menggunakan kunci privat untuk menandatangani data dan memastikan keasliannya, sementara kunci publik digunakan oleh aplikasi mobile untuk memverifikasi tanda tangan tersebut. Algoritma tanda tangan digital yang digunakan adalah ECDSA (*Elliptic Curve Digital Signature Algorithm*), yang memanfaatkan keunggulan ECC dalam menghasilkan tanda tangan yang aman dan efisien.

Proses pengiriman data dari backend ke mobile mencakup beberapa langkah penting. Pertama, backend menerima data glukosa terenkripsi dari perangkat IoT. Data ini kemudian didekripsi menggunakan kunci privat yang dimiliki oleh backend. Setelah didekripsi, data tersebut diproses dan dienkripsi ulang menggunakan kunci publik aplikasi mobile. Selanjutnya, data yang telah dienkripsi ditandatangani menggunakan kunci privat backend untuk memastikan keasliannya. Tanda tangan digital ini memungkinkan aplikasi mobile untuk memverifikasi bahwa data yang diterima tidak telah diubah atau dimanipulasi selama transmisi.

Seluruh proses ini memastikan bahwa data glukosa yang dikirim dari backend ke aplikasi mobile tetap aman dan terjamin keasliannya. Penggunaan ECDSA memungkinkan pembuatan tanda tangan digital yang kuat dan efisien, sementara enkripsi ulang dengan kunci publik aplikasi mobile memastikan bahwa hanya aplikasi yang sah yang dapat mengakses data tersebut. Dengan demikian, implementasi ini menyediakan lapisan keamanan tambahan yang melindungi data pengguna dari potensi ancaman dan serangan selama transmisi.

D. *Implementasi Database -> Backend*

Implementasi kriptografi dari database ke backend adalah langkah penting untuk memastikan bahwa data yang telah disimpan di database dapat diakses kembali oleh backend secara aman. Proses ini dimulai ketika backend melakukan permintaan data ke database. Data yang disimpan di database sebelumnya telah dienkripsi menggunakan ECC saat dikirim dari perangkat IoT. Ketika data ini diminta oleh backend, data tersebut harus didekripsi menggunakan kunci privat yang hanya dimiliki oleh backend.

Proses dekripsi ini memastikan bahwa hanya backend yang dapat mengakses dan memproses data sensitif tersebut. Algoritma yang digunakan untuk dekripsi adalah ECC, sama seperti pada proses enkripsi. Data yang telah didekripsi kemudian diverifikasi keasliannya menggunakan tanda tangan digital. Backend akan menggunakan kunci publik yang terkait untuk memverifikasi bahwa data tersebut tidak telah diubah atau dimanipulasi selama penyimpanan di database.

Selain dekripsi dan verifikasi, backend juga harus memastikan bahwa data yang diterima sesuai dengan yang diharapkan. Hal ini dilakukan dengan memeriksa integritas data menggunakan hash. Fungsi hash, seperti SHA-256, digunakan untuk menghasilkan hash value dari data yang diterima dan dibandingkan dengan hash value yang disimpan di database. Jika hash value sesuai, maka data dianggap tidak berubah dan aman untuk digunakan.

Proses ini mencakup beberapa langkah penting: menerima data terenkripsi dari database, mendekripsi data menggunakan kunci privat backend, memverifikasi tanda tangan digital, dan memeriksa integritas data menggunakan hash. Implementasi ini memastikan bahwa data yang disimpan di database tetap aman selama penyimpanan dan dapat diakses kembali oleh backend tanpa risiko manipulasi atau kehilangan integritas. Dengan demikian, proses ini menyediakan lapisan keamanan tambahan yang melindungi data pengguna selama penyimpanan dan akses oleh backend.

IV. Hasil Dan Pembahasan

Pada bagian ini, akan dibahas hasil dari implementasi kriptografi pada sistem manajemen diabetes Glutara. Fokus utama adalah pada bagaimana data pengguna dienkripsi dan didekripsi pada setiap tahap, serta bagaimana keamanan dan integritas data dijaga sepanjang proses. Kami juga akan memberikan contoh data yang berhasil dienkripsi dan didekripsi di setiap tahap untuk menunjukkan efektivitas dari solusi yang diusulkan.

A. *Hasil Implementasi Perangkat IoT (Arduino) -> Database*

Implementasi enkripsi menggunakan Elliptic Curve Cryptography (ECC) pada perangkat IoT (Arduino) berhasil dilakukan. Data glukosa yang dikumpulkan oleh sensor CGM dienkripsi menggunakan kunci publik yang disimpan pada perangkat Arduino sebelum dikirim ke database. Berikut adalah contoh data yang berhasil dienkripsi pada tahap ini.

Data Asli	95.4
Data Terenkripsi	04B2A5F6D4C3A8E92B7C1A 2D3F1E6B5A4F3D2C1A8E7B 6C5D4F3A2B1C9D8E7F6C5A 4B3D2E1F4

Data terenkripsi ini kemudian dikirim melalui koneksi *WiFi* yang aman ke database menggunakan protokol HTTP POST. Proses ini memastikan bahwa data yang dikirim dari perangkat IoT ke database tidak dapat diakses oleh pihak yang tidak berwenang, menjaga privasi dan keamanan informasi pengguna.

B. *Hasil Implementasi Database -> Backend*

Setelah data glukosa yang terenkripsi disimpan di database, backend melakukan permintaan data untuk memproses dan menganalisis informasi tersebut. Data yang diterima dari database dienkripsi menggunakan kunci privat backend, yang memastikan bahwa hanya backend yang dapat mendekripsi dan mengakses data tersebut. Berikut adalah contoh data yang berhasil didekripsi pada tahap ini.

Data Terenkripsi	04B2A5F6D4C3A8E92B7C1A 2D3F1E6B5A4F3D2C1A8E7B 6C5D4F3A2B1C9D8E7F6C5A

	4B3D2E1F4
Data Didekripsi	95.4

Proses dekripsi ini melibatkan penggunaan kunci privat yang hanya dimiliki oleh backend. Setelah data didekripsi, backend juga melakukan verifikasi tanda tangan digital untuk memastikan bahwa data tidak dimanipulasi selama penyimpanan di database. Proses verifikasi ini menggunakan kunci publik yang terkait untuk memverifikasi keaslian data.

C. Hasil Implementasi Backend -> Mobile

Data yang diterima dan diproses oleh *backend* kemudian dienkripsi kembali sebelum dikirim ke aplikasi *mobile* pengguna. Pada tahap ini, backend menggunakan kunci publik aplikasi mobile untuk mengenkripsi data dan kunci privat untuk menandatangani data. Berikut adalah contoh data yang berhasil dienkripsi dan ditandatangani pada tahap ini.

Data Asli	95.4
Data Terenkripsi	8F2D4B6A9C1D3E4F7B8 A9C1D2F3E6B5A4F3D2C 1A8E7B6C5D4F3A2B1C9 D8E7F6C5A4B3D2E1F4
Tanda Tangan Digital	3045022100C1D2E3F4A5B 6C7D8E9F0A1B2C3D4E5F 60708090A1B2C3D4E5F60 708090A1B2C022100D1E2 F3A4B5C6D7E8F9A0B1C2 D3E4F507060708090A1B2 C3D4E5F60708090A1B2C 3D4

Aplikasi mobile kemudian menerima data terenkripsi ini dan menggunakan kunci privatnya untuk mendekripsi data serta kunci publik backend untuk memverifikasi tanda tangan digital. Proses ini memastikan bahwa data yang diterima oleh aplikasi mobile asli dan tidak dimanipulasi selama transmisi.

V. KESIMPULAN

Implementasi kriptografi modern pada sistem manajemen diabetes Glutara telah berhasil meningkatkan keamanan dan integritas data pengguna secara signifikan. Dengan menggunakan Elliptic Curve Cryptography (ECC) untuk enkripsi data di perangkat IoT, tanda tangan digital untuk memastikan keaslian data, dan protokol TLS untuk mengamankan komunikasi antar komponen, Glutara mampu menciptakan sebuah ekosistem yang aman dan efektif. Hasil dari implementasi ini menunjukkan bahwa teknologi kriptografi dapat melindungi data sensitif pengguna dari potensi ancaman, memastikan bahwa data tetap utuh dan tidak dimanipulasi selama proses pengumpulan, transmisi, penyimpanan, dan analisis. Dengan demikian, Glutara dapat memberikan layanan yang lebih aman dan terpercaya kepada penggunanya, membantu mereka mengelola kondisi diabetes mereka dengan lebih efektif dan efisien.

UCAPAN TERIMA KASIH

Penulis mengucapkan puji dan syukur yang sebesar-besarnya kepada Tuhan Yang Maha Esa atas berkat dan Rahmat-Nya sehingga makalah yang berjudul "Implementasi Kriptografi dalam Glutara: IoT Manajemen Diabetes" dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih kepada dosen pengajar IF4020 Kriptografi, Dr. Rinaldi Munir, S.T., M.T., yang telah memberikan bimbingan dan ilmu terkait materi kriptografi, khususnya dalam penerapan algoritma tanda tangan digital. Ucapan terima kasih yang sebesar-besarnya juga penulis sampaikan kepada para penulis sumber referensi yang digunakan dalam makalah ini, yang telah memberikan wawasan dan pengetahuan yang diperlukan untuk menyelesaikan tugas ini.

PRANALA KODE PROGRAM IMPLEMENTASI

Kode program implementasi dapat diakses pada pranala berikut.

https://github.com/AustinPardosi/Crytography-IoT.git

DAFTAR PUSTAKA

[1] Munir, Rinaldi. 2024. Slide Kuliah Kriptografi (Bandung: Institut Teknologi Bandung)

PERNYATAAN

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

Bandung, 12 Juni 2024

Austin Gabriel Pardosi
13521084