Analisa Penggunaan *Elliptic Curve Cryptography* pada Sistem Autentikasi pada *Internet of Things*

Daniel Perdana Putra Purwiko[1], Favian Dewanta[1], Farah Afianti[2]
[1]Teknik Telekomunikasi, Fakultas Elektro, Telkom University
[2]Teknik Informatika, Fakultas Informatika, Telkom University
Bandung, Jawa Barat
anerket@student.telkomuniversity.ac.id, favian@telkomuniversity.ac.id, farahafi@telkomuniversity.ac.id

Diterima: 16 Agustus 2022. Disetujui: 27 Agustus 2022. Dipublikasikan: 29 Agustus 2022.

Abstract - The Internet of Things is a complex system that is widely used in many ways to advance human life. As a result, Internet of Things (IoT) has many security vulnerabilities and requires an authentication system to protect user data. Selecting the authentication type that suits your needs is critical to achieving excellent performance on your Internet of Things (IoT) devices with minimal specifications. Due to this situation, Elliptic Curve Cryptography* (ECC) *algorithm is one of the recommended algorithms which consumes less resources in the process. This study aims to test and compare the Fiat-Shamir based Elliptic Curve Cryptography (ECC) and Elliptic Curve Diffie-Hellman based Hash Message Authentication Code (ECDH-HMAC) authentication algorithms. Parameters for this test are computation time, delay, memory usage, and communication cost of the authentication algorithm. The experimental results show that the Elliptic Curve Diffie-Hellman based Hash Message Authentication Code (ECDH-HMAC) algorithm has the lowest computational time, delay, and memory usage, and the Fiat-Shamir based Elliptic Curve Cryptography (ECC) algorithm has the lowest communication cost value.

Keywords: ECC, ECDH, MAC, Fiat-Shamir, HMAC

Abstrak*-- Internet of Things adalah sistem kompleks yang banyak digunakan dalam banyak cara untuk memajukan kehidupan manusia. Akibatnya, *Internet of Things* (IoT) memiliki banyak kerentanan keamanan dan memerlukan sistem autentikasi untuk melindungi data pengguna. Memilih jenis autentikasi yang sesuai dengan kebutuhan, sangat penting untuk mencapai kinerja yang sangat baik pada perangkat *Internet of Things* (IoT) dengan spesifikasi yang minimal. Karena situasi ini, algoritma *Elliptic Curve Cryptography* (ECC) adalah salah satu algoritma yang direkomendasikan yang mengkonsumsi lebih sedikit sumber daya dalam prosesnya. Penelitian ini bertujuan untuk menguji dan membandingkan algoritma autentikasi *Elliptic Curve Cryptography* (ECC) berbasis *Fiat-Shamir dan Elliptic Curve Diffie-Hellman* berbasis *Hash Message Authentication Code* (ECDH-HMAC). Parameter untuk pengujian ini adalah waktu komputasi, *delay*, penggunaan memori, dan *communication cost* dari algoritma autentikasi. Hasil eksperimen menunjukkan bahwa algoritma *Elliptic Curve Diffie-Hellman* berbasis *Hash Message Authentication Code* (ECDH-HMAC) memiliki waktu komputasi, *delay*, dan penggunaan penggunaan memori terendah, dan algoritma *Elliptic Curve Cryptography* (ECC) berbasis *Fiat-Shamir* memiliki nilai *communication cost* terendah.

Kata kunci: ECC, ECDH, MAC, Fiat-Shamir, HMAC

I. PENDAHULUAN

Kriptografi adalah suatu ilmu yang mempelajari bagaimana mengirimkan suatu pesan secara rahasia, sehingga hanya orang yang dituju saja yang dapat membaca pesan rahasia tersebut. Kata kriptografi berasal dari dua kata Yunani yaitu *crypto* yang memiliki arti rahasia dan *grapho* yang memiliki arti tulisan [1]. Dalam sistem, kriptografi memiliki empat tujuan utama dalam kebutuhan kemanan menurut [2] yaitu kerahasiaan, keutuhan dan keaslian data, keaslian pengirim, pengendalian akses, dan ketersediaan. Kriptografi dapat digolongkan ke dalam dua jenis yaitu kriptografi kunci simetris dan kriptografi kunci asimetris [3]. Kriptografi kunci simetris menggunakan kunci yang sama untuk mengenkripsi *plaintext* dan mendekripsi *ciphertext*. Enkripsi tersebut aman selama kunci privat hanya diketahui oleh kedua pihak yang saling mengirim pesan. Kriptografi asimetris merupakan prosedur pemecahan yang memakai kunci yang tidak selaras untuk mengenkripsi dan mendekripsi. Kunci untuk mengenkripsi akan didistribusikan sebagai akibatnya siapapun yg ingin mengirim pesan secara misteri bisa memakai kunci tersebut, dan kunci untuk mendekripsi dirahasiakan sebagai akibatnya hanya pemilik kunci yang dapat

mendekripsi *ciphertext* yg sudah dienkripsi menggunakan kunci publik.

Internet of Things (IoT) adalah konsep penyematan teknologi seperti sensor dan perangkat lunak ke dalam objek yang berkomunikasi, mengontrol, menghubungkan, dan bertukar data melalui perangkat lain dengan tetap terhubung ke pada Internet. [4]. Pada Saat ini IoT menggunakan perangkat yang disematkan dan juga dikonfigurasi sesuai kebutuhan seperti jam tangan pintar, rumah pintar, dll. Hampir semua perangkat IoT memiliki kemampuan untuk merekam atau menyimpan data tertentu dan memerlukan autentikasi data untuk mencegah penyadapan dan modifikasi data oleh pihak yang tidak bertanggung jawab. Namun, karena perangkat IoT umumnya memiliki daya komputasi yang rendah, maka penting untuk memperhatikan autentikasi yang digunakan. Untuk alasan ini, algoritma yang digunakan harus cukup sederhana secara komputasi untuk memenuhi persyaratan ini. Algoritma yang digunakan adalah *Elliptic Curve Cryptography* (ECC). *Elliptic Curve Cryptography* (ECC) dikembangkan secara independen oleh Victor Miller pada tahun 1986 dan oleh Neil Koblitz pada tahun 1987 [5]. Sejak itu, ECC telah dievaluasi oleh matematikawan dan pakar komputer di seluruh dunia, dan ECC banyak digunakan untuk keamanan *cryptocurrency*.

Penelitian terkait *Fiat-Shamir* juga dilakukan untuk Verifikasi Tiket Rahasia [6]. Pada sistem kerja yang diajukan, menggunakan protokol *Feige-Fiat Shamir* dapat berjalan dengan baik dan aman [6]. Protokol *Feige-Fiat* tersebut dirancang supaya melakukan proses verifikasi tiket serta mencegah penyalahgunaan data, yaitu duplikasi maupun penggunaan data sebanyak dua kali atau lebih pada satu sesi yang sama dengan menggunakan implementasi *Zero Knowledge Proof* berbasis pada protokol *Fiat-Shamir*, sehingga menjadi peluang untuk penerapannya ke dalam perangkat yang berpotensi memiliki peluang untuk di akses oleh seseorang yang tidak memiliki hak atau wewenang ke dalam perangkat tersebut.

Dalam studi yang dilakukan Ravi dan Ashwitha pada tahun 2016, *Elliptic Curve Diffie-Hellman* (ECDH) diterapkan ke perangkat *Internet of Things* (IoT) yang disebut ESP8266 [7]. Dalam hal ini, peluang *Internet of Things* (IoT) akan menjadi revolusi industri berikutnya, dengan perangkat WiFi yang aman dan murah memainkan peran penting. Ini dapat digunakan di banyak aplikasi seperti jaringan mesh, otomatisasi rumah, pengukur pintar, perangkat yang dapat dikenakan, tag ID keamanan, jaringan sensor, dan banyak lagi.

Penelitian yang dilakukan oleh Fredrik [8] pada tahun 2020 mengajukan implementasi autentikasi pada protokol *Constrained Application Protocol* (CoAP) menggunakan *Fiat-Shamir identification scheme*, mengkaji tentang algoritma autentikasi yang berhasil diimplementasikan dan memberikan ilustrasi mengenai peningkatan performa memori dan waktu komputasinya. Kerangka kerja autentikasi pada protokol CoAP diajukan hanya mengimplementasikan satu algortima autentikasi yaitu menggunakan algoritma autentikasi *Fiat-Shamir*. Berdasarkan hasil penelitian tersebut, diharapkan dapat membandingkan dengan berbagai jenis algoritma autentikasi yang ada saat ini.

Penelitian yang dilakukan oleh Maulana [9] pada tahun 2019 mengajukan implementasi metode autentikasi dengan *Zero Knowledge Proof* (ZKP) menggunakan protokol *Fiat-Shamir identification scheme* pada perangkat *Internet of Things* (IoT), dengan mengkaji tentang algoritma autentikasi *Fiat-Shamir* yang diimpelemtasikan ke dalam perangkat IoT dengan menggunakan bahasa pemrograman *python*. Berdasarkan hasil penelitian tersebut, algoritma autentikasi *Fiat-Shamir* berhasil diimplementasikan ke dalam perangkat IoT tetapi ketika menguji tingkat keamanan dengan serangan *sniffing* dan serangan *bruteforce*, pada serangan *sniffing* tidak berhasil mendapatkan informasi rahasia atau tahan terhadap serangan *sniffing* namun dapat digunakan sebagai pengintai untuk serangan lainnya. Sedangkan pada serangan *bruteforce*, algoritma autentikasi *Fiat-Shamir* berhasil dilakukan atau tidak rentan terhadap serangan *bruteforce*. Oleh karena itu diharapkan dapat mennggunakan algoritma tambahan atau algoritma lain yang menambah tingkat keamanan dalam algoritma autentikasi tersebut.

Sedangkan pada penelitian yang dilakukan oleh Miller dan Koblitz pada pertengahan tahun 1987, *Elliptic Curve Cryptography* (ECC) telah diterapkan secara luas dalam kriptografi kunci publik, terutama dalam menghubungkan beberapa kriptosistem [10]. Hal tersebut dikarenakan ECC menggunakan ukuran kunci yang lebih pendek dan juga memiliki tingkat keamanan yang cukup tinggi dibandingkan kunci publik lainnya. Misalnya, *Elliptic* ECC kunci 160 *bits* setara dengan Rivest Shamir Adleman (RSA) kunci 1024 *bits*, ECC kunci 224 *bits* setara dengan RSA kunci 2048 *bits*, dan

ECC kunci 256 *bits* setara dengan RSA kunci 3072 *bits* [11].

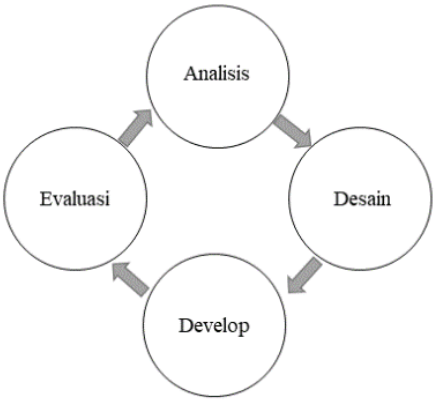
Karena panjang kunci yang pendek, kecepatan tinggi, dan konsumsi daya yang rendah, *Elliptic Curve Cryptography* (ECC) diterapkan secara luas pada perangkat dengan ruang penyimpanan dan bandwidth yang terbatas [12]. Hal ini membuat ECC sangat cocok untuk perangkat *Internet of Things* (IoT) karena memiliki komputasi yang relatif rendah. Meskipun optimasi ECC baru-baru ini, yang mengurangi kompleksitas komputasi, ECC masih kompleks dan membutuhkan optimasi lebih lanjut. Hal terpenting dan inti dari operasi *di* ECC untuk enkripsi, dekripsi, tanda tangan digital, dan pertukaran kunci.

Penelitian ini akan menerapkan metode dengan menganalisis kinerja Elliptic Curve Cryptography (ECC) berbasis *Fiat-Shamir* dan *Elliptic Curve Diffie-Hellman* berbasis *Hash Message Authentication Code* (ECDH-HMAC) pada perangkat *Internet of Things* (IoT). Penelitian ini juga akan membandingkan kinerja ECC berbasis *Fiat-Shamir* dengan ECDH-HMAC pada perangkat IoT yang bertujuan untuk memaksimalkan kinerja terbaik pada algoritma ECC untuk diterapkan sebagai otentikasi pada perangkat IoT tersebut.

Penelitian ini, disusun sebagai berikut. Di bagian 2 berisi tentang perencanaan proses penelitian yang lebih terstruktur dan juga menjelaskan sistem dari algoritma *Elliptic Curve Cryptography* (ECC) berbasis *Fiat-Shamir* dan algoritma *Elliptic Curve Diffie-Hellman* berbasis *Hash Message Authentication Code* (ECDH-HMAC). Di bagian 3 berisi tentang deskripsi yang lebih rinci tentang hasil yang ingin diperoleh dari penelitian ini yaitu mengetahui waktu komputasi, *delay*, penggunaan memori, dan *communication cost* dari perbandingan algoritma autentikasi ECC berbasis *Fiat-Shamir* dengan ECDH-HMAC. Di bagian 4, berisi tentang rangkuman kesimpulan dari penelitian ini.

II. METODE PENELITIAN

Dalam melakukan penelitian diperlukan sebuah perencanaan supaya proses penelitian lebih terstruktur, oleh sebab itu pada penelitian ini akan membuat sebuah diagram perencanaan untuk membantu proses penelitian ini berlangsung secara sistematis.



Gambar 1. Alur Penelitian

Gambar 1 menjelaskan alur pada penelitian ini, yang dimulai dengan melakukan analisis kuantitatif terhadap algoritma *Elliptic Curve Cryptography* (ECC) yang masih ralatif sedikit terkait implementasinya. Analisis ini dilakukan dengan membaca berbagai literatur yang membahas mengenai algoritma ECC. ECC dianggap lebih unggul dibandingkan kriptografi kunci asimetris lainnya. Alasan utamanya adalah bahwa dengan menggunakan kunci yang jauh lebih pendek, ECC dapat memberikan tingkat keamanan yang sama seperti algoritma asimetris lainnya yang menggunakan kunci yang lebih besar. Semakin tinggi tingkat keamanan dan semakin kecil ukuran kunci, maka semakin efisien implementasi ECC. Oleh karena itu, mengkonsumsi lebih sedikit energi untuk memproses ECC. *Fiat-Shamir* adalah metode paralel yang menggunakan pasangan kunci publik dan privat dan juga merupakan suatu skema identifikasi protokol yang dapat digunakan untuk prosedur autentikasi ke suatu sistem. Keuntungan dari skema ini adalah operasi matematika yang sederhana. Dari literatur-literatur tersebut diperoleh bahwa algoritma ECC sangat sesuai diimplementasikan pada perangkat yang memiliki spesifikasi rendah. Untuk membuktikan pernyataan tersebut maka pada penelitian akan meneliti mengenai algoritma ECC ini dengan cara membandingkan dengan menggunakan ECC berbasis *Fiat-Shamir* dan ECDH-HMAC. ECDH merupakan suatu protokol perjanjian kunci yang memungkinkan antara dua pihak pengirim dan penerima, yang pada awalnya masing-masing pihak memiliki kurva ellips dengan sepasang kunci publik dan privat, kemudian masing-masing pihak tersebut mempunyai tujuan yang sama yaitu mendapatkan satu kunci rahasia untuk digunakan bersama-sama namun melalui saluran publik yang sangat beresiko

[13]. HMAC merupakan salah satu varian dari *Message Authentication Code* (MAC) berbasis fungsi hash satu arah dan juga merupakan metode yang digunakan untuk memastikan integritas serta autentikasi sebuah data melalui algoritma hash yang diproses bersama dengan kunci rahasia. HMAC menjamin autentikasi karena adanya kunci rahasia, sedangkan integritas data diperoleh dari algoritma hash yang digunakan. Meski demikian, kekuatan HMAC tetap didasarkan pada algoritma hash yang digunakan.

Desain dari penelitian ini menggunakan algoritma pertukaran kunci algoritma ECDH-HMAC dan ECC berbasis *Fiat-Shamir*. Selanjutnya, untuk mengembangkan desain penelitian ini, peneliti membuat program untuk proses autentikasi dan akan mensimulasikan program dan melakukan pengambilan dan pengolahan data untuk mengevaluasi perbandingan performansi algoritma ECC berbasis *Fiat-Shamir* dengan ECDH-HMAC sesuai dengan parameter-parameter yang diinginkan, sehingga diakhir dapat ditarik kesimpulan mengenai bagaimana performansi dari masing-masing algoritma.

Gambar 2 menjelaskan tentang perancangan sistem algoritma autentikasi ECC berbasis *Fiat-Shamir*, B dan A akan melakukan verifikasi dengan menggunakan kunci ECC yang telah disepakati yaitu titik G dan titik H, kemudian B akan membuat suatu pesan yang akan dijadikan kunci yaitu x dan kemudian B melakukan perkalian x dengan titik G dan titik H yang menghasilkan xG dan xH yang akan dikirimkan kepada A. Sehingga A memiliki nilai xG dan xH, kemudian A akan memilih nilai acak yaitu c yang akan dikirimkan kepada B. Setelah B mendapatkan nilai c dari A, B akan memilih nilai acak yaitu v dan kemudian B melakukan perkalian v dengan titik G dan titik H yang menghasilkan vG dan vH dan juga membuat nilai r dengan menggunakan rumus (1),

$$r = v - xc \tag{1}$$

yang kemudian akan dikirimkan kepada A. Setelah A memiliki nilai r, vG dan vH, A akan melakukan pengecekan dan pembuktian. Untuk pengecekan yaitu dengan rumus (2)-(3).

$$vG = rG + c(xG) \tag{2}$$

$$vH = rH + c(xH) \tag{3}$$

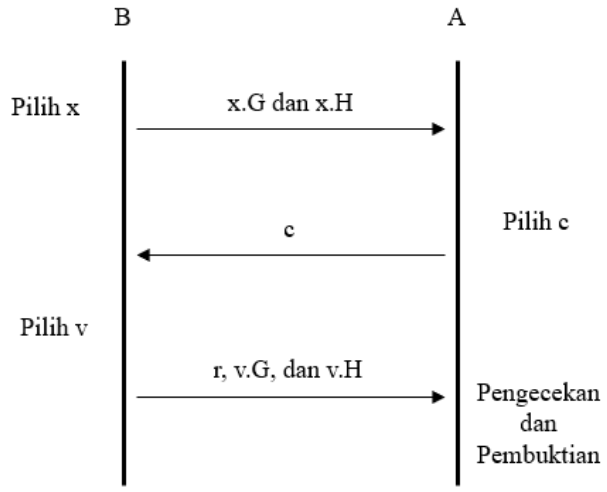
Kemudian untuk pembuktiannya dengan rumus (4)-(5).

$$vG = rG + c(xG) \tag{4}$$

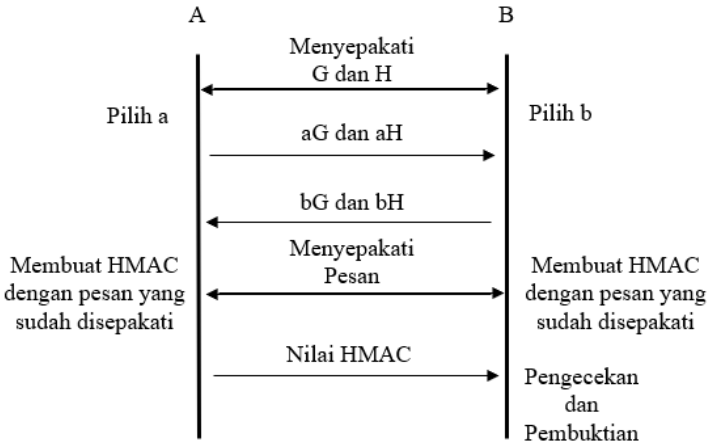
$$(v - cx)G + cxG = vG \tag{5}$$

Jika semua pengecekan dan pembuktian berhasil atau nilainya sama, maka A berhasil melakukan verifikasi dengan B.

Gambar 3 menjelaskan tentang skema perancangan sistem autentikasi ECDH-HMAC. Untuk sistem autentikasi menggunakan ECDH-HMAC sebelum pesan dikirimkan, B akan melakukan pertukaran kunci dengan menggunakan algoritma ECDH kepada A, kemudian B akan menginput kunci ECC dan HMAC yang telah disepakati, berikutnya kunci



Gambar 2. Perancangan Sistem Autentikasi ECC Berbasis *Fiat-Shamir*



Gambar 3. Perancangan Sistem Autentikasi ECDH-HMAC

dan pesan akan proses ke dalam persamaan algoritma HMAC yang akan menghasilkan nilai MAC dan kemudian pesan yang telah diproses akan dikirimkan beserta dengan nilai MAC kepada A.

Kemudian A juga akan memasukan kunci dan pesan kedalam persamaan algoritma HMAC sehingga menghasilkan nilai MAC yang kemudian akan dibandingkan dengan nilai MAC dari B, apabila nilai MAC berbeda, maka pesan telah dirubah oleh pihak lain atau pengiriman pesan mengalami kegagalan. Untuk penjelasan dari notasi perancangan sistem algoritma autentikasi dapat dilihat pada Tabel I.

TABEL I. NOTASI

Notasi	Penjelasan
G	Kunci publik
H	Kunci publik
x	Kunci rahasia
xG	Perkalian antara kunci rahasia x dengan kunci publik G
xH	Perkalian antara kunci rahasia x dengan kunci publik H
c	Nilai acak milik A
v	Nilai acak milik B
xc	Perkalian antara nilai acak c dengan kunci rahasia x
R	Pengurangan antara nilai acak v dengan xc
vG	Perkalian antara nilai acak v dengan kunci publik G
vH	Perkalian antara nilai acak v dengan kunci publik H
a	Kunci rahasia milik A
b	Kunci rahasia milik B
aG	Perkalian antara kunci rahasia a dengan kunci publik G
aH	Perkalian antara kunci rahasia a dengan kunci publik H
bG	Perkalian antara kunci rahasia b dengan kunci publik G
bH	Perkalian antara kunci rahasia b dengan kunci publik H

Sedangkan untuk hasil data yang ingin diperoleh dari penelitian ini yaitu untuk mengetahui waktu komputasi, *delay*, ruang penyimpanan program, dan *communication cost* dari algoritma autentikasi ECC berbasis *Fiat-Shamir* yang disimulasikan kedalam sistem IoT. Hasil uji tersebut akan dibandingkan dengan menggunakan algoritma ECDH-HMAC dengan parameter-parameter yang akan digunakan dalam pengujian.

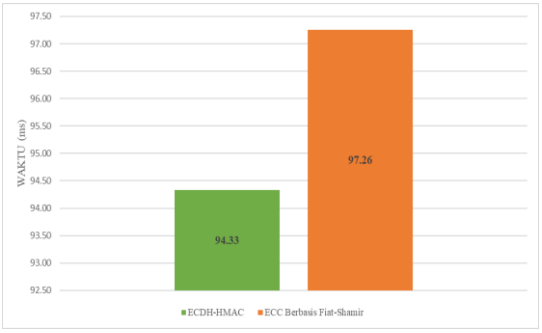
III. HASIL DAN PEMBAHASAN

Bab ini berisi hasil yang ingin diperoleh dari penelitian ini yaitu mengetahui waktu komputasi, *delay*, penggunaan memori, dan *communication cost*

dari perbandingan algoritma autentikasi ECC berbasis *Fiat-Shamir* dengan ECDH-HMAC.

A. Waktu Komputasi

Pengujian ini dilakukan untuk mengetahui perbandingan kecepatan atau waktu komputasi algoritma autentikasi. Lama waktu komputasi algoritma autentikasi merupakan faktor sangat penting dari perangkat IoT yang memiliki spesifikasi rendah. Oleh sebab itu, diperlukan suatu pengujian waktu komputasi yang berfungsi untuk mengetahui berapa lama waktu yang diperlukan dalam proses autentikasinya.

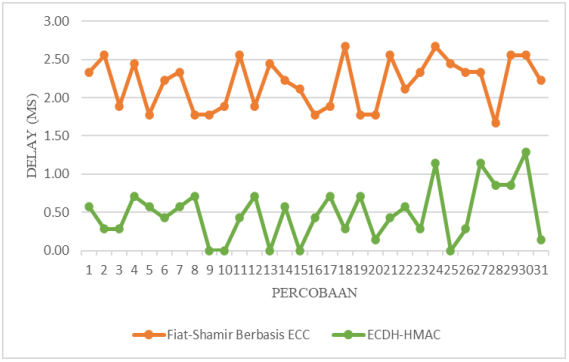


Gambar 4. Grafik Rata-rata Waktu Komputasi Algoritma Autentikasi

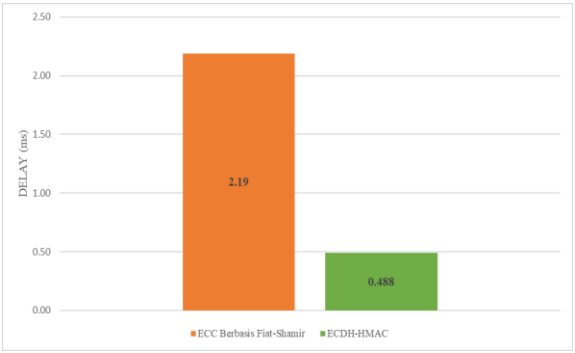
Pada Gambar 4 diperoleh bahwa secara umum, jenis algoritma autentikasi ECDH-HMAC memiliki rata-rata waktu komputasi yang lebih rendah dibandingkan algoritma autentikasi ECC berbasis *Fiat-Shamir*. Pada algoritma ECC berbasis *Fiat-Shamir* memiliki rata-rata waktu komputasi lebih besar daripada algoritma autentikasi ECDH-HMAC atau 3,1% lebih lama dari pada rata-rata waktu komputasi algortima autentikasi *Elliptic Curve Diffie-Hellman* berbasis *Hash Message Authentication Code* (ECDH-HMAC).

B. *Delay*

Lama waktu pemrosesan autentikasi (*delay* komputasi) suatu data dapat memengaruhi waktu sampainya data ke client. Banyak faktor yang dapat memengaruhi lamanya delay komputasi, salah satunya yaitu device yang digunakan. Pada device yang digunakan ECDH-HMAC memiliki kecepatan komputasi yang lebih cepat dibandingkan ECC berbasis *Fiat-Shamir*. Oleh sebab itu, diperlukan suatu pengujian delay yang berfungsi untuk mengetahui berapa lama waktu tunda yang diperlukan dalam proses transmisi paket ke tempat tujuannya. Perhitungan delay yang digunakan dalam pengujian ini yaitu perhitungan *one-way delay* dengan cara menghitung selisih antara waktu paket diterima dengan waktu pengiriman paket.



Gambar 5. Grafik *Delay* Algoritma ECC Berbasis *Fiat-Shamir* dan ECDH-HMAC

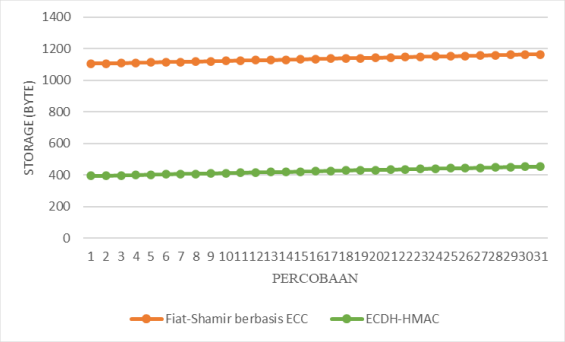


Gambar 6. Grafik Rata-rata *Delay* Algoritma Autentikasi

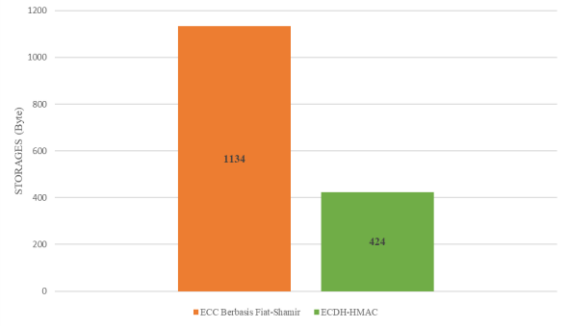
Pada Gambar 5 dalam setiap eksperimen, panjang pesan akan bertambah sebanyak 2 *Byte* atau 2 karakter *string*, tetapi berdasarkan rekap hasil perhitungannya penambahan sebanyak 2 *Byte* atau 2 karakter *string* dalam panjang pesan tidak mempengaruhi *delay* dari algoritma autentikasi ECC berbasis *Fiat-Shamir* dan algoritma autentikasi ECDH-HMAC. Sedangkan untuk algoritma autentikasi, berdasarkan rekap hasil perhitungan yang ditunjukkan oleh Gambar 6 diperoleh bahwa secara umum, jenis algoritma autentikasi ECDH-HMAC memiliki nilai rata-rata *delay* yang lebih rendah dibandingkan algoritma autentikasi ECC berbasis *Fiat-Shamir*. Jenis algoritma autentikasi sangat berpengaruh terhadap lama rata-rata *delay* yang terjadi, dimana pada algoritma ECC berbasis *Fiat-Shamir* memiliki nilai rata-rata *delay* lebih besar daripada algoritma autentikasi ECDH-HMAC atau 348,7% lebih besar dari pada delay algortima autentikasi ECDH-HMAC. Namun dari semua data *delay* hasil pengujian, *delay* yang diperoleh masih tergolong dalam kategori baik karena nilainya kurang dari 200 ms dengan referensi perbandingan standarisasi *delay* ITU-T G. 1010 [14].

C. *Penggunaan Memori*

Penggunaan memori merupakan total data paramater yang digunakan dalam program yang dapat memengaruhi suatu kinerja perangkat keras. Pemilihan algoritma autentikasi yang efisien dalam penggunaan memori sangat diperlukan, karena dalam perangkat IoT rata-rata memiliki kemampuan komputasi dan memori yang rendah. Sehingga semakin sedikit penggunaan memori yang terpakai untuk proses autentikasi maka semakin baik performa dari sistem IoT tersebut. Oleh sebab itu, diperlukan suatu pengujian penggunaan memori yang berfungsi untuk mengetahui berapa besar penggunaan memori yang diperlukan dalam proses menjalankan programnya. Perhitungan penggunaan memori yang digunakan dalam pengujian ini yaitu dengan cara menghitung setiap *Byte* pada parameter yang ada dalam programnya.



Gambar 7. Grafik Penggunaan Memori Algoritma Autentikasi ECC Berbasis *Fiat-Shamir* dan ECDH-HMAC

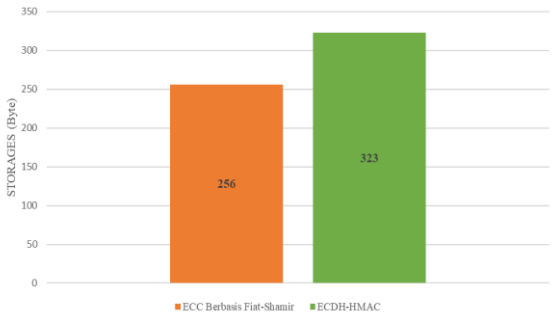


Gambar 8. Grafik Rata-rata Penggunaan Memori Algoritma Autentikasi

Pada Gambar 8 terlihat bahwa ruang penyimpanan program yang paling banyak secara umum yaitu berada pada algoritma autentikasi ECC berbasis *Fiat-Shamir*. Secara keseluruhan, algoritma autentikasi ECDH-HMAC memiliki ruang penyimpanan program paling sedikit yang hanya memiliki ruang penyimpanan program sebesar 442 *Byte* atau 62,6% lebih rendah dari pada algoritma autentikasi ECC berbasis *Fiat-Shamir*.

D. *Communication Cost*

Communication cost merupakan jumlah total paket yang akan dikirimkan atau ditransmisikan dari satu node ke node lain [15]. Banyak faktor yang dapat memengaruhi kinerja suatu pengiriman dan penerimaan data paket, salah satunya yaitu jenis komunikasi yang digunakan. Untuk pengiriman data paket yang digunakan adalah *socket programming*, ECDH-HMAC memiliki penggunaan memori yang lebih sedikit dibandingkan ECC berbasis *Fiat-Shamir*. Oleh sebab itu, diperlukan suatu pengujian *communication cost* yang berfungsi untuk mengetahui berapa jumlah total paket yang dikirimkan dan diterima dalam proses menjalankan programnya. Perhitungan *communication cost* yang digunakan dalam pengujian ini yaitu dengan cara menghitung setiap *Byte* pada setiap paket yang dikirimkan atau ditransmisikan yang ada dalam programnya.



Gambar 9. Grafik Rata-rata *Communication Cost* Algoritma Autentikasi

Dari Gambar 9 terlihat bahwa pengiriman dan penerimaan paket data yang paling banyak secara umum yaitu berada pada jenis algoritma autentikasi ECDH-HMAC. Jenis algoritma autentikasi yang digunakan akan berpengaruh terhadap besar *communication cost*. Secara keseluruhan, algoritma autentikasi ECC berbasis *Fiat-Shamir* memiliki pengiriman paket data yang paling sedikit atau sebesar 20,7% lebih sedikit dari pada algoritma autentikasi ECDH-HMAC.

IV. KESIMPULAN DAN SARAN

Dari hasil penelitian, performansi dari algoritma otentikasi ECC berbasis *Fiat-Shamir* merupakan algoritma otentikasi yang memiliki rata-rata waktu komputasi yang paling lama yaitu 97,26 ms, juga merupakan algoritma otentikasi yang memiliki rata-rata *delay* paling besar yaitu 2,19 ms, juga merupakan algoritma yang memiliki rata-rata penggunaan memori yang paling besar yaitu 1134 *Byte*, dan juga merupakan algoritma yang memiliki

nilai *communication cost* yang paling rendah yaitu sebesar 256 *Byte*.

Sedangkan performansi dari algoritma otentikasi ECDH-HMAC merupakan algoritma otentikasi yang memiliki rata-rata waktu komputasi yang paling cepat yaitu 94,33 ms, yang merupakan algoritma otentikasi yang memiliki rata-rata *delay* paling kecil yaitu 0,488 ms, juga merupakan algoritma yang memiliki rata-rata ruang penyimpanan program paling kecil yaitu 424 *Byte*, dan juga merupakan algoritma yang memiliki nilai *communication cost* yang paling besar yaitu sebesar 323 *Byte*.

Sehingga dapat disimpulkan, jika ingin menerapkan ke dalam perangkat IoT yang memiliki *bandwidth* kecil tetapi memiliki memori yang besar dapat menggunakan algoritma otentikasi ECC berbasis *Fiat-Shamir*, sedangkan jika perangkat IoT memiliki *bandwidth* yang besar tetapi memiliki memori yang kecil dapat menggunakan algoritma otentikasi ECDH-HMAC.

Saran yang dapat diambil dari proses penelitian ini adalah melakukan pengujian dengan implementasi langsung pada jaringan yang lebih luas sehingga nilai yang diperoleh lebih objektif. Selain itu, untuk menguji tingkat keamanan, disarankan untuk melakukan tes serangan dengan berbagai jenis serangan yang berbeda untuk menentukan ketahanan algoritma otentikasi tersebut.

Secara teoritis, hasil penelitian ini dapat dijadikan sebagai referensi pengembangan penelitian selanjutnya. Selain itu, hasil penelitian ini dapat digunakan sebagai dasar untuk implementasi ke dalam berbagai alat, terutama IoT agar dapat membantu keamanan perangkat dalam proses otentikasi.

Secara praktis berdasarkan penelitian yang telah dilakukan, implementasi algoritma ECDH-HMAC dan algoritma ECC berbasis *Fiat-Shamir* dapat diterapkan untuk meningkatkan kemanan perangkat, terutama perangkat yang memiliki spesifikasi yang cukup rendah.

REFERENSI

[1] Hasugian, B. S. (2017). Peranan Kriptografi Sebagai Keamanan Sistem Informasi Pada Usaha Kecil Dan Menengah.

[2] Hidayatullah, A., & Insanudin, E. (2016). Pengenalan Kriptografi dan Pemakaianya Sehari-Hari.

[3] "Symmetric and Asymmetric Encryption - Overview." https://wizardforcel.gitbooks.io/practical-cryptography-for-developers-book (Diakses Juni 30, 2022).

[4] "Apa Itu Internet of Things." https://www.dicoding.com/blog/apa-itu-internet-of-things/ (Diakses Juni 30, 2022).

[5] Damanik S Implementasi Algoritma Elliptic Curve Cryptography (ECC) Untuk Penyandian Pesan Pada Aplikasi Chatting Client Server Berbasis Desktop. *JURIKOM*. 2019; 6(4): 395-400.

[6] W. Raharjo and D. Sutanti, "Implementasi Zero Knowledge Proof Menggunakan Protokol Feige Fiat Shamir Untuk Verifikasi Tiket Rahasia", Ultimatics : Jurnal Teknik Informatika, vol. 7, no. 2, pp. 91-97, Aug. 2016.

[7] R. K. Kodali and A. Naikoti, "ECDH based security model for IoT using ESP8266," 2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2016, pp. 629-633.

[8] Fredrik, J. D., Kusyanti, A., & Siregar, R. A. (2020). Implementasi Autentikasi pada Protokol CoAP menggunakan Feige-Fiat-Shamir Identification Scheme (Vol. 4, Issue 11).

[9] Maulana, L., Kusyanti, A., & Bakhtiar, F. A. (2019). Implementasi Metode Autentikasi dengan Zero Knowledge Proof menggunakan Protokol Feige-Fiat-Shamir Identification Scheme pada Perangkat Internet of Things (Vol. 3, Issue 9).

[10] Yang, Lu & Zhang, Quanling & Li, Jiguo. (2015). Cryptanalysis of Two Tripartite Authenticated Key Agreement Protocols. 159-162.

[11] "ECC vs RSA: Comparing SSL/TLS Algorithms." https://cheapsslsecurity.com/p/ecc-vs-rsa-comparing-ssl-tls-algorithms/ (Diakses Juni 30, 2022).

[12] Dhafin Kawakibi, - (2019) Implementasi Algoritma Elliptic Curve Integrated Encryption Scheme Pada Perangkat Iot Untuk Keamanan Data Smart Home. S1 Thesis, Universitas Pendidikan Indonesia.

[13] D. P. Shah and P. G. Shah, "Revisting of elliptical curve cryptography for securing Internet of Things (IOT)," 2018 Advances in Science and Engineering Technology International Conferences (ASET), 2018, pp. 1-3.

[14] International Telecommunication Union Telecommunication Standarization Sector. G.1010. *End-User Multimedia QoS*. Switzerland Geneva: ITU; 2002.

[15] Anjali, Shikha, and M. Sharma, "Wireless sensor networks: Routing protocols and security issues," Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2014, pp. 1-5.