



## Rancang Bangun Sistem Pembaruan *Firmware Over-the-Air* (OTA) untuk Perangkat ESP32 Berbasis Layanan Cloud

Faris Munir Mahdi<sup>1\*</sup>, Basuki Rahmat<sup>2</sup>, Fawwaz Ali Akbar<sup>3</sup>

<sup>1-3</sup>Universitas Pembangunan Nasional "Veteran" Jawa Timur, Indonesia

Alamat : Jalan Raya Rungkut Madya No. 1 Surabaya, Indonesia

Korespondensi penulis : [farismunirmahdii@gmail.com](mailto:farismunirmahdii@gmail.com)\*

**Abstract.** *The rapid development of Internet of Things (IoT) technology has driven the need for efficient, secure, and reliable firmware update systems. Firmware plays a vital role in maintaining the performance, security, and functionality of IoT devices, including those based on the ESP32, which is widely used due to its cost-effectiveness and connectivity capabilities. However, conventional firmware update methods that require physical connections are inefficient for managing large-scale, distributed devices. Over-the-Air (OTA) technology offers a more relevant solution by enabling wireless firmware updates without manual intervention. This study aims to design and implement a cloud-based OTA firmware update system for ESP32 devices. In this approach, the ESP32 acts as a client that automatically downloads the latest firmware from a cloud server, eliminating the need for a local server infrastructure. The expected outcome is a system capable of efficiently distributing firmware, ensuring the integrity and authenticity of update files, and supporting large-scale IoT device management in a more practical and sustainable manner.*

**Keywords :** *Cloud, Device Security, ESP32, Firmware Update, Internet of Things.*

**Abstrak.** Perkembangan pesat teknologi Internet of Things (IoT) mendorong kebutuhan akan sistem pembaruan firmware yang efisien, aman, dan dapat diandalkan. Firmware berperan penting dalam menjaga performa, keamanan, dan fungsi perangkat IoT, termasuk perangkat berbasis ESP32 yang banyak digunakan karena efisiensi dan kapabilitas konektivitasnya. Namun, metode pembaruan firmware secara konvensional yang memerlukan koneksi fisik terbukti tidak efisien untuk perangkat yang tersebar luas dalam jumlah besar. Teknologi Over the Air (OTA) menjadi solusi yang lebih relevan, memungkinkan pembaruan firmware secara nirkabel tanpa intervensi manual. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pembaruan firmware OTA berbasis cloud untuk perangkat ESP32. Dengan pendekatan ini, ESP32 berperan sebagai klien yang secara otomatis mengunduh firmware terbaru dari server cloud, tanpa membutuhkan infrastruktur server lokal. Hasil yang diharapkan adalah sistem yang mampu mendistribusikan firmware secara efisien, menjaga integritas dan keaslian file pembaruan, serta mendukung pengelolaan perangkat IoT dalam skala besar secara lebih praktis dan berkelanjutan.

**Kata Kunci :** *Internet of Things, ESP32, Pembaruan Firmware, Cloud, Keamanan Perangkat.*

### 1. LATAR BELAKANG

Teknologi Internet of Things (IoT) berkembang sangat cepat dalam beberapa tahun terakhir, dengan dampak signifikan pada cara manusia menggunakan perangkat digital dalam berbagai sektor, termasuk rumah tangga dan industri. Dalam ekosistem ini, firmware memegang peran strategis dalam menjaga fungsi dan keberlanjutan perangkat. Pembaruan firmware menjadi kebutuhan mutlak untuk mengikuti tuntutan fungsionalitas, performa, perbaikan kesalahan, dan perlindungan terhadap celah keamanan (Brightwood, 2024).

Metode pembaruan firmware secara konvensional, yang mengharuskan adanya koneksi fisik ke komputer, dianggap tidak efisien. Cara ini memakan waktu, tidak praktis bagi perangkat yang tersebar luas atau dalam jumlah banyak, dan menyulitkan perawatan

serta pengembangan perangkat dalam jangka panjang. Oleh karena itu, dibutuhkan sebuah pendekatan pembaruan yang lebih efisien dan otomatis (Ikhsani et al., 2023).

Teknologi Over the Air (OTA) menjadi alternatif yang lebih relevan terhadap kebutuhan pembaruan saat ini. Perangkat dapat memperoleh dan menginstal pembaruan firmware tanpa koneksi fisik dan tanpa campur tangan manual. Hal ini secara signifikan mengurangi biaya operasional, mempercepat proses pembaruan, serta memberikan fleksibilitas lebih dalam pengelolaan perangkat. Dalam konteks IoT, OTA menjadi bagian penting dari strategi pemeliharaan jangka panjang dan perlindungan perangkat (Paramartha et al., 2021).

ESP32 digunakan secara luas dalam pengembangan IoT karena karakteristiknya yang sesuai dengan kebutuhan konektivitas dan efisiensi biaya (Harianto, n.d., 2024). Walaupun ESP32 mendukung fitur OTA, implementasinya tetap membutuhkan infrastruktur backend yang kuat. Pendekatan tradisional biasanya mengharuskan perangkat ESP32 untuk berfungsi sebagai server lokal atau access point, sehingga firmware harus diunggah secara manual dari perangkat lain yang terhubung. Metode ini jelas membatasi efisiensi dan tidak cocok untuk penggunaan dalam skala besar (Yusri et al., 2022).

Untuk mengatasi keterbatasan tersebut, integrasi dengan layanan cloud menjadi solusi yang lebih efektif. Tidak seperti pendekatan konvensional, metode berbasis cloud memungkinkan ESP32 bertindak sebagai klien yang langsung mengunduh firmware dari server cloud. Layanan cloud menyediakan berbagai fitur seperti penyimpanan data dan keamanan yang dapat dimanfaatkan untuk mengelola proses pembaruan secara efisien. Pendekatan ini memungkinkan pengembang untuk lebih fokus pada fungsionalitas inti perangkat tanpa perlu membangun dan memelihara infrastruktur server yang kompleks. Selain itu, pembaruan dapat dilakukan secara otomatis dan jarak jauh dalam jumlah besar (Pradita et al., 2025).

Dengan mempertimbangkan tantangan dan peluang yang ada, penelitian ini bertujuan untuk merancang dan membangun sistem pembaruan firmware Over the Air (OTA) yang aman dan andal untuk perangkat ESP32 menggunakan layanan cloud. Sistem ini diharapkan dapat mendistribusikan firmware secara efisien serta menjamin keaslian dan integritas file pembaruan. Dengan demikian, penelitian ini dapat memberikan solusi atas tantangan dalam pembaruan firmware pada perangkat IoT berbasis ESP32 dan mendukung pengembangan aplikasi IoT yang lebih maju dan berkelanjutan.

## 2. KAJIAN TEORITIS

### *Internet of Things (IoT)*

Internet of Things (IoT) adalah teknologi modern yang dirancang untuk mengoptimalkan penggunaan koneksi internet yang selalu aktif. Teknologi ini memungkinkan berbagai benda terhubung untuk mempermudah aktivitas sehari-hari, menjadikannya lebih efisien dan praktis, sehingga berbagai tugas manusia dapat dilakukan dengan lebih mudah. IoT kini semakin banyak diterapkan di berbagai bidang kehidupan. Dalam proses komunikasi, IoT diperkenalkan melalui metode identifikasi RFID (Radio Frekuensi Identifikasi). Selain itu, teknologi ini juga melibatkan sensor, jaringan nirkabel, dan kode QR (Adani & Salsabil, 2021).

### **Mikrokontroler ESP32**

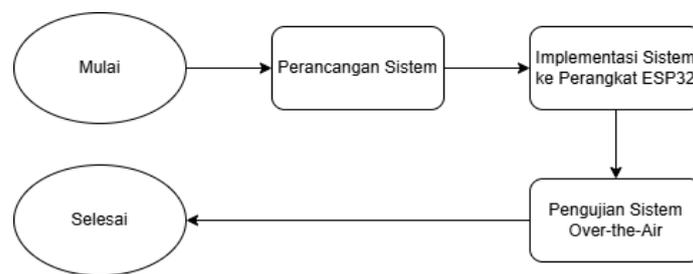
ESP32 adalah mikrokontroler yang dikenalkan oleh Espressif System merupakan penerus dari mikrokontroler ESP8266. Mikrokontroler ESP32 merupakan perangkat SoC (System on Chip) yang lengkap dengan fitur WiFi 802.11 b/g/n, Bluetooth versi 4.2, serta berbagai periferal. ESP32 ini sangat komprehensif karena memiliki prosesor, penyimpanan, dan akses GPIO (General Purpose Input Output). ESP32 dapat digunakan sebagai pengganti rangkaian pada Arduino, serta mampu terhubung ke jaringan WiFi secara langsung (Fathor Azhar & Nurpulaela, 2024).

### *Over The Air (OTA)*

Over The Air (OTA) Update adalah metode pembaruan perangkat lunak yang dilakukan secara nirkabel, memungkinkan perangkat seperti smartphone, tablet, atau perangkat IoT menerima pembaruan tanpa perlu terhubung secara fisik ke komputer atau server. Proses ini penting dalam industri teknologi karena memudahkan distribusi pembaruan yang memperbaiki bug, meningkatkan keamanan, atau menambahkan fitur baru. Keamanan dalam OTA Update dijamin melalui enkripsi dan tanda tangan digital, memastikan hanya pembaruan yang diverifikasi yang dapat di-instal, sehingga perangkat tetap diperbarui dan aman (Ikhsani et al., 2023).

### 3. METODE PENELITIAN

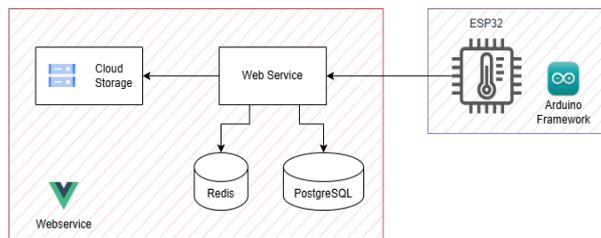
Penelitian ini mengusung pendekatan metodologi yang sistematis untuk mengembangkan dan mengimplementasikan solusi pembaruan firmware Over-the-Air (OTA) yang efisien dan aman bagi perangkat ESP32, dengan memanfaatkan integrasi layanan cloud. Metodologi ini dirancang untuk mengatasi keterbatasan pembaruan konvensional serta memanfaatkan potensi OTA dan cloud dalam ekosistem Internet of Things (IoT). Proses penelitian secara umum mencakup tahapan perancangan, implementasi pada perangkat ESP32, dan pengujian.



**Gambar 1.** Alur Penelitian

Seperti yang ditunjukkan pada Gambar 1, metodologi penelitian ini diawali dengan tahap "Mulai", menandai dimulainya siklus pengembangan. Tahap selanjutnya adalah "Perancangan Sistem", di mana arsitektur keseluruhan sistem pembaruan OTA, termasuk integrasi dengan layanan cloud dan mekanisme keamanan, dirumuskan secara detail. Setelah perancangan sistem selesai dan dianggap memadai, proses berlanjut ke "Implementasi Sistem ke Perangkat ESP32". Tahap ini mencakup penanaman firmware dan client OTA yang memungkinkan perangkat ESP32 berkomunikasi dengan layanan cloud untuk menerima pembaruan. Proses implementasi ini juga mencakup pengaturan konektivitas dan mekanisme pengunduhan firmware. Selanjutnya, dilakukan "Pengujian Sistem Over-the-Air" untuk memverifikasi fungsionalitas, keandalan, dan keamanan sistem pembaruan firmware secara nirkabel. Pengujian ini memastikan firmware dapat didistribusikan secara efisien dan integritas file pembaruan terjamin. Setelah semua pengujian berhasil dan tujuan penelitian tercapai, proses diakhiri dengan tahap "Selesai".

## Perancangan Sistem



**Gambar 2.** Desain Arsitektur Sistem

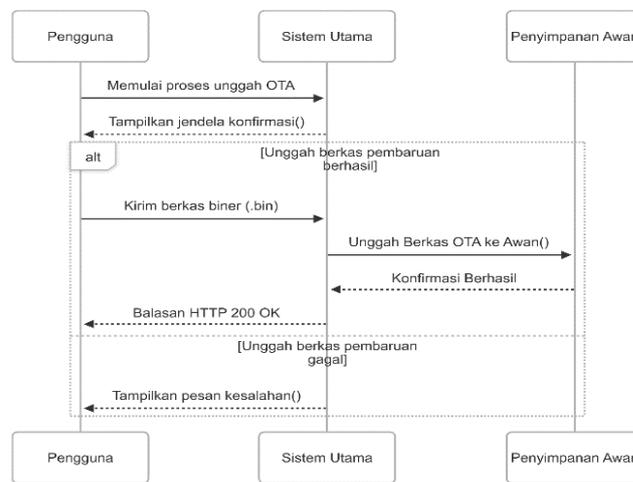
Pada gambar 2. menunjukkan Desain Arsitektur Sistem yang diusulkan dalam penelitian ini. Arsitektur ini terdiri dari dua bagian utama: sistem berbasis Cloud Service dan perangkat ESP32. Pada sisi Cloud Service, terdapat komponen Web Service yang berfungsi sebagai pusat komunikasi dan pengelolaan. Web Service ini terhubung ke Cloud Storage untuk menyimpan file firmware yang akan didistribusikan. Selain itu, Web Service juga berinteraksi dengan dua jenis basis data: Redis dan PostgreSQL. Redis digunakan sebagai cache untuk mempercepat akses data yang sering digunakan atau untuk menyimpan informasi sementara, sementara PostgreSQL digunakan sebagai basis data relasional utama untuk menyimpan data konfigurasi perangkat, riwayat pembaruan, dan informasi relevan lainnya secara persisten.

Di sisi lain, perangkat ESP32, yang dikembangkan menggunakan Arduino Framework, berperan sebagai klien yang akan menerima pembaruan firmware. ESP32 berkomunikasi langsung dengan Web Service melalui koneksi internet. Ketika pembaruan tersedia, ESP32 akan meminta dan mengunduh file firmware dari Cloud Storage melalui perantara Web Service. Desain arsitektur ini memungkinkan pembaruan firmware dapat dilakukan secara Over-the-Air (OTA) dengan aman dan efisien, memanfaatkan skalabilitas dan keandalan layanan cloud untuk mengelola distribusi firmware ke banyak perangkat ESP32 secara terpusat.

### Implementasi Sistem ke Perangkat ESP32

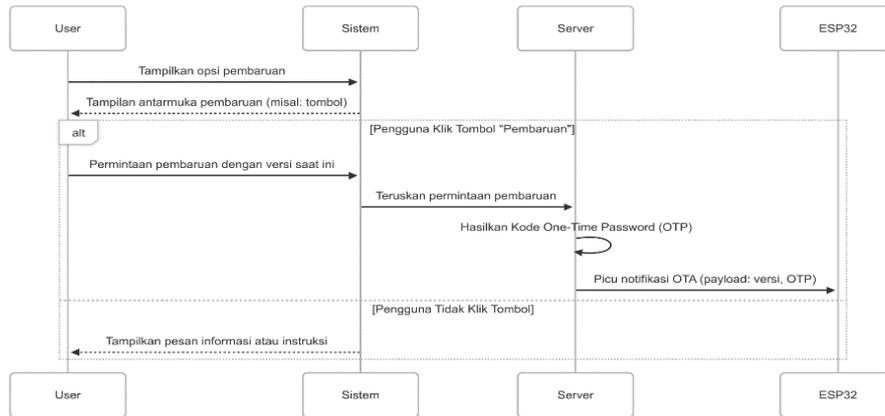
Setelah perancangan arsitektur sistem yang melibatkan Cloud Service dan perangkat ESP32, tahap selanjutnya adalah implementasi sistem ke perangkat ESP32 itu sendiri. Implementasi ini mencakup pengembangan alur kerja untuk pengunggahan file firmware, permintaan pembaruan dan pemicuan OTA, serta proses otentikasi dan pembaruan firmware pada perangkat ESP32. Alur-alur ini dirancang untuk memastikan komunikasi yang lancar dan aman antara perangkat ESP32 dengan layanan cloud yang telah dijelaskan dalam bagian perancangan sistem.

Proses pembaruan firmware dimulai dengan pengunggahan file firmware yang baru ke sistem. Seperti yang digambarkan pada Gambar 3. Alur Pengunggahan File, proses ini diinisiasi oleh Pengguna. Pengguna memulai proses unggah OTA melalui Sistem Utama (yang merepresentasikan Web Service pada arsitektur sistem), yang kemudian akan menampilkan jendela konfirmasi kepada pengguna. Jika pengunggahan berkas pembaruan berhasil, Sistem Utama akan mengirimkan berkas biner (.bin) firmware ke Penyimpanan Awan (Cloud Storage). Setelah berhasil diunggah, konfirmasi akan dikirim kembali ke Sistem Utama, dan Sistem Utama akan memberikan balasan HTTP 200 OK kepada Pengguna. Namun, jika pengunggahan gagal, Sistem Utama akan menampilkan pesan kesalahan kepada pengguna.

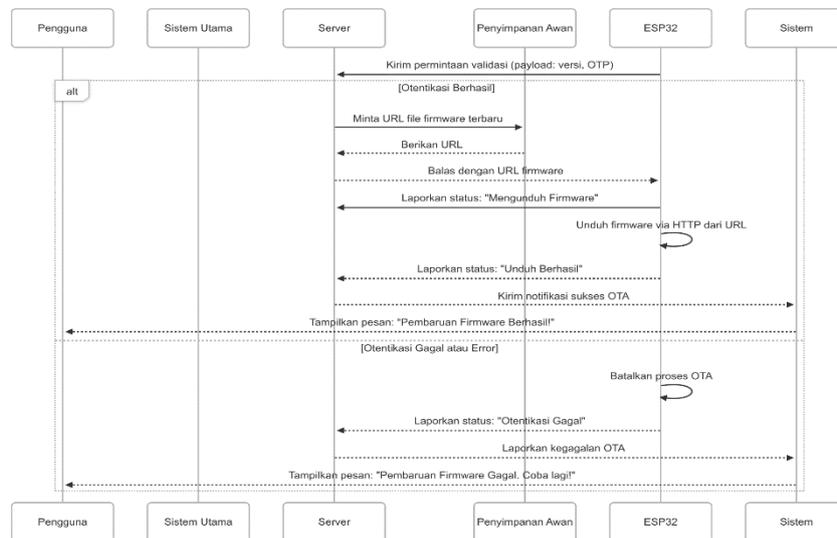


**Gambar 3.** Alur Pengunggahan File

Setelah firmware berhasil diunggah dan tersimpan di Cloud Storage, perangkat ESP32 perlu diberitahu tentang adanya pembaruan. Gambar 4. Alur Permintaan Pembaruan dan Pemicu OTA menjelaskan bagaimana permintaan pembaruan dimulai dari sisi pengguna dan memicu notifikasi OTA ke perangkat ESP32. Alur ini dimulai ketika seorang User menampilkan opsi pembaruan melalui Sistem, yang kemudian menampilkan antarmuka pembaruan, seperti tombol. Jika Pengguna mengklik tombol "Pembaruan", Sistem akan mengirimkan permintaan pembaruan dengan versi firmware saat ini ke Server (bagian dari Web Service). Server akan meneruskan permintaan tersebut dan menghasilkan Kode One-Time Password (OTP) sebagai mekanisme keamanan tambahan. Kode OTP ini kemudian digunakan untuk memicu notifikasi OTA ke perangkat ESP32 dengan menyertakan payload berupa versi firmware terbaru dan OTP tersebut. Jika Pengguna tidak mengklik tombol pembaruan, Sistem akan menampilkan pesan informasi atau instruksi yang relevan.



**Gambar 4.** Alur Permintaan Pembaruan dan Pemicu OTA



**Gambar 5.** Alur Otentikasi dan Pembaruan Firmware ESP32

Tahap terakhir dalam proses implementasi adalah otentikasi perangkat ESP32 dan proses pembaruan firmware itu sendiri. Alur Otentikasi dan Pembaruan Firmware ESP32. Ketika ESP32 menerima notifikasi pemicu OTA, ESP32 akan mengirimkan permintaan validasi (berisi payload versi dan OTP) ke Server. Jika otentikasi berhasil, Server akan meminta URL file firmware terbaru dari Penyimpanan Awan, lalu membalas dengan URL tersebut ke ESP32. Server juga melaporkan status "Mengunduh Firmware" ke Sistem Utama untuk dipantau Pengguna. ESP32 kemudian akan mengunduh firmware langsung via HTTP dari URL yang diberikan.

Setelah pengunduhan berhasil, ESP32 akan melaporkan status "Unduh Berhasil" ke Server. Server kemudian mengirimkan notifikasi sukses OTA ke Sistem Utama, yang pada gilirannya akan menampilkan pesan "Pembaruan Firmware Berhasil" kepada Pengguna. Namun, jika otentikasi gagal atau terjadi error, Server akan membatalkan

proses OTA dan melaporkan status "Otentikasi Gagal" atau "Kegagalan OTA" ke Sistem Utama. Sistem Utama kemudian akan menampilkan pesan "Pembaruan Firmware Gagal. Coba lagi!" kepada Pengguna. Alur ini memastikan bahwa hanya perangkat yang sah yang dapat mengunduh dan menginstal pembaruan firmware, menjaga keamanan dan integritas sistem.

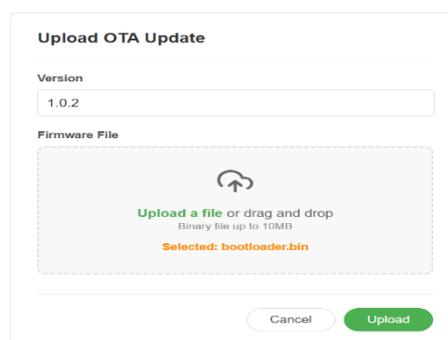
### Pengujian Sistem Over-the-Air

Setelah tahapan perancangan dan implementasi sistem selesai, langkah krusial berikutnya adalah melakukan pengujian menyeluruh terhadap fungsionalitas pembaruan firmware Over-the-Air (OTA) pada perangkat ESP32. Pengujian ini esensial untuk memverifikasi bahwa sistem yang dibangun dapat melakukan distribusi firmware secara efisien serta menjaga keaslian dan integritas file pembaruan.

- a. Dilakukan pengujian koneksi dan pengunggahan firmware ke ESP32 melalui OTA.
- b. Dilakukan pengujian 10x pembaruan, lalu dicatat tingkat keberhasilan dan waktu pembaruan.

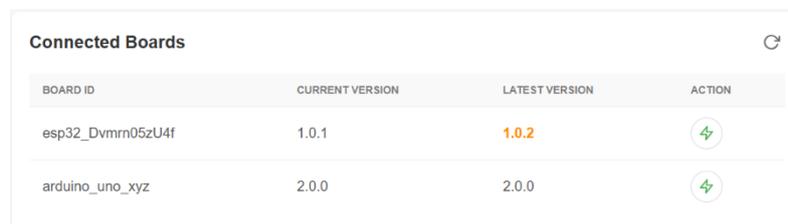
## 4. HASIL DAN PEMBAHASAN

Pembaruan firmware Over-the-Air (OTA) untuk perangkat ESP32 diawali dengan tahapan kompilasi kode program menggunakan Arduino IDE. Developer menyiapkan source code aplikasi pada lingkungan pengembangan Arduino IDE, memastikan semua library yang diperlukan terinstal, dan konfigurasi board telah sesuai, seperti pemilihan "ESP32 Dev Module". Setelah source code selesai disiapkan, verifikasi dan kompilasi dilakukan melalui fitur "Verify" dan "Upload" pada Arduino IDE. Proses "Upload" ini secara otomatis mengkompilasi kode program menjadi file biner (.bin), yang merupakan firmware yang siap untuk diunggah ke perangkat ESP32. File biner inilah yang akan menjadi payload firmware dalam sistem pembaruan OTA.



Gambar 6. Interface Upload File OTA

Setelah firmware berhasil dikompilasi, tahapan selanjutnya adalah pengunggahan file firmware tersebut ke dalam sistem. Proses ini berkorespondensi dengan "Alur Pengunggahan File" (Gambar 3). Antarmuka pengguna untuk pengunggahan file OTA, seperti yang ditunjukkan pada "Interface Upload File OTA" (Gambar 6), menampilkan bidang input untuk versi firmware (contohnya "1.0.2") dan area pemilihan file firmware (.bin) yang telah dikompilasi sebelumnya. Setelah file dipilih, pengguna mengklik tombol "Upload". Antarmuka ini merepresentasikan secara visual tahapan "Pengguna Mengunggah Berkas OTA" dan "Sistem Utama Menampilkan Jendela Konfirmasi" dalam Gambar 3. Jika pengunggahan berhasil, Sistem Utama akan mengirimkan file biner tersebut ke Cloud Storage, sesuai dengan alur pada Gambar 3.



BOARD ID	CURRENT VERSION	LATEST VERSION	ACTION
esp32_Dvmrn05zU4f	1.0.1	1.0.2	
arduino_uno_xyz	2.0.0	2.0.0	

**Gambar 7.** Interface Proses Update

Setelah firmware baru berhasil diunggah dan disimpan di Cloud Storage, perangkat ESP32 perlu menerima notifikasi mengenai ketersediaan pembaruan. Proses ini diilustrasikan oleh "Alur Permintaan Pembaruan dan Pemicu OTA" (Gambar 4) dan "Alur Otentikasi dan Pembaruan Firmware ESP32" (Gambar 5), serta direpresentasikan secara visual oleh "Interface Proses Update" (Gambar 7). Pada Gambar 7, dapat dilihat daftar "Connected Boards" yang terhubung ke sistem. Contohnya, perangkat dengan "BOARD ID" esp32\_DVmrn05zJU4f memiliki "CURRENT VERSION" 1.0.1 dan "LATEST VERSION" 1.0.2, mengindikasikan ketersediaan pembaruan. Tombol dengan ikon petir pada kolom "ACTION" di Gambar 7 berfungsi sebagai pemicu pembaruan OTA. Ketika pengguna mengklik tombol ini, Sistem (Web Service), sesuai dengan Gambar 4, akan mengirimkan permintaan pembaruan ke Server. Server kemudian menghasilkan One-Time Password (OTP) dan memicu notifikasi OTA ke perangkat ESP32. Notifikasi ini mencakup payload berupa versi firmware terbaru dan OTP.

Selanjutnya, mengikuti alur pada Gambar 5, saat ESP32 menerima notifikasi pemicu OTA, ia akan mengirimkan permintaan validasi ke Server. Jika otentikasi berhasil, Server akan menyediakan URL file firmware dari Cloud Storage, dan ESP32 akan mengunduh firmware secara langsung. Status pembaruan, seperti "Mengunduh Firmware" atau "Unduh Berhasil", akan dilaporkan kembali ke Sistem Utama dan ditampilkan kepada

pengguna. Proses ini pada akhirnya akan memperbarui "CURRENT VERSION" pada Gambar 7 menjadi 1.0.2 jika pembaruan berhasil. Keseluruhan alur ini mengotomatiskan distribusi firmware dan menjamin hanya perangkat yang terotentikasi yang dapat menerima pembaruan.

BOARD ID	CURRENT VERSION	LATEST VERSION	ACTION
esp32_Dvmrm05zU4f	1.0.2	1.0.2	⚡
arduino_uno_xyz	2.0.0	2.0.0	⚡

**Gambar 8.** Interface Proses Setelah Update

Setelah proses pembaruan selesai, antarmuka sistem akan merefleksikan status firmware terbaru. Gambar 8, "Interface Proses Setelah Update", menunjukkan tampilan yang mengkonfirmasi keberhasilan pembaruan. Pada gambar tersebut, terlihat bahwa "BOARD ID" esp32\_DVmmr05zJU4f kini memiliki "CURRENT VERSION" 1.0.2 dan "LATEST VERSION" juga 1.0.2. Ini menandakan bahwa firmware perangkat telah berhasil diperbarui ke versi terbaru dan tidak ada lagi pembaruan yang tertunda untuk perangkat tersebut. Untuk mengevaluasi keandalan sistem OTA yang dirancang, dilakukan pengujian berulang sebanyak 10 kali pembaruan. Hasil pengujian ini dapat di lihat pada Tabel 1 dibawah ini.

**Tabel 1.** Hasil Pengujian OTA

Test	Update Time	Status
1	20.51 seconds	SUCCESS
2	22.96 seconds	SUCCESS
3	23.87 seconds	SUCCESS
4	20.94 seconds	SUCCESS
5	21.33 seconds	SUCCESS
6	21.78 seconds	SUCCESS
7	22.12 seconds	SUCCESS
8	21.03 seconds	SUCCESS
9	21.50 seconds	SUCCESS
10	20.44 seconds	SUCCESS

Berdasarkan hasil tabel diatas, dapat diamati bahwa kesepuluh percobaan pembaruan firmware OTA berhasil (Status: SUCCESS). Waktu pembaruan bervariasi antara 20.44 hingga 23.87 detik, menunjukkan konsistensi dalam performa proses update. Tingkat keberhasilan 100% dari pengujian ini mengindikasikan bahwa sistem pembaruan OTA yang dirancang berfungsi dengan sangat baik dan dapat diandalkan untuk mendistribusikan firmware ke perangkat ESP32 secara nirkabel. Konsistensi waktu dan status keberhasilan ini menegaskan efisiensi dan robustitas mekanisme OTA yang diimplementasikan, mendukung tujuan penelitian untuk menciptakan sistem pembaruan firmware yang aman dan andal.

## **5. KESIMPULAN DAN SARAN**

### **Kesimpulan**

Penelitian ini berhasil merancang dan mengimplementasikan sistem pembaruan firmware Over-the-Air (OTA) berbasis layanan cloud untuk perangkat ESP32. Sistem ini terbukti efisien dalam melakukan pembaruan firmware secara nirkabel, aman melalui mekanisme otentikasi OTP, dan andal dengan tingkat keberhasilan 100% dari 10 kali pengujian, serta rata-rata waktu pembaruan sekitar 21.608 detik. Keberhasilan ini secara efektif mengatasi keterbatasan pembaruan konvensional dan mendukung manajemen firmware yang lebih baik pada perangkat IoT berbasis ESP32.

### **Saran**

Penulis memberikan beberapa saran untuk penelitian selanjutnya dan pengembangan lebih lanjut, yakni mengembangkan sistem OTA dengan menambahkan mekanisme keamanan lanjutan seperti digital signature dan komunikasi terenkripsi menggunakan HTTPS/TLS untuk meningkatkan integritas dan kerahasiaan data. Selain itu, implementasi fitur rollback menggunakan dua partisi firmware (dual partition) perlu dipertimbangkan agar perangkat dapat kembali ke versi sebelumnya jika pembaruan gagal. Pengujian lebih lanjut juga perlu dilakukan dalam kondisi jaringan yang tidak stabil untuk mengevaluasi keandalan sistem di lingkungan nyata. Pengembangan integrasi dengan platform cloud komersial seperti AWS atau Azure dapat memperluas skalabilitas sistem, sementara penggunaan metode delta update dapat meningkatkan efisiensi pembaruan. Terakhir, penambahan fitur dashboard monitoring dan analisis konsumsi daya selama proses OTA juga dapat menjadi arah pengembangan untuk menciptakan sistem yang lebih lengkap dan optimal.

## DAFTAR REFERENSI

- Adani, F., & Salsabil, S. (2021). Internet of Things: Sejarah teknologi dan penerapannya. *Isu Teknologi STT Mandala*, 14(2), 92–99.
- Brightwood, S. (2024, Oktober). The importance of secure firmware updates in maintaining system integrity.
- Fathor Azhar, M., & Nurpulaela, L. (2024). Implementasi penggunaan ESP32 sebagai IoT pada project smart charger di PT. Pasifik Satelit Nusantara Bekasi. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 8(4), 7248–7253. <https://doi.org/10.36040/jati.v8i4.10201>
- Hariato, D. (n.d.). *Development and evaluation of an ESP32-based temperature and humidity control unit for textile storage* (pp. 1–19).
- Ikhsani, M. F., Budi, A. S., & Tim Penulis. (2023). Implementasi over the air update firmware secara massal pada ESP8266 untuk perangkat Internet of Things. *Jurnal Teknologi dan Sistem Komputer*, 1(1), 1–7. (Catatan: Nama jurnal disesuaikan jika tersedia, atau bisa ditambahkan jika Anda punya datanya)
- Lestari, D. A., & Prasetyo, H. (2021). Strategi pemerintah desa dalam meningkatkan pelayanan publik berbasis digital. *Jurnal Ilmu Administrasi Publik*, 18(2), 112–123.
- Nugroho, R. (2020). *Public policy: Dinamika kebijakan, analisis kebijakan, manajemen kebijakan*. Jakarta: Elex Media Komputindo.
- Paramartha, I. G. N. D., Kurniawan, I. N. H., Subiksa, G. B., & Kartika, A. S. (2021). Arsitektur Internet of Things (IoT) berskala industri dengan fitur over the air update. *TIERS Information Technology Journal*, 2(2), 31–36. <https://doi.org/10.38043/tiers.v2i2.3311>
- Pradita, U., Sangereng, C., Dua, K. K., & Tangerang, K. (2025). Pemanfaatan cloud computing untuk meningkatkan efisiensi bisnis pada platform Google Cloud. *Jurnal Teknologi Informasi*, 9(1), 84–89. (Catatan: Nama jurnal disesuaikan jika ada data resminya)
- Yusri, I. K., Kasim, K., & Akbar, A. M. (2022). Penerapan Hypertext Transfer Protocol Web Server untuk over-the-air auto update firmware pada perangkat IoT. *Elektron: Jurnal Ilmiah*, 14(2), 67–71. <https://doi.org/10.30630/eji.14.2.298>