PROCEEDING NATIONAL CONFERENCE OF RESEARCH AND COMMUNITY SERVICE SISI INDONESIA
2025
https://conference.sinesia.id/ncrcs-sinesia

Pembaruan Firmware secara Over the Air pada Banyak Perangkat IoT berbasiskan AsyncElegantOTA dan AutoIt

Over-the-Air Firmware Update for Multiple IoT Devices Using AsyncElegantOTA and AutoIt

Hendra Rahmawan[1], Auriza Rahmad Akbar[2], Fathan Mubina Sarjoko[3], Sri Wahjuni[4]
E-mail Korespondensi
hrahmawan@apps.ipb.ac.id[1], auriza@apps.ipb.ac.id[2], fmsarjoko@gmail.com[3], my_juni04@apps.ipb.ac.id[4]
Institut Pertanian Bogor

Info Article
|Submitted: 20 Mei 2025 **|Revised**: 25 May 2025 **|Accepted**: 9 June 2025
How to cite: Hendra Rahmawan, etc., "Pembaruan Firmware secara Over the Air pada Banyak Perangkat IoT berbasiskan AsyncElegantOTA dan AutoIt", *Proceeding National Conference Of Research And Community Service Sisi Indonesia*, 2025, P. 162-175.

ABSTRACT

Objective: The increasing adoption of Internet of Things (IoT) technologies has enabled remote control and connectivity among various devices. These IoT devices require firmware to operate effectively and often need updates to maintain optimal performance. However, updating firmware becomes challenging when devices are deployed in hard-to-reach or remote areas and when updates must be performed across multiple devices. This study aims to develop an Over-the-Air (OTA) firmware update system for multiple ESP32-based IoT devices. Methods: The proposed system integrates the AsyncElegantOTA library with AutoIT scripting to automate and synchronize firmware updates across multiple devices. The research followed four main stages: analysis, design, implementation, and testing. Testing was conducted by uploading five different firmware programs to three ESP32 microcontrollers using a local network environment. Results: All five firmware programs were successfully uploaded to all devices with an average update time of approximately 44–45 seconds. Conclusions: The developed OTA update system is effective for local network environments and it has strong potential for future expansion to broader and more complex wireless networks.

Keyword: *AsyncElegantOTA, AutoIT, ESP32, firmware update, Internet of Things (IoT), Over-the-Air (OTA)*

ABSTRAK

Perkembangan teknologi Internet of Things (IoT) memungkinkan berbagai perangkat terhubung dan dikendalikan dari jarak jauh. Perangkat-perangkat IoT memerlukan *firmware* agar dapat berfungsi sesuai dengan tujuan yang telah ditentukan, dan seringkali memerlukan pembaruan *firmware* agar perangkat-perangkat tersebut dapat berjalan dengan lebih optimal. Pembaruan *firmware* menjadi tantangan ketika lokasi perangkat berada di tempat yang sulit dijangkau dan pembaruan harus menargetkan banyak perangkat. Berdasarkan masalah tersebut, penelitian ini bertujuan untuk mengembangkan sistem pembaruan *firmware* secara *Over-the-Air* (OTA) secara serentak pada banyak perangkat IoT berbasis ESP32. Kombinasi pustaka AsyncElegantOTA dengan perangkat lunak AutoIT digunakan dalam proses pembaruan *firmware* untuk beberapa perangkat IoT secara bersamaan. Tahap penelitian yang dilakukan meliputi tahap analisis, perancangan, implementasi, dan pengujian. Pengujian dilakukan dengan mengunggah lima jenis program pada tiga mikrokontroler ESP32. Hasil pengujian menunjukkan seluruh pembaruan berjalan sukses dengan waktu unggah rata-rata sekitar 44–45 detik. Berdasarkan penelitian ini, sistem ini efektif digunakan dalam lingkungan lokal dan berpotensi dikembangkan lebih lanjut untuk penggunaan jaringan luas.

Kata Kunci: *AsyncElegantOTA, AutoIT, ESP32, pembaruan firmware, Internet of Things (IoT), Over-the-Air (OTA)*

This work is licensed under CC BY-SA 4.0

Pendahuluan

Sistem IoT merupakan salah satu teknologi dalam ranah teknologi informasi dan komunikasi yang menjadi bagian dari berkembangnya revolusi industri 4.0[1]. Sebuah sistem IoT merupakan sebuah jaringan dari perangkat-perangkat dengan skala yang luas. Perangkat-perangkat tersebut memiliki tugas untuk mengumpulkan data dan melakukan tugas-tugas yang ditentukan[2]. Sebuah ekosistem dari IoT terdiri atas perangkat-perangkat cerdas yang memiliki komponen-komponen berupa prosesor, sensor-sensor, dan perangkat komunikasi[3]. Dengan melihat manfaat besar yang ditawarkan oleh teknologi IoT, maka saat ini penerapan teknologi IoT telah menjangkau area yang cukup luas di berbagai bidang seperti perkotaan, transportasi, lingkungan, hingga pertanian[4,5,6,7].

Perangkat-perangkat IoT biasanya berupa perangkat mikrokontroler seperti ESP32[8] yang terhubung dalam sebuah jaringan IoT dan bekerja berdasarkan sebuah program yang tertanam di dalamnya yang dikenal sebagai *firmware*. Pengembangan sebuah *firmware* melalui beberapa tahap yaitu penulisan kode sumber pada sebuah komputer menggunakan sebuah *integrated development environment* (IDE), kompilasi kode sumber menjadi *firmware* yang dapat dijalankan pada mikrokontroler menggunakan sebuah *toolchain*, dan proses *upload firmware* ke mikrokontroler menggunakan kabel *universal serial bus* (USB) yang terhubung ke komputer[9].

Pembaruan *firmware* pada perangkat IoT dapat terjadi kapan saja seiring dengan adanya potensi perubahan tugas dari perangkat IoT. Jika proses *upload firmware* ke perangkat-perangkat IoT dilakukan dengan menggunakan kabel yang terhubung langsung ke komputer maka tentunya perubahan *firmware* akan sulit

---

[1] Alexandros Gazis, "What Is IoT? The Internet of Things Explained," *Academia Letters*, June 7, 2021, https://doi.org/10.20935/al1003.

[2] Asif Ali Laghari et al., "A Review and State of Art of Internet of Things (IoT)," *Archives of Computational Methods in Engineering* (Springer Science and Business Media B.V., May 1, 2022), https://doi.org/10.1007/s11831-021-09622-6.

[3] Mohamed Medhat Gaber et al., "Internet of Things and Data Mining: From Applications to Techniques and Systems," *WIREs Data Mining and Knowledge Discovery* 9, no. 3 (May 9, 2019), https://doi.org/10.1002/widm.1292.

[4] Wan-Soo Kim, Won-Suk Lee, and Yong-Joo Kim, "A Review of the Applications of the Internet of Things (IoT) for Agricultural Automation," *Journal of Biosystems Engineering* 45, no. 4 (December 26, 2020): 385–400, https://doi.org/10.1007/s42853-020-00078-3.

[5] Laghari et al., "A Review and State of Art of Internet of Things (IoT)."

[6] Ibrahim Allafi and Tariq Iqbal, "Design and Implementation of a Low Cost Web Server Using ESP32 for Real-Time Photovoltaic System Monitoring," in *2017 IEEE Electrical Power and Energy Conference (EPEC)* (IEEE, 2017), 1–5, https://doi.org/10.1109/EPEC.2017.8286184.

[7] Jie Lin et al., "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal* 4, no. 5 (October 2017): 1125–42, https://doi.org/10.1109/JIOT.2017.2683200.

[8] Neil Cameron, "ESP32 Microcontroller," 2023, 1–54, https://doi.org/10.1007/978-1-4842-9376-8_1.

[9] Espressif Systems, "ESP-IDF Programming Guide," https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html, May 12, 2025.

dilakukan terhadap perangkat-perangkat yang lokasinya sulit dijangkau seperti di tengah hutan, di tengah lautan, atau di tubuh manusia. Oleh karena itu, teknik pembaruan *firmware* secara *over the air* (OTA) perlu digunakan dalam kondisi pembaruan tersebut[10,11].

Sebelumnya, metode OTA digunakan untuk mendistribusikan pembaruan *software* atau pengaturan konfigurasi ke perangkat-perangkat seperti telepon selular dan *set-top box*. Namun demikian, dengan semakin popularnya penggunaan teknologi *wireless sensor networks* (WSN) dan IoT, metode OTA diadopsi pada kedua teknologi tersebut[12]. Mengingat dalam sebuah sistem IoT biasanya terdapat banyak perangkat, maka diperlukan peningkatan metode OTA agar dapat melakukan pembaruan *firmware* terhadap banyak perangkat IoT secara otomatis[13].

Beberapa penelitian mengenai pengembangan teknik OTA untuk banyak perangkat IoT telah dilakukan seperti pada pengembangan OTA *auto-update* melalui *server hypertext transfer protocol* (HTTP) pada perangkat IoT[14]. Meskipun penelitian tersebut telah berhasil mengembangkan dan mengevaluasi sistem OTA untuk pembaruan *firmware* pada banyak perangkat, namun dalam penelitian tersebut belum dibahas secara rinci pustaka atau *framework* yang digunakan di dalamnya. Arsitektur IoT berskala industri dengan fitur OTA berbasiskan *platform* Thingsboard juga telah diusulkan pada penelitian lainnya[15]. Namun demikian, penggunaan Thingsboard untuk OTA dalam usulan arsitektur tersebut berpotensi mengalami keterbatasan fitur jika menggunakan versi *community edition*. Fitur lengkap pada Thingsboard hanya dapat diperoleh dengan membeli versi *professional edition* atau versi *cloud*.

Berdasarkan penjelasan dari hasil peneletian-penelitian tersebut, maka dapat disimpulkan bahwa diperlukan penelitian yang memberikan metode alternatif OTA dengan menggunakan *tools* yang dapat diakses dengan mudah dan tidak berbayar. Oleh karena itu, pada penelitian ini dikembangkan metode pembaruan

---

[10] Benjamin Bucklin Brown, "Over-the-Air (OTA) Updates in Embedded Microcontroller Applications: Design Trade-Offs and Lessons Learned," *Analog Dialogue Technical Journal* 52, no. 11 (November 2018), https://www.allelcoelec.es/datasheets.b3/ADF7242BCPZ.pdf.

[11] Konstantinos Arakadakis et al., "Firmware Over-the-Air Programming Techniques for IoT Networks - A Survey," *ACM Computing Surveys* 54, no. 9 (December 31, 2022): 1–36, https://doi.org/10.1145/3472292.

[12] Francesc Campà Aubets, "Design of an Arduino Shield for OTA Programming" (Universitat Politècnica de Catalunya, 2013), http://hdl.handle.net/2099.1/17305.

[13] Penghui Shen et al., "OTA Measurement for IoT Wireless Device Performance Evaluation: Challenges and Solutions," *IEEE Internet of Things Journal* 6, no. 1 (February 2019): 1223–37, https://doi.org/10.1109/JIOT.2018.2868787.

[14] Iin Karmila Yusri, Kasim Kasim, and Andi Muhammad Akbar, "Penerapan Hypertext Transfer Protocol Web Server Untuk Over-The-Air Auto Update Firmware Pada Perangkat IoT," *Elektron : Jurnal Ilmiah*, December 24, 2022, 67–71, https://doi.org/10.30630/eji.14.2.298.

[15] I Gusti Ngurah Darma Paramartha et al., "Arsitektur Internet of Things (IoT) Berskala Industri Dengan Fitur Over The Air Update," *TIERS Information Technology Journal* 2, no. 2 (December 29, 2021): 31–36, https://doi.org/10.38043/tiers.v2i2.3311.

OTA dengan memadukan *tools* AsyncElegantOTA dan AutoIT yang bersifat tidak berbayar.

Pada penelitian ini pustaka AsyncElegantOTA digunakan untuk proses *upload firmware* ke perangkat IoT melalui protokol HTTP[16] dan AutoIT digunakan untuk otomasi *upload firmware* ke beberapa perangkat IoT[17,18]. Perangkat IoT yang digunakan pada penelitian ini adalah ESP32 yang merupakan perangkat *system on chip* (SoC) yang di dalamnya terintegrasi prosesor 32-bit Xtensa LX6 *dual core*, Wi-Fi dengan frekuensi 2.4 GHz, *Bluetooth*, dan sejumlah *port input-ouput* serbaguna.

Metode Penelitian

Pada penelitian ini, metode pembaruan *firmware* secara OTA dikembangkan dalam tiga tahap yaitu analisis, perancangan, implementasi, dan pengujian. Berikut penjelasan ketiga tahap tersebut.

A. Analisis

Pada tahap ini dilakukan identifikasi kebutuhan fungsional sistem, kebutuhan perangkat keras dan kebutuhan perangkat lunak yang diperlukan dalam pengembangan metode pembaruan *firmware* secara OTA. Analisis kebutuhan fungsional dilakukan dengan mengobservasi fitur-fitur yang umunya ada dalam proses OTA pada perangkat IoT yaitu:

1. Menyediakan antarmuka untuk mengunggah *firmware* ke perangkat IoT. Antarmuka dapat berbentuk halaman web.
2. Melakukan pembaruan *firmware* terotomasi ke banyak perangkat IoT.

Adapun kebutuhan perangkat keras dan perangkat lunak agar dapat memenuhi kebutuhan fungsional tersebut adalah sebagai berikut:

1. Kebutuhan Perangkat Keras
   - ESP32 sebanyak tiga unit, berfungsi sebagai target pembaruan firmware. Jumlah unit ESP32 dapat disesuaikan dengan ketersediaan perangkat.
   - *Laptop*/komputer sebagai *client* untuk menjalankan AutoIt dan Arduino IDE.
   - *Router*/*access point* untuk membangun jaringan WiFi.
   - Kabel USB untuk konfigurasi awal ESP32.

2. Kebutuhan Perangkat Lunak
   - Arduino IDE 2.1.1 untuk menulis dan mengunggah program ke ESP32.
   - Beberapa pustaka ESP32 yaitu WiFi.h, AsyncTCP.h, ESPAsyncWebServer.h, dan AsyncElegantOTA.h untuk mengimplementasikan proses pembaruan *firmware* secara OTA.

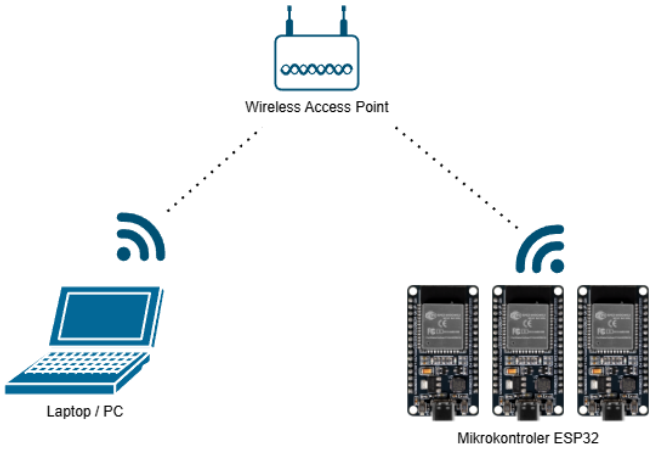
[16] Fernando G. Tinetti, Alejo A Santi, and Mariano Méndez, "Continuous Integration and Continuous Delivery with Microcontrollers Applications," in *2023 International Conference on Computational Science and Computational Intelligence (CSCI)* (IEEE, 2023), 1085–90, https://doi.org/10.1109/CSCI62032.2023.00179.
[17] de Fez Laso and Emilio Aristides, *Instant AutoIt Scripting* (Packt Publishing, 2013).
[18] Jonathan Bennet and AutoIt Team, "AutoIT," https://www.autoitscript.com/autoit3/docs/introduction.htm, 2022.

- AutoIt v3 untuk membuat skrip otomatisasi proses unggah *firmware*.
- *Web browser* (Microsoft Edge) untuk memuat antarmuka pengunggahan OTA.
- Sistem operasi minimal Windows 10.

B. Perancangan

1. Perancangan Topologi Jaringan

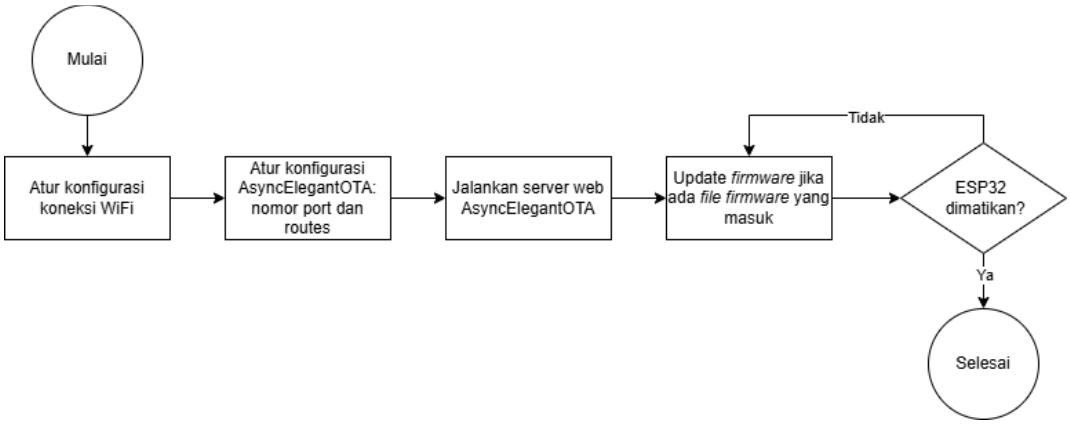
Topologi jaringan merupakan bagian pertama yang dirancang pada penelitian ini. Topologi jaringan yang digunakan adalah topologi *star* dengan menggunakan jaringan WiFi. Jaringan WiFi terdiri atas satu *access point* yang menghubungkan komputer *client* yang menjalankan skrip AutoIt dengan beberapa perangkat ESP32 yang bertindak sebagai *server* OTA (Gambar 1). Melalui jaringan WiFi, setiap ESP32 akan mendapatkan alamat IP dinamis dari *access point* melalui protokol DHCP. Alamat tersebut akan digunakan untuk mengakses halaman *web* pembaruan *firmware*.



Gambar 1 Topologi jaringan WiFi yang digunakan

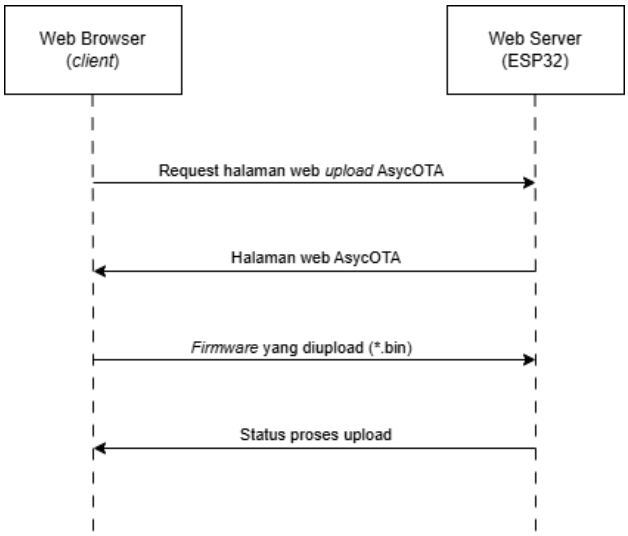
2. Perancangan Alur Update *Firmware* secara OTA

Agar dapat dilakukan *upload firmware* secara OTA maka perlu dilakukan tahap persiapan terlebih dahulu. Pada tahap tersebut dilakukan proses *upload firmware* dasar yang menjalankan beberapa fungsi yang diperlukan untuk *upload firmware* secara OTA pada tahap selanjutnya. Fungsi-fungsi utama yang diperlukan adalah pengaturan koneksi WiFi serta pengaturan dan pemanggilan fungsi AsyncElegantOTA yang akan menyediakan fungsionalitas OTA. Diagram alir proses *update firmware* di dalam perangkat IoT dinyatakan pada Gambar 2. Tahap ini harus dilakukan dengan menghubungkan perangkat IoT ke komputer langsung dengan menggunakan kabel USB.

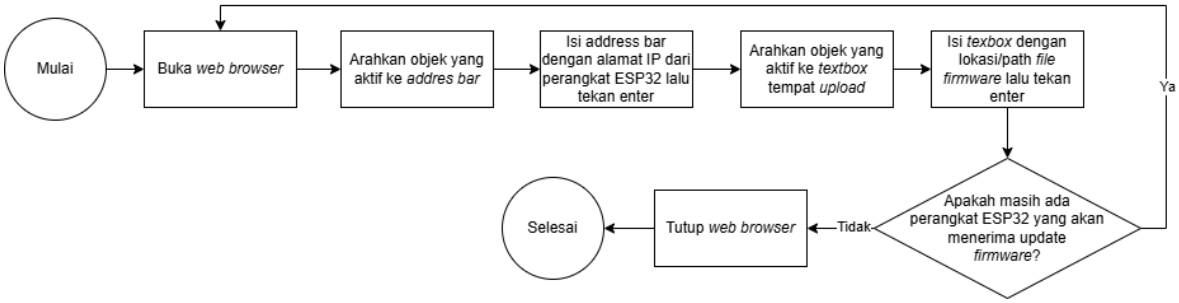


Gambar 2 Diagram alir proses aktivasi fungsi OTA pada perangkat IoT.

Tahap selanjutnya adalah tahap *upload firmware* ke perangkat ESP32 secara OTA. Pada tahap ini *firmware* sudah dapat diunggah ke perangkat IoT melalui aplikasi *web* yang sudah berjalan di perangkat IoT. Alur proses *upload firmware* ke perangkat IoT, dalam hal ini ESP32, digambarkan dalam diagram sekuen pada Gambar 3. Pada tahap ini digunakan perangkat otomasi, yaitu AuotIT, untuk mengotomasi proses *upload firmware* ke beberapa perangkat IoT. Algoritma *update firmware* dengan perangkat otomasi AutoIT dinyatakan dalam diagram alir pada Gambar 4.



Gambar 3 Diagram sekuan proses *upload firmware* ke ESP32



Gambar 4 Diagram alir proses otomasi *upload firmware* ke perangkat IoT ESP32

C. Implementasi

Pada tahap ini algoritma yang telah dirancang dalam bentuk diagram alir selanjutnya diimplementasikan menjadi kode program. Kode program yang dibuat yaitu program dalam bentuk *fimware* dasar (implementasi dari Gambar 2) yang diunggah ke perangkat IoT ESP32, yang bertugas sebagai *web server*. Kode program berikutnya yang dibuat adalah skrip AutoIT (implementasi dari Gambar 4) yang dijalankan di komputer untuk melakukan otomasi *upload firmware* ke perangkat ESP32 melalui sebuah *web browser*, dalam hal ini Microsoft Edge.

Potongan kode program *firmware* dasar dinyatakan pada Gambar 5. Kode program tersebut berisi penggunaan pustaka-pustaka terkait OTA, penyediaan antarmuka *web* yang dapat diakses melalui URL http://[alamat IP]/update, dan menangani koneksi asinkron serta proses unggah *file firmware*. *Firmware* dasar tersebut diunggah ke perangkat IoT ESP32 menggunakan perangkat lunak Arduino IDE melalui kabel USB.

Adapun potongan kode program AutoIT seperti dinyatakan pada Gambar 6 digunakan untuk otomasi pengiriman *firmware* melalui halaman web seperti ditampilkan pada Gambar 7. Kode program tersebut berisi otomasi pemilihan objek *address bar* pada *web browser* dan objek *text box* pada halaman web serta pengiriman *file* melalui otomasi penekanan tombol Tab dan Enter. Potongan kode program AutoIT tersebut perlu diulang beberapa kali sejumlah perangkat IoT ESP32 yang memerlukan *update* firmware. Program ini disimpan di komputer dan dijalankan ketika akan melakukan *update firmware* secara OTA ke perangkat-perangkat IoT ESP32. Dalam penelitian ini, proses pengiriman *firmware* ke perangkat-perangkat IoT masih dilakukan tanpa menggunakan konstruksi perulangan (*looping*) karena sifat dari program ini masih dalam bentuk prototipe untuk menunjukan perintah-perintah dasar apa saja yang perlu digunakan pada saat mengirim *firmware* secara OTA ke perangkat-perangkat IoT ESP32. Untuk peningkatan selanjutnya, perintah-perintah dasar ini dapat diotomasi dengan menggunakan konstruksi perulangan. Kode program AutoIT ini mengirimkan *firmware* ke perangkat-perangkat IoT melalui jaringan WiFi yang menghubungkan komputer dengan perangkat-perangkat IoT tersebut.

D. Pengujian

Pengujian dilakukan dengan mengunggah lima variasi *firmware* ke tiga perangkat ESP32 secara OTA masing-masing diulang sebanyak 5 kali lalu dicatat keberhasilannya. Deskripsi kelima *firmware* tersebut dinyatakan pada Tabel 1.

```
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebSrv.h>
#include <AsyncElegantOTA.h>

//Atur ssid dan password sesuai dengan jaringan WiFi yang akan digunakan
const char* ssid = "*******";
const char* password = "*******";

//Buat objek server
AsyncWebServer server(80) ;

void setup(void) {
//Hubungkan perangkat ESP32 ke wireless access point
WiFi.begin(ssid, password) ;

//Tunggu sampai ESP32 terhubung ke jaringan WiFi
while (WiFi.status() !=WL_CONNECTED) {
delay(500) ;
}

//Atur routes
server.on("/", HTTP_GET, [] (AsyncWebServerRequest *request)
{request->send(200, "text/plain", "ESP OTA (Over The Air).") ;} );

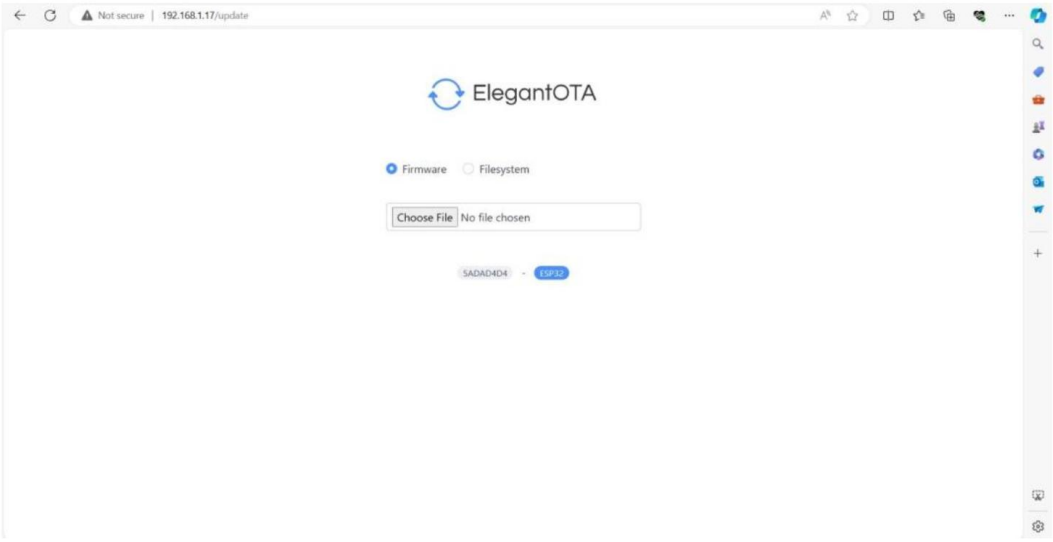
//Aktifkan server
AsyncElegantOTA.begin(&server);
server.begin();
}

//Set ESP32 agar terus menerus memeriksa apakah ada permintaan update firmware
void loop(void) {
       AsyncElegantOTA.loop() ;
```

Gambar 5 Potongan kode program *firmware* dasar.

```
Run("C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe
-test-type --ignore-certificate-errors --guest --user-
datadir=C:\tempsession\session" & Random(10000,99999,1) & "\
http://192.168.1.117/update")
Sleep(4000)
Send("{TAB}")
Sleep(2500)
Send("{TAB}")
Sleep(2500)
Send("{Enter}")
Sleep(2500)
Send("E:\Implementasi skripsi
OTA\OTA\Repeater\build\esp32.esp32.esp32\Repeater.ino.bin")
Send("{Enter}")
```

Gambar 6 Potongan kode program AutoIT untuk *upload firmware* ke perangkat ESP32



Gambar 7 Tampilan *landing web page* AsyncElegantOTA

Tabel 1 Lima variasi *firmware* yang diunggah ke perangkat ESP32 secara OTA

No	Nama Program	Deskripsi Fungsi
1	Blinking	Lampu LED berkedip perlahan.
2	Blinking Cepat	Lampu LED berkedip cepat.
3	LED Menyala	LED menyala terus-menerus.
4	WiFi Scanner	ESP32 memindai jaringan WiFi di sekitar.
5	Repeater	ESP32 menjalankan program loop yang terus-menerus.

Selain menguji keberhasilan upload *firmware* ke perangkat-perangkat ESP32, pada penelitian ini juga diukur waktu *update firmware* yang diperlukan. Waktu pembaruan dihitung dari awal proses unggah *firmware* hingga ESP32 berhasil *reboot* dan menjalankan program. Pengukuran dilakukan menggunakan *stopwatch*.

Hasil dan pembahasan

Pada bagian ini dipaparkan hasil-hasil pengujian yang diperoleh beserta pembahasannya. Sebagaimana telah dijelaskan sebelumnya, pengujian dilakukan dalam dua aspek yaitu fungsionalitas update *firmware* secara OTA dan waktu *update firmware* yang diperlukan.

A. Pengujian Fungsionalitas Update Firmware secara OTA

Berdasarkan pengujian yang telah dilakukan diperoleh hasil seperti dinyatakan pada Tabel 2. Hasil pengujian tersebut menunjukkan bahwa seluruh percobaan mencapai tingkat keberhasilan 100%. Ini membuktikan bahwa sistem pembaruan OTA yang dikembangkan menggunakan AsyncElegantOTA dan AutoIt dapat bekerja secara andal dan konsisten untuk memperbarui beberapa perangkat IoT ESP32 tanpa mengalami kegagalan.

Tabel 2 Hasil pengujian fungsionalitas update *firmware* secara OTA

Nama Program	Jumlah ESP32	Jumlah Percobaan	Keberhasilan (%)
Blinking	3	5	100%
Blinking Cepat	3	5	100%
LED Menyala	3	5	100%
WiFi Scanner	3	5	100%
Repeater	3	5	100%

B. Pengujian Waktu Update Firmware secara OTA

Pada pengujian ini berhasil diperoleh waktu *update firmware* secara OTA sebagaimana disajikan pada Tabel 3. Tabel 3 menampilkan rata-rata waktu pembaruan program secara OTA untuk lima jenis *firmware* yang diuji pada perangkat ESP32, yang masing-masing diulang sebanyak 5 kali percobaan. Ukuran *file firmware* bervariasi, mulai dari 232 KB hingga 699 KB, dengan waktu unggah rata-rata berkisar antara 43,77 detik hingga 45,67 detik. Meskipun terdapat perbedaan ukuran file, hasil menunjukkan bahwa perbedaan waktu unggah tidak terlalu signifikan. Hal ini menunjukkan bahwa sistem pembaruan OTA yang dikembangkan memiliki kinerja yang cukup stabil dan efisien, serta mampu menangani variasi ukuran file tanpa memengaruhi durasi pembaruan secara drastis.

Tabel 3 Hasil pengujian waktu update *firmware* secara OTA

Nama Program	Ukuran File (KB)	Rata-rata Waktu Upload (detik)
Blinking	232	44,58
Blinking Cepat	232	45,67
LED Menyala	232	45,44
WiFi Scanner	699	44,76
Repeater	263	43,77

C. Perbandingan dengan Penelitian Sebelumnya

Jika dibandingkan dengan beberapa penelitian sebelumnya, seperti pada pengembangan OTA *auto-update* melalui *server hypertext transfer protocol* (HTTP) pada perangkat IoT[19], meskipun telah berhasil mengembangkan dan mengevaluasi sistem OTA untuk pembaruan *firmware* pada banyak perangkat, namun dalam penelitian tersebut belum dibahas secara rinci pustaka atau *framework* yang digunakan di dalamnya. Pada penelitian ini, pustaka dan *tools* yang digunakan telah dijelaskan, yaitu AsyncElegantOTA dan AutoIT.

[19] Yusri, Kasim, and Akbar, "Penerapan Hypertext Transfer Protocol Web Server Untuk Over-The-Air Auto Update Firmware Pada Perangkat IoT."

Penelitian lainnya adalah mengenai pengembangan arsitektur IoT berskala industri dengan fitur OTA berbasiskan *platform* Thingsboard[20]. Dalam penelitian tersebut, penggunaan Thingsboard untuk OTA berpotensi mengalami keterbatasan fitur jika menggunakan versi *community edition*. Fitur lengkap pada Thingsboard hanya dapat diperoleh dengan membeli versi *professional edition* atau versi *cloud*. Pada penelitian ini, pustaka dan *tools* yang digunakan, yaitu AsyncElegantOTA dan AutoIT, seluruhnya bersifat gratis dengan seluruh fitur-fiturnya dapat digunakan.

Penelitian yang mengembangkan *update script* pada modul *Smart Indoor Farming System* berbasiskan IoT[21] menggunakan OTA dengan metode *pull*. Dengan metode tersebut, *firmware* harus diletakkan pada sebuah *server*. Selanjutnya, perangkat-perangkat IoT akan memeriksa apakah terdapat versi *firmware* terbaru di *server* tersebut. Jika ditemukan *firmware* versi terbaru, maka perangkat-perangkat IoT akan mengunduh *firmware* tersebut dan memasangnya. Pada penelitian ini, *firmware* tidak perlu diletakkan di sebuah *server* khusus, melainkan cukup disimpan di komputer lokal, dan akan dikirimkan jika perangkat-perangkat IoT memerlukan pembaruan *firmware*. Metode ini dikenal juga dengan metode *push*.

Penelitian berikutnya adalah pengembangan *deployment system* untuk TinyML *models* pada perangkat-perangkat IoT secara OTA[22]. Fokus pada penelitian tersebut adalah pada pembaruan model *machine learning*, *firmware re-flashing*, *firmware re-configuration*, dan *firmware re-purposing*. Adapun fokus pada penelitian ini adalah pada penggantian *firmware* (*replacing*), meskipun dapat juga digunakan untuk *firmware re-flashing*.

Penutup

Pada penelitian ini berhasil dikembangkan metode pembaruan *firmware* secara OTA untuk banyak perangkat IoT berbasis ESP32 menggunakan kombinasi AsyncElegantOTA dan AutoIt sehingga pembaruan *firmware* dapat dilakukan tanpa harus menggunakan koneksi kabel USB namun cukup dengan menggunakan jaringan WiFi. Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa teknik ini mampu memperbarui *firmware* pada beberapa perangkat IoT ESP32 dengan tingkat keberhasilan 100% pada lima jenis *firmware* yang berbeda. Hal ini menunjukkan bahwa metode yang digunakan efektif dalam melakukan pembaruan perangkat di lokasi yang sulit dijangkau. Selain fungsionalitas yang baik, kinerja metode ini juga menunjukkan hasil yang stabil dengan waktu unggah rata-rata yang relatif konsisten, meskipun ukuran *file firmware* bervariasi. Metode ini dapat menjadi alternatif untuk diterapkan pada

[20] Paramartha et al., "Arsitektur Internet of Things (IoT) Berskala Industri Dengan Fitur Over The Air Update."
[21] Tomi Aditya and Oktaf Agni Dhewa, "Design and Implementation of Update Script in the IoT-Based Smart Indoor Farming System Module at PT Inastek Using Over-the-Air Programming," *Media of Computer Science* 1, no. 2 (December 2024): 129–38, https://epublikasi.digitallinnovation.com/index.php/mcs/article/view/201/110.
[22] Bharath Sudharsan et al., "OTA-TinyML: Over the Air Deployment of TinyML Models and Execution on IoT Devices," *IEEE Internet Computing* 26, no. 3 (May 1, 2022): 69–78, https://doi.org/10.1109/MIC.2021.3133552.

berbagai skenario penerapan IoT yang memerlukan pembaruan perangkat lunak secara berkala.

Saran

Penelitian ini masih memiliki peluang untuk dikembangkan lebih lanjut, terutama dalam hal peningkatan tingkat otomatisasi dan skalabilitas pembaruan *firmware* secara OTA. Implementasi metode pembaruan dapat diperluas dengan menggunakan jaringan yang lebih fleksibel seperti layanan *cloud* agar proses pembaruan dapat diakses dari lokasi manapun, tidak hanya terbatas pada jaringan lokal. Selain itu disarankan juga untuk mengeksplorasi penggunaan metode *multicast*, yang memungkinkan pengiriman data secara simultan ke banyak perangkat secara efisien dalam satu transmisi, sehingga dapat mengurangi beban jaringan dan dapat mempercepat proses pembaruan *firmware* pada skala besar. Teknologi REST API juga dapat digunakan untuk mengatur proses pembaruan melalui permintaan HTTP yang lebih fleksibel sehingga memungkinkan pembaruan dilakukan melalui berabagai macam antarmuka *front-end*.

Ucapan Terimakasih

Terima kasih kami ucapkan kepada Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Institut Pertanian Bogor (IPB) yang telah memfasilitasi penelitian ini.

Daftar Pustaka

Aditya, Tomi, and Oktaf Agni Dhewa. "Design and Implementation of Update Script in the IoT-Based Smart Indoor Farming System Module at PT Inastek Using Over-the-Air Programming." *Media of Computer Science* 1, no. 2 (December 2024): 129–38. https://epublikasi.digitallinnovation.com/index.php/mcs/article/view/201/110.

Allafi, Ibrahim, and Tariq Iqbal. "Design and Implementation of a Low Cost Web Server Using ESP32 for Real-Time Photovoltaic System Monitoring." In *2017 IEEE Electrical Power and Energy Conference (EPEC)*, 1–5. IEEE, 2017. https://doi.org/10.1109/EPEC.2017.8286184.

Arakadakis, Konstantinos, Pavlos Charalampidis, Antonis Makrogiannakis, and Alexandros Fragkiadakis. "Firmware Over-the-Air Programming Techniques for IoT Networks - A Survey." *ACM Computing Surveys* 54, no. 9 (December 31, 2022): 1–36. https://doi.org/10.1145/3472292.

Bennet, Jonathan, and AutoIt Team. "AutoIT." https://www.autoitscript.com/autoit3/docs/introduction.htm, 2022.

Brown, Benjamin Bucklin. "Over-the-Air (OTA) Updates in Embedded Microcontroller Applications: Design Trade-Offs and Lessons Learned." *Analog Dialogue Technical Journal* 52, no. 11 (November 2018). https://www.allelcoelec.es/datasheets.b3/ADF7242BCPZ.pdf.

Cameron, Neil. "ESP32 Microcontroller," 1–54, 2023. https://doi.org/10.1007/978-1-4842-9376-8_1.

Espressif Systems. "ESP-IDF Programming Guide." https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html, May 12, 2025.

Francesc Campà Aubets. “Design of an Arduino Shield for OTA Programming.” Universitat Politècnica de Catalunya, 2013. http://hdl.handle.net/2099.1/17305.

Gaber, Mohamed Medhat, Adel Aneiba, Shadi Basurra, Oliver Batty, Ahmed M. Elmisery, Yevgeniya Kovalchuk, and Muhammad Habib Ur Rehman. “Internet of Things and Data Mining: From Applications to Techniques and Systems.” *WIREs Data Mining and Knowledge Discovery* 9, no. 3 (May 9, 2019). https://doi.org/10.1002/widm.1292.

Gazis, Alexandros. “What Is IoT? The Internet of Things Explained.” *Academia Letters*, June 7, 2021. https://doi.org/10.20935/al1003.

Kim, Wan-Soo, Won-Suk Lee, and Yong-Joo Kim. “A Review of the Applications of the Internet of Things (IoT) for Agricultural Automation.” *Journal of Biosystems Engineering* 45, no. 4 (December 26, 2020): 385–400. https://doi.org/10.1007/s42853-020-00078-3.

Laghari, Asif Ali, Kaishan Wu, Rashid Ali Laghari, Mureed Ali, and Abdullah Ayub Khan. “A Review and State of Art of Internet of Things (IoT).” *Archives of Computational Methods in Engineering*. Springer Science and Business Media B.V., May 1, 2022. https://doi.org/10.1007/s11831-021-09622-6.

Laso, de Fez, and Emilio Aristides. *Instant AutoIt Scripting*. Packt Publishing, 2013.

Lin, Jie, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. “A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications.” *IEEE Internet of Things Journal* 4, no. 5 (October 2017): 1125–42. https://doi.org/10.1109/JIOT.2017.2683200.

Paramartha, I Gusti Ngurah Darma, I Nyoman Hary Kurniawan, Gde Brahupadhya Subiksa, and Ayu Satya Kartika. “Arsitektur Internet of Things (IoT) Berskala Industri Dengan Fitur Over The Air Update.” *TIERS Information Technology Journal* 2, no. 2 (December 29, 2021): 31–36. https://doi.org/10.38043/tiers.v2i2.3311.

Shen, Penghui, Yihong Qi, Wei Yu, Jun Fan, and Fuhai Li. “OTA Measurement for IoT Wireless Device Performance Evaluation: Challenges and Solutions.” *IEEE Internet of Things Journal* 6, no. 1 (February 2019): 1223–37. https://doi.org/10.1109/JIOT.2018.2868787.

Sudharsan, Bharath, John G. Breslin, Mehreen Tahir, Muhammad Intizar Ali, Omer Rana, Schahram Dustdar, and Rajiv Ranjan. “OTA-TinyML: Over the Air Deployment of TinyML Models and Execution on IoT Devices.” *IEEE Internet Computing* 26, no. 3 (May 1, 2022): 69–78. https://doi.org/10.1109/MIC.2021.3133552.

Tinetti, Fernando G., Alejo A Santi, and Mariano Méndez. “Continuous Integration and Continuous Delivery with Microcontrollers Applications.” In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1085–90. IEEE, 2023. https://doi.org/10.1109/CSCI62032.2023.00179.

Yusri, Iin Karmila, Kasim Kasim, and Andi Muhammad Akbar. “Penerapan Hypertext Transfer Protocol Web Server Untuk Over-The-Air Auto Update Firmware Pada Perangkat IoT.” *Elektron : Jurnal Ilmiah*, December 24, 2022, 67–71. https://doi.org/10.30630/eji.14.2.298.

Biografi Singkat Penulis



Hendra Rahmawan adalah **s**taf pengajar di Program Studi Ilmu Komputer, Sekolah Sains Data, Matematika dan Informatika, Institut Pertanian Bogor. Ia menyelesaikan studi sarjana di Departemen Ilmu Komputer, Institut Pertanian Bogor, dan master serta doktoral di Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung. Ia memiliki minat riset di bidang *high performance computing*, IoT, dan jaringan komputer.



Auriza Rahmad Akbar mendapatkan gelar sarjana dan magister bidang Ilmu Komputer dari Institut Pertanian Bogor (IPB). Minat risetnya di bidang sistem dan jaringan komputer, pemrosesan paralel, dan IoT. Saat ini bekerja sebagai dosen di IPB dan sedang melanjutkan studi doktoral di MBZUAI.



Fathan Mubina Sarjoko lulus dari Program Studi Sarjana Ilmu Komputer Institut Pertanian Bogor pada tahun 2024. Ia memiliki pengalaman di bidang tata kelola TI di GMF AeroAsia. Selain itu ia juga memiliki pengalaman sebagai pembicara publik yang terampil dalam memberdayakan audiens dengan wawasan tentang tren dan praktik baik di bidang TI.



Sri Wahjuni (Assoc. Prof, Anggota IEEE) meraih gelar sarjana dari Institut Teknologi Sepuluh November, Surabaya, Indonesia, gelar Magister Teknik, dan gelar Doktor dari Universitas Indonesia. Saat ini aktif mengajar di Program Studi ilmu komputer, IPB University. Minat penelitiannya adalah di bidang embedded system, pervasive systems, heterogeneous networks, Internet of Things, smart farming.