

PENGAPLIKASIAN TEKNIK *OVER THE AIR* (OTA) UNTUK MELAKUKAN PEMBARUAN KODE PROGRAM PADA SMART BREAKER ESP32 DENGAN PROTOKOL HTTP

APPLICATION OF OVER THE AIR (OTA) TECHNIQUES TO PERFORM PROGRAM CODE UPDATE ON SMART BREAKER ESP32 WITH HTTP PROTOCOL

Akbar Briliyanto¹, Mashoedah²

Program Studi Pendidikan Teknik Elektronika, Fakultas Teknik, Universitas Negeri Yogyakarta
akbarbriliyanto.2020@student.uny.ac.id¹⁾, mashoedah@uny.ac.id²⁾

ABSTRAK

Penelitian ini bertujuan untuk: (1) menghasilkan desain produk yang mengaplikasikan teknik OTA pada *smart breaker* ESP32; (2) mengetahui kinerja dari pengaplikasian OTA untuk melakukan pembaruan kode program pada *smart breaker*; (3) mengetahui hasil uji dari pengaplikasian teknik OTA untuk melakukan pembaruan kode program di dua perangkat secara bersamaan dan rentang waktu yang dibutuhkan selama proses pembaruan. Penelitian ini menerapkan metode penelitian R&D dengan model pengembangan waterfall yang terdiri dari analisis kebutuhan, desain, implementasi, pengujian, dan perawatan. Hasil dari penelitian ini: (1) pengaplikasian teknik OTA pada *smart breaker* didesain dengan memanfaatkan tiga produk yaitu website, aplikasi android dan *device smart breaker*; (2) ketiga produk yang telah dibuat dapat berfungsi dengan optimal; (3) Melalui teknik OTA pembaruan dapat dilakukan pada dua perangkat secara bersamaan dan rentang waktu pembaruan akan semakin cepat jika ukuran file kode program semakin kecil serta kecepatan internet yang dipakai tinggi, begitu pula sebaliknya.

Kata kunci: *Over The Air*, Pembaruan Kode Program, *Smart Breaker* ESP32

ABSTRACT

This research aims to: (1) produce a product design that applies OTA techniques to the ESP32 smart breaker; (2) determine the performance of the OTA application to update the program code on the smart breaker; (3) knowing the test results of applying the OTA technique to update program code on two devices simultaneously and the time span required during the update process. This research applies the R&D research method with the waterfall development model which consists of needs analysis, design, implementation, testing and maintenance. The results of this research: (1) the application of OTA techniques on smart breakers is designed by utilizing three products, namely websites, Android applications and smart breaker devices; (2) the three products that have been made can function optimally; (3) Through the OTA technique, updates can be carried out on two devices simultaneously and the update time span will be faster if the program code file size is smaller and the internet speed used is high, and vice versa.

Keywords: over the air, program code update, smart breaker ESP32

PENDAHULUAN

Perkembangan teknologi *internet of things* (IoT) yang begitu pesat menjadikan setiap pekerjaan dapat dimonitoring dan dikendalikan secara jarak jauh sehingga pekerjaan menjadi lebih mudah. Produk perangkat IoT yang akan dijual dipasaran tentunya perlu dilakukan pembaruan program secara berkala agar perangkat IoT dapat berfungsi dengan baik. Tujuan dari pembaruan program seperti menindak lanjuti laporan dari user, meningkatkan sistem keamanan, memperbaiki bug, menambah fitur, dan mengganti protokol komunikasi. Pembaruan program biasanya dilakukan dengan mengambil perangkat tersebut dari tempat asalnya kemudian menghubungkan perangkat tersebut kekomputer melalui kabel USB, kemudian melakukan *upload* program terbaru dan meletakkan perangkat tersebut ke tempat asalnya jika proses pembaruan sudah selesai, proses untuk melakukan pembaruan program tersebut tidak efektif apabila perangkat IoT yang akan diperbarui programnya dalam jumlah yang banyak serta berada dilokasi yang berbeda-beda, oleh karenanya perlu adanya sistem pembaruan program pada perangkat IoT yang dapat dilakukan secara jarak jauh dan dapat langsung menjangkau banyak perangkat.

Dari permasalahan diatas penulis tertarik membuat "Pengaplikasian Teknik Over The Air (OTA) Untuk Melakukan Pembaruan Kode Program Pada *Smart breaker* ESP32 Dengan Protokol HTTP". *Smart breaker* merupakan sebuah inovasi pintar yang menawarkan otomatisasi dan kendali pada sebuah aliran listrik melalui aplikasi yang sudah di instal pada *smartphone*. OTA adalah teknik untuk mengirimkan sebuah *firmware*/program melalui internet. Menurut Paramartha et al. (2021) teknik OTA juga diterapkan oleh perusahaan Tesla untuk melakukan pembaruan pada *firmware* di setiap kendaraan mereka dan para konsumen dapat mengatur pembaruan tersebut saat sedang parkir. Melalui OTA user atau produsen tidak perlu mengakses perangkat IoT secara fisik untuk melakukan pembaruan kode program, cukup melakukan pembaruan kode program melalui internet.

Protokol komunikasi yang dipakai pada penelitian ini yaitu memakai protokol komunikasi HTTP. Menurut Darmawan (2020) penggunaan protokol komunikasi HTTP mampu mengirimkan data dalam ukuran yang besar, serta tingkat keberhasilan pengaplikasian OTA lebih tinggi jika dibandingkan dengan protokol komunikasi MQTT. Selain itu penelitian yang dilakukan oleh Madani (2022) sistem update *firmware* perangkat IoT

menggunakan teknik OTA berbasis HTTP dari 10 percobaan didapat keberhasilan paket yang dapat diterima sebesar 100%. Sedangkan berdasarkan penelitian dari Hakim et al. (2020) tentang over the air update *firmware* pada perangkat IoT dengan protokol MQTT, didapat keberhasilan update program sebesar 80% dari 10 kali percobaan dengan QoS 2. Berdasarkan pada kedua penelitian diatas bahwa penggunaan HTTP cocok diterapkan untuk pengaplikasian teknik OTA untuk melakukan pembaruan kode program pada *smart breaker* ESP32. Menurut penelitian yang dilakukan oleh Wildan et al. (2021) jarak *device* terhadap router mempengaruhi kecepatan internet dan membuat proses pembaruan kode program OTA dapat semakin lama.

Penelitian ini memerlukan file program arduino yang sudah di export dalam bentuk file biner, untuk mengkonversi file program arduino menjadi biner dilakukan dengan menekan menu *export compiled binary* di arduino IDE, proses *export* yang telah berhasil ditandai dengan adanya file .bin di folder program tersebut. penelitian ini juga memerlukan website sebagai tempat untuk mengunggah file biner dari program tersebut, dan memerlukan aplikasi android sebagai kontrol versi kode program yang akan diterapkan untuk *smart breaker*nya.

Dari penjabaran diatas dalam penelitian ini berfokus untuk menyelesaikan permasalahan pembaruan program pada *device* IoT yang kurang efektif saat proses pembaruan program melalui kabel, dengan mengaplikasikan OTA yang memanfaatkan produk website, aplikasi android, dan *device* IoT berupa *smart breaker*, proses pembaruan program dapat dilakukan melalui jaringan internet dan dapat menjangkau banyak *device* dalam waktu yang bersamaan serta tidak terkendala lokasi *device* yang berbeda-beda.

METODE PENELITIAN

Jenis Penelitian

Penelitian ini menggunakan jenis penelitian *research and development* (R&D). Penelitian ini mengadopsi model pengembangan *waterfall*, meliputi:

1. Analisis Kebutuhan

Pengembangan diawali dengan mengumpulkan informasi yang dengan melakukan studi literatur mengenai permasalahan yang ada. Pengumpulan informasi bertujuan untuk memahami tujuan dari

pembuatan sistem, dalam proses studi literatur peneliti menganalisis kajian teori dari jurnal dan website yang ada di internet dan diperoleh mengenai gambaran cara kerja sistem, kebutuhan software, dan kebutuhan hardware.

2. Desain

Pada tahap desain berisi proses perancangan sistem sebelum di buat. Tahap ini berfokus pada desain arsitektur sistem, desain antarmuka, desain database, dan desain *device smart breaker* ESP32. Desain arsitektur dibuat menggunakan bahasa pemodelan *unified modeling language* (UML) yang berisikan *use case* dan *activity diagram* yang bertujuan untuk memberikan gambaran interaksi aktor dengan sistem dan menggambarkan keseluruhan cara kerja sistem akan dibuat.

3. Implementasi

Pada tahap ini berisi proses merealisasikan seluruh desain yang telah dibuat pada tahap sebelumnya. Pada tahap ini diawali dengan merealisasikan desain website, kemudian dilanjutkan dengan pembuatan aplikasi android, dan yang terakhir pembuatan *device smart breaker*.

4. Pengujian

Pengujian terdiri dari pengujian unjuk kerja produk, pengujian OTA di dua perangkat, Pengujian rentang waktu saat pembaruan.

5. Penerapan Program

Penerapan program merupakan tahapan yang dapat berupa mendistribusikan produk yang telah dibuat ke server asli agar dapat diakses oleh orang lain.

Waktu dan Tempat Penelitian

Tempat penelitian dilakukan di Klaten, Jawa Tengah. Waktu penelitian dilakukan selama 3 bulan yang dimulai dari tanggal 1 Februari-30 April 2024.

Instrumen Pengumpulan Data

Teknik pengumpulan data yang dilakukan pada penelitian ini melibatkan beberapa pengamatan/observasi secara langsung. Instrumen pengumpulan data merupakan perangkat yang digunakan untuk memperoleh data dengan maksud agar mengetahui hasil dari pengaplikasian OTA untuk melakukan pembaruan kode program di dua perangkat secara bersamaan, serta mengetahui rentang waktu yang diperlukan ESP32 untuk melakukan pembaruan kode program sampai menjalankan kode program terbarunya. Software arduino IDE digunakan untuk melihat lamanya waktu

yang diperlukan *smart breaker* untuk melakukan pembaruan kode program hingga menjalankan kode program terbarunya. Untuk kisi-kisi pengujian yang akan dilakukan dapat dilihat pada tabel dibawah ini:

Tabel 1. Kisi-Kisi Pengujian Pembaruan di Dua Perangkat

Aspek	Indikator	Hasil
Keberhasilan update program di <i>smart breaker</i> 1	<i>Smart breaker</i> mengunduh file program dan menjalankan versi program terbaru.	Keterangan berhasil atau tidak
Keberhasilan update program di <i>smart breaker</i> 2	<i>Smart breaker</i> mengunduh file program dan menjalankan versi program terbaru	Keterangan berhasil atau tidak

Tabel 2. Kisi-Kisi Pengujian Pembaruan di Dua Perangkat

Aspek	Ukuran File	Indikator	Hasil
Rentang waktu proses pembaruan dengan kecepatan internet 10 Mbps-25 Mbps	885Kb	Rentang waktu pembaruan dimulai hingga proses pembaruan selesai	Rata-rata rentang waktu pembaruan
	911Kb	Rentang waktu pembaruan dimulai hingga proses selesai	Rata-rata rentang waktu pembaruan
	1025Kb	Rentang waktu pembaruan dimulai hingga proses selesai	Rata-rata rentang waktu pembaruan
Rentang waktu proses pembaruan dengan	885Kb	Rentang waktu pembaruan dimulai hingga	Rata-rata rentang waktu pembaruan

kecepatan internet 0 Mbps-5 Mbps	911Kb	proses selesai Rentang waktu saat pembaruan dimulai hingga proses selesai	Rata-rata rentang waktu pembaruan
	1025Kb	Rentang waktu saat pembaruan dimulai hingga proses selesai	Rata-rata rentang waktu pembaruan

Teknik Analisis Data

Dalam penelitian ini, teknik analisis statistik deskriptif digunakan untuk meringkas dan menggambarkan data yang diperoleh dari hasil pengumpulan data pengujian pembaruan di dua perangkat dan pengujian rentang waktu pembaruan. Menurut Sugiyono (2018) metode ini melibatkan perhitungan statistik dasar seperti nilai rata-rata, deviasi standar, dan distribusi data. Dengan menggunakan teknik ini peneliti dapat memahami karakteristik dari hasil pengujian pembaruan di dua perangkat dan data dari hasil pengujian rentang waktu pembaruan kode program.

HASIL DAN PEMBAHASAN

Hasil penelitian

Pada model pengembangan *waterfall* diawali dengan melakukan analisis kebutuhan yang terdiri dari kebutuhan software yang terdiri dari:

Tabel 3. Kebutuhan Software

No	Nama	Fungsi
1	Arduino IDE	Membuat program
2	Visual Studio Code	Membuat program
3	Fritzing	Membuat skema
4	Android Studio	Membuat program
5	Eagle	Membuat layout PCB
6	Fusion 360	Membuat desain 3D

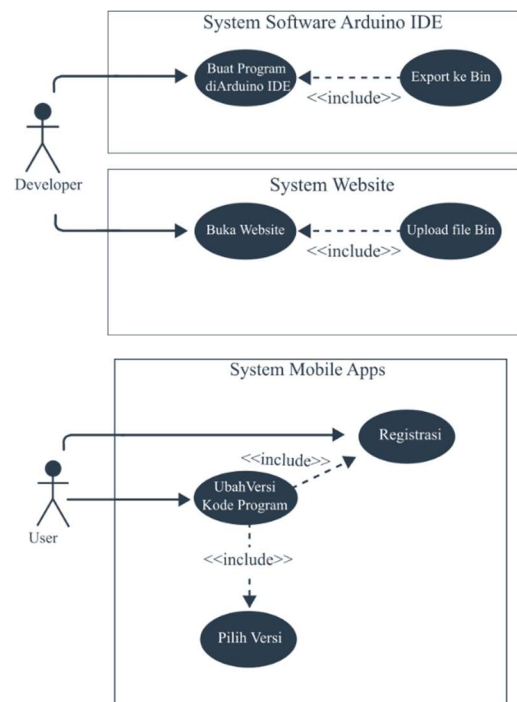
7	Pencil	Membuat desain UI
8	Web server Uvicorn	Menjalankan server

Sedangkan untuk kebutuhan hardware terdiri dari:

Tabel 4. Kebutuhan Hardware

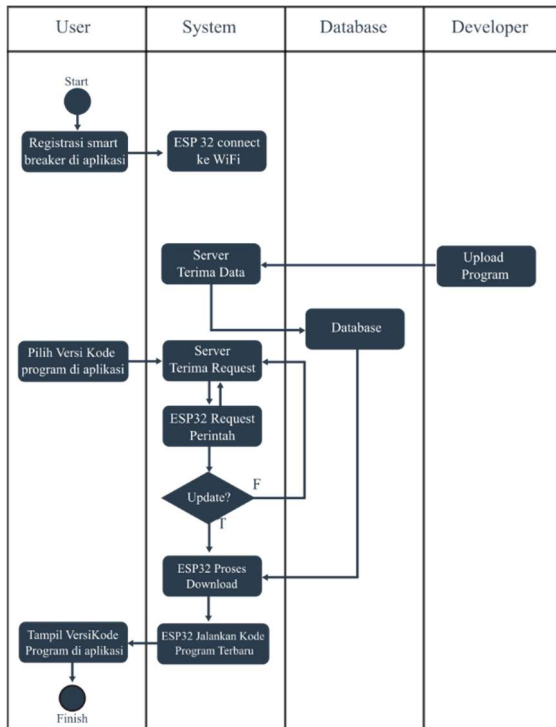
No	Nama	Jumlah
1	Microcontroller ESP32	1
2	Button	1
3	Modul relay 4 channel	1
4	PCB	1
5	Adaptor 9V	1
6	IC 7805	1
7	Resistor 10k	1
8	6 Pin JST XH Pitch	2
9	Pin Header Female	1

Setelah melakukan identifikasi masalah maka selanjutnya dilakukan proses desain yang terdiri dari desain arsitektur, desain antarmuka, desain database dan desain hardware. Untuk desain arsitektur terdiri dari *use case diagram* yang berisi gambaran bagaimana aktor akan menggunakan suatu sistem.



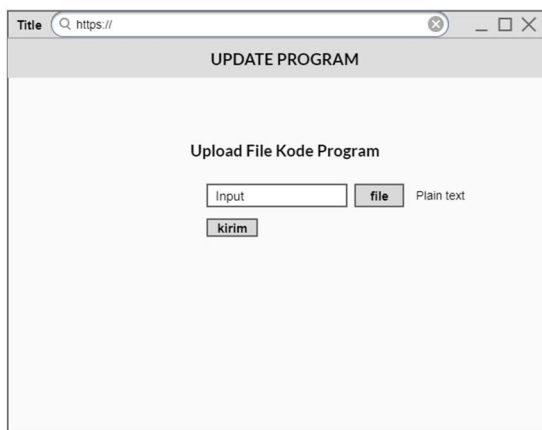
Gambar 1. Use Case Diagram

Selain use case diagram terdapat juga activity diagram yang menggambarkan aliran kerja dari sebuah sistem.



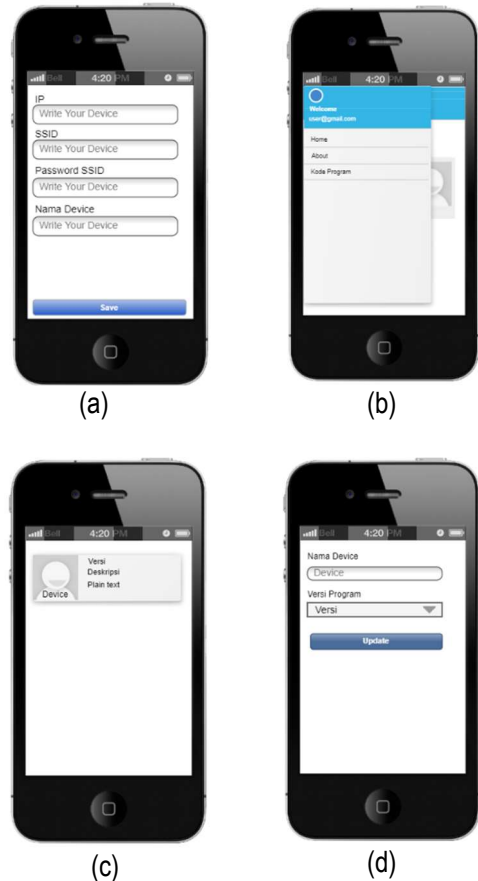
Gambar 2. Activity Diagram

Setelah yaitu pembuatan desain antarmuka website dan aplikasi android, bentuk desain website dapat dilihat pada gambar dibawah ini.



Gambar 3. Desain Website

Selanjutnya untuk bentuk desain aplikasi android dapat dilihat pada gambar berikut.



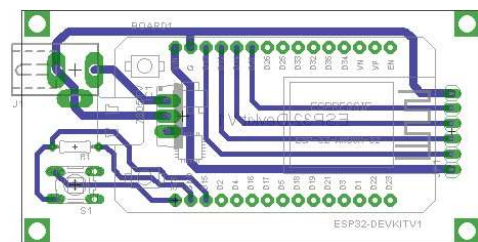
Gambar 4. Desain Aplikasi Android: (a) Desain Halaman Registrasi, (b) Desain Halaman Home, (c) Desain Halaman List Device, (d) Desain Halaman Perintah Update

Selanjutnya untuk desain dari database yang dipakai dapat dilihat pada tabel berikut.

Tabel 5. Struktur Database

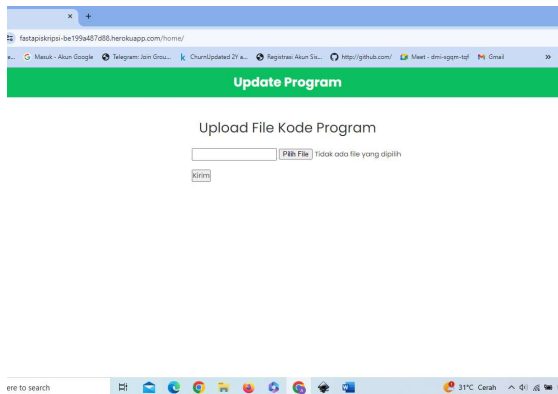
No	Nama	Jenis	Key
1	Id	Integer	Primary Key
2	Versi	Text	
3	Nama	Text	
4	Data	Blob	

Sedangkan untuk bentuk layout rangkaian dari device smart breaker ESP32 dapat dilihat pada gambar berikut.



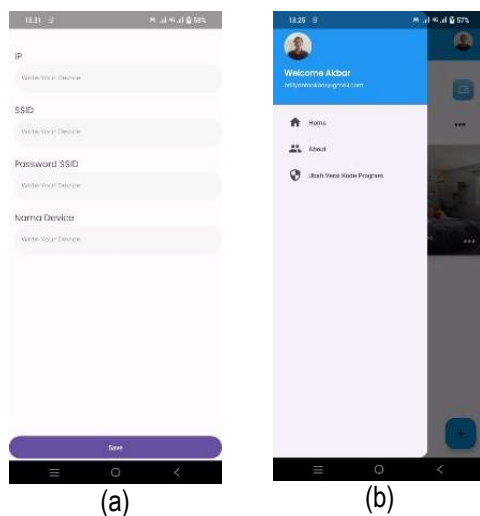
Gambar 5. Layout device smart breaker

Setelah tahap desain telah selesai selanjutnya adalah proses implementasi dari tahap desain yang telah dilakukan, proses implementasi tahap pertama adalah implementasi desain dari website yang telah dibuat, untuk bentuk hasil implementasi website dapat dilihat pada gambar dibawah ini.



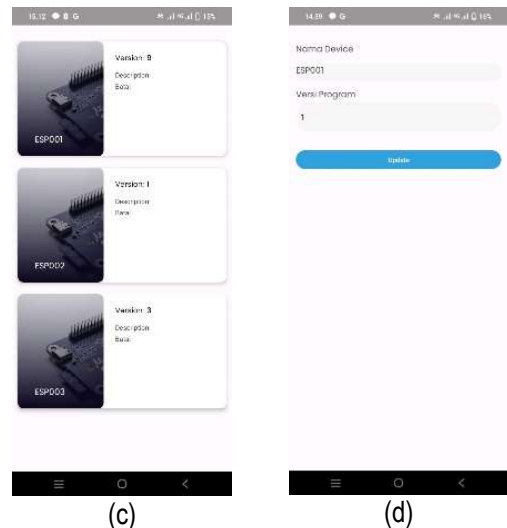
Gambar 6. Tampilan Website

Setelah pembuatan website, tahap selanjutnya yaitu implementasi desain aplikasi android, aplikasi android dibuat menggunakan bahasa pemrograman kotlin. Desain tampilan pada aplikasi android berdasarkan pada desain aplikasi yang sebelumnya telah dibuat. Langkah pembuatan aplikasi android diawali dengan membuat halaman registrasi, dilanjutkan dengan membuat halaman home, kemudian membuat halaman *list device*, dan yang terakhir membuat halaman perintah *update*. Hasil dari pembuatan aplikasi android dapat dilihat pada gambar dibawah ini:



(a)

(b)



(c)

(d)

Gambar 7. Tampilan Aplikasi Android: (a) Halaman Registrasi, (b) Halaman Home, (c) Halaman *List Device*, (d) Halaman Perintah *Update*

Selanjutnya dilakukan pembuatan database sesuai dengan struktur desain database yang telah dibuat, untuk bentuk struktur database yang telah dibuat dapat dilihat pada gambar berikut.

Name	Type	NN	PK	AI	U	Default
id	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
versi	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
name	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
data	BLOB	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Gambar 8. Bentuk *Database*

Selanjutnya adalah tahap implementasi pembuatan device smart breaker ESP32, untuk bentuk dari device smart breaker ESP32 dapat dilihat pada gambar dibawah ini.



Gambar 9. *Device Smart Breaker*

Setelah tahap implementasi selesai selanjutnya yaitu tahap pengujian, pada tahap pengujian terdiri dari:

a. Pengujian Unjuk Kerja Produk

Pengujian ini meliputi; pengujian website, pengujian aplikasi android, pengujian pembaruan kode program di *smart breaker* ESP32.

Tabel 6. Pengujian Website

Pengujian	Keterangan	
	Berhasil	Tidak
Uji ke 1 upload program	✓	
Uji ke 2 upload program	✓	
Uji ke 3 upload program	✓	
Uji ke 4 upload program	✓	

Berdasarkan pada 4 kali pengujian yang telah dilakukan diperoleh hasil bahwa website dapat mengunggah file kode program ke dalam database.

Tabel 7. Pengujian Aplikasi Android

Pengujian	Keterangan	
	Berhasil	Tidak
Uji ke 1 kirim perintah	✓	
Uji ke 2 kirim perintah	✓	
Uji ke 3 kirim perintah	✓	
Uji ke 4 kirim perintah	✓	

Berdasarkan pada hasil pengujian diatas didapat hasil bahwa aplikasi android dapat melakukan perintah ke *smart breaker* untuk melakukan pembaruan kode program.

Tabel 8. Pengujian Pembaruan Kode Program di Smart Breaker ESP32

Pengujian	Keterangan	
	Berhasil	Tidak
Uji ke 1 update program	✓	
Uji ke 2 update program	✓	
Uji ke 3 update program	✓	
Uji ke 4 update program	✓	

Berdasarkan pada hasil percobaan di tabel 8, dari 4 kali percobaan dapat disimpulkan bahwa *smart breaker* mampu melakukan pembaruan kode program melalui jaringan internet.

b. Pengujian pembaruan kode program pada *smart breaker* ESP32 dengan protokol HTTP di dua perangkat *smart breaker* secara bersamaan.

Tabel 9. Tabel Pengujian OTA di Dua Perangkat

Pengujian	Device	
	1	2
Uji ke 1 update program	✓	✓
Uji ke 2 update program	✓	✓
Uji ke 3 update program	✓	✓
Uji ke 4 update program	✓	✓

Berdasarkan pada hasil pengujian yang telah dilakukan diperoleh hasil bahwa dengan pembaruan kode program melalui teknik OTA dapat melakukan pembaruan kode program pada dua *smart breaker* dalam waktu yang bersamaan.

c. Pengujian rentang waktu saat pembaruan kode program *smart breaker* ESP32 hingga menjalankan kode program terbaru berdasarkan ukuran program dan kecepatan internet yang dipakai.

Tabel 10. Pengujian Rentang Waktu Saat Pembaruan Kecepatan Internet 10 Mbps-25 Mbps

No	885Kb	911Kb	1025Kb
1.	1 menit 20 detik	1 menit 23 detik	1 menit 38 detik
2.	1 menit 20 detik	1 menit 23 detik	1 menit 34 detik
3.	1 menit 21 detik	1 menit 22 detik	1 menit 35 detik
4.	1 menit 18 detik	1 menit 29 detik	1 menit 31 detik
5.	1 menit 21 detik	1 menit 22 detik	1 menit 35 detik
6.	1 menit 22 detik	1 menit 23 detik	1 menit 31 detik
7.	1 menit 22 detik	1 menit 23 detik	1 menit 38 detik
8.	1 menit 20 detik	1 menit 23 detik	1 menit 34 detik
9.	1 menit 21 detik	1 menit 27 detik	1 menit 35 detik
10.	1 menit 21 detik	1 menit 22 detik	1 menit 31 detik
11.	1 menit 20 detik	1 menit 23 detik	1 menit 35 detik
12.	1 menit 20 detik	1 menit 21 detik	1 menit 31 detik
13.	1 menit 21 detik	1 menit 23 detik	1 menit 31 detik
14.	1 menit 19 detik	1 menit 23 detik	1 menit 32 detik

15.	1 menit 24 detik	1 menit 21 detik	1 menit 32 detik
16.	1 menit 20 detik	1 menit 25 detik	1 menit 38 detik
17.	1 menit 21 detik	1 menit 23 detik	1 menit 41 detik
18.	1 menit 21 detik	1 menit 34 detik	1 menit 38 detik
19.	1 menit 20 detik	1 menit 24 detik	1 menit 32 detik
20.	1 menit 19 detik	1 menit 26 detik	1 menit 32 detik
Rata-Rata	1 menit 20,55 detik	1 menit 24 detik	1 menit 33,75 detik

Tabel 11. Pengujian Rentang Waktu Saat Pembaruan Kecepatan Internet 0 Mbps-5 Mbps

No	885Kb	911Kb	1025Kb
1.	2 menit 35 detik	1 menit 32 detik	1 menit 52 detik
2.	1 menit 52 detik	1 menit 37 detik	1 menit 55 detik
3.	1 menit 48 detik	1 menit 52 detik	1 menit 51 detik
4.	2 menit 3 detik	2 menit 31 detik	2 menit 4 detik
5.	1 menit 14 detik	1 menit 40 detik	1 menit 59 detik
6.	1 menit 45 detik	1 menit 39 detik	1 menit 43 detik
7.	2 menit 0 detik	1 menit 26 detik	1 menit 58 detik
8.	2 menit 11 detik	1 menit 26 detik	1 menit 35 detik
9.	1 menit 32 detik	1 menit 35 detik	2 menit 27 detik
10.	2 menit 5 detik	1 menit 39 detik	2 menit 6 detik
11.	1 menit 41 detik	1 menit 32 detik	2 menit 18 detik
12.	1 menit 42 detik	3 menit 50 detik	1 menit 32 detik
13.	1 menit 48 detik	1 menit 53 detik	3 menit 1 detik
14.	1 menit 39 detik	1 menit 25 detik	1 menit 34 detik
15.	1 menit 37 detik	2 menit 21 detik	2 menit 55 detik
16.	1 menit 47 detik	1 menit 26 detik	3 menit 6 detik
17.	2 menit 6 detik	1 menit 51 detik	2 menit 9 detik

18.	1 menit 30 detik	1 menit 33 detik	1 menit 53 detik
19.	1 menit 35 detik	1 menit 37 detik	2 menit 6 detik
20.	1 menit 32 detik	2 menit 17 detik	1 menit 34 detik

Setelah tahap pengujian selesai maka selanjutnya tahap penerapan program yang merupakan tahapan mendistribusikan produk yang telah dibuat ke server asli agar dapat diakses oleh orang lain. Pada tahap ini program yang telah diterapkan dapat diubah dan diadaptasi dengan situasi dengan lingkungan baru dan dapat berubah karena user yang menginginkan kebutuhan perkembangan fungsional dari produk yang telah dibuat. Untuk website di *deploy* pada server heroku, sedangkan aplikasi android di *bulid* kedalam bentuk apk. Hasil *deploy* diserver telah dapat dilihat orang lain dengan menulis <https://smartbreakerota-9b29e4c4112f.herokuapp.com/home/> ke dalam web browser.

Pembahasan

Berdasarkan pada hasil penelitian yang telah dilakukan menghasilkan tiga produk yaitu website, aplikasi android, dan *device smart breaker*. Ketiga produk tersebut telah melalui rangkaian pengujian yang ditunjukkan pada tabel 6, tabel 7, dan tabel 8 yang menunjukkan bahwa ketiga produk tersebut memiliki kinerja yang optimal. Adapun penjelasan mengenai desain dan kinerja dari ketiga produk dapat dilihat sebagai berikut:

a. Website

Website dibuat sebagai sarana untuk *developer* agar bisa mengupload kode program yang telah dibuat ke dalam database. Tampilan website dibuat menggunakan HTML dan CSS untuk membuat tampilannya lebih bagus, sedangkan agar website dapat lebih interaktif maka ditambahkan kode program python untuk membuat website bisa interaktif. Website yang dibuat memiliki empat komponen yaitu text field, form upload, label text, dan button. Website di uji untuk dapat mengunggah file kode program yang telah dibuat oleh *developer* ke dalam database. Berdasarkan pada tabel pengujian nomor 6 dengan 4 kali pengujian yang dilakukan website dapat mengunggah file kode program ke database dengan optimal.

b. Aplikasi Android

Aplikasi android di desain hanya digunakan oleh *user* untuk memberikan perintah update program pada *device smart breaker*. Aplikasi android

terdiri dari empat halaman yaitu halaman registrasi, halaman home, halaman *list device*, dan halaman perintah *update*. Aplikasi android diuji dengan menekan icon button pada aplikasi dan dilihat apakah aplikasi dapat mengirimkan perintah *update* ke *smart breaker* atau tidak. Berdasarkan pada tabel 7 dengan 4 kali pengujian dihasilkan kesimpulan bahwa aplikasi android dapat berfungsi dengan optimal untuk mengirimkan perintah *update* ke *smart breaker*.

c. *Device Smart Breaker ESP32*

Device smart breaker ESP32 dibuat mengikuti desain yang telah dibuat pada tahap perancangan sebelumnya, pembuatan hardware terdiri dari dua produk yaitu *board PCB* dan *box casing*. Pada *board PCB* dibuat menggunakan bahan PCB jenis Fr4 dengan hasil akhir ukuran dimensi *board PCB* adalah 76 mm x 39 mm. Untuk desain *box casing* dibuat menggunakan bahan dasar akrilik dengan ketebalan 3 mm, dimensi dari *box casing* berukuran 110 mm x 90 mm x 40 mm. *Device smart breaker* di uji dengan melakukan mengupdate versi lama program ke versi program terbaru. Berdasarkan pada tabel 8 dengan 4 kali pengujian menghasilkan kesimpulan bahwa *smart breaker* mampu *update* ke program terbaru melalui internet dengan optimal.

Pada hasil pengujian pembaruan di dua perangkat diperoleh hasil bahwa pembaruan kode program dengan teknik OTA dapat melakukan pembaruan kode program pada dua *smart breaker* dalam waktu yang bersamaan. Melalui pengujian sebanyak 4 kali dapat disimpulkan bahwa pengaplikasian OTA untuk melakukan pembaruan kode program pada dua *smart breaker* secara bersamaan tetap berfungsi dengan baik.

Untuk rentang waktu yang dibutuhkan oleh *smart breaker* selama proses pembaruan dapat dilihat pada tabel 10 dan 11. Pada tabel 10 dengan kecepatan internet berkisar 10Mbps - 25Mbps diperoleh rata-rata waktu 1 menit 20,55 detik untuk ukuran file 885Kb, sedangkan ukuran file 911Kb membutuhkan waktu 1 menit 24 detik, dan untuk ukuran file 1025Kb membutuhkan waktu 1 menit 33,75 detik.

Sedangkan pada hasil penelitian di tabel 11 dengan kecepatan internet berkisar 0 Mbps – 5 Mbps untuk ukuran file 885Kb rata-rata membutuhkan waktu 1 menit 48,5 detik, untuk ukuran file 911Kb membutuhkan waktu 1 menit 50,1 detik, dan untuk ukuran file 1025Kb membutuhkan waktu 1 menit 58,9 detik.

Berdasarkan tabel 10 dan tabel 11 dapat disimpulkan bahwa ukuran file kode program dan kecepatan internet mempengaruhi lamanya proses

pembaruan kode program pada *smart breaker*, semakin besar kode program dan semakin lambat kecepatan internet akan membuat waktu yang dibutuhkan selama proses pembaruan kode program akan semakin lama, dan sebaliknya apabila semakin kecil ukuran file kode program dengan didukung kecepatan internet yang tinggi akan membuat proses pembaruan kode program pada *smart breaker* semakin cepat.

KESIMPULAN DAN SARAN

Kesimpulan

Dari penelitian yang telah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Hasil desain untuk pengaplikasian teknik *over the air* (OTA) dibuat dengan memanfaatkan tiga produk yaitu website, aplikasi android, dan *device smart breaker* yang keseluruhan produk tersebut memiliki tugas yang saling mendukung untuk pengoptimalan pengaplikasian OTA pada *smart breaker*.
2. Produk yang telah dibuat untuk mendukung pengaplikasian teknik *over the air* (OTA) untuk melakukan pembaruan kode program pada *smart breaker* ESP32 dengan protokol HTTP telah dilakukan pengujian unjuk kerja produk dan dihasilkan kesimpulan bahwa website dan aplikasi android dapat menunjukkan kinerja yang optimal, serta *device smart breaker* dapat melakukan pembaruan kode program dengan optimal.
3. Pengaplikasian OTA untuk melakukan pembaruan kode program pada dua *smart breaker* secara bersamaan dapat berfungsi dengan optimal, untuk pengujian rentang waktu pembaruan diperoleh hasil bahwa semakin kecil ukuran file kode program dan semakin tinggi kecepatan internet akan membuat waktu yang dibutuhkan selama proses pembaruan kode program akan semakin cepat, dan hal tersebut akan berbanding terbalik jika berada di kondisi sebaliknya.

Saran

Berdasarkan hasil penelitian yang sudah dilaksanakan. adapun beberapa saran dari penulis terhadap pengaplikasian teknik *Over The Air* (OTA) untuk melakukan pembaruan kode program pada *smart breaker* ESP32 dengan protokol HTTP, diantaranya sebagai berikut:

1. Pengembangan mengenai pengaplikasian teknik *Over The Air* (OTA) pada *smart breaker*

- dapat lebih dikembangkan dengan menerapkan protokol komunikasi HTTPS agar lebih aman.
2. Produk aplikasi yang sudah dibuat dapat dikembangkan kembali sehingga dapat dikembangkan kembali sehingga dapat dijalankan pada smartphone yang memakai sistem operasi IOS.

DAFTAR PUSTAKA

- Darmawan, R. (2020). *Studi komparasi performa protokol HTTP dengan MQTT pada sistem smarhome neuronthings*. [Skripsi, Tidak diterbitkan]. Universitas Pertamina
- Hakim, L., Kusuma, W. A., Faiqurahman, M., & Supriyanto. (2020). Over the air update firmware pada perangkat IoT dengan protokol MQTT. *Jurnal Sistem Dan Informatika (JSI)*, 14(2), 99–105. <https://doi.org/10.30864/jsi.v14i2.244>
- Madani, K., Hidayati, R., & Ristian, U. (2022). Sistem update firmware perangkat IoT menggunakan teknik OTA. *JURIKOM (Jurnal Riset Komputer)*, 9(4), 1160-1166. <http://dx.doi.org/10.30865/jurikom.v9i4.4685>
- 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>
- Sugiyono. (2018). *Metode penelitian pendidikan: pendekatan kuantitatif, kualitatif, dan R&D*. Alfabeta
- Wildan, A., Hannats, M., Ichsan, H., & Syauqy, D. (2021). Implementasi over the air update menggunakan protokol SSDP untuk pencarian perangkat. 5(9), 3705–3711. <http://j-ptiik.ub.ac.id>