

System Design For Iot File Firmware Update Using Over The Air Update

1st Anggawa Kharisma Hananta
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

anggawakharisma@telkomuniversity.ac.i

d

2nd Muhammad Ary Murti
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

arymurti@telkomuniversity.ac.id

3rd Novi Prihatiningrum
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia

nprihatiningrum@telkomuniversity.ac.id

Abstract— Perangkat IoT yang telah diimplementasi di berbagai tempat membutuhkan pembaruan dalam perangkat lunak dikarenakan akan adanya perubahan parameter baik yang terkait komunikasi dengan sistem maupun pengguna, memperbaiki kesalahan sistem maupun keamanan sistemnya. Perubahan akan dilakukan dengan cara update firmware perangkat menggunakan usb to micro usb menghubungkan ke perangkat IoT dan *personal computer* maupun laptop melalui komunikasi secara serial. Masalah akan timbul jika perangkat IoT tidak dapat terjangkau dengan mudah. Solusi yang akan dilakukan adalah memanfaatkan metode Over The Air (OTA) update firmware pada perangkat IoT dan menggunakan aplikasi mobile berbasis website sebagai interface pengguna untuk memantau hasil pembaruan maupun hasil pengolahan datanya. Penelitian ini menggunakan NodeMCU ESP8266, sensor ACS712, dan FTDI. Penulis akan memasukkan data *ssid* dan *password* pada *mobile application*. Hasil dari pembacaan data sensor akan ditampilkan pada *mobile application* sebagai output terakhir.

Keywords— OTA, Firmware, IoT, Access point, NodeMCU ESP8266

I. PENDAHULUAN

Internet of Things (IoT) adalah sebuah konsep yang memiliki tujuan untuk memperluas ruang lingkup manfaat dari konektivitas internet yang terintegrasi. Singkatnya IoT dapat menggabungkan antara benda fisik dan virtual melalui eksploitasi

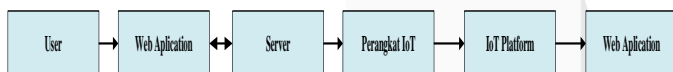
data dan kemampuan komunikasi dalam jaringan. Konsep Internet of Things (IoT) telah dinyatakan pada akhir tahun 90-an. Pada akhir tahun 2013 terdapat 9,1 miliar unit IoT yang terpasang, dengan konektivitas IP dan berkomunikasi tanpa interaksi manusia. Diharapkan jumlah unit yang ada dapat mencapai 28,1 miliar unit pada tahun 2020 [1]. Perubahan pada perangkat IoT dilakukan dengan cara mengubah firmware-nya, untuk mengubah firmware-nya maka perlu menghubungkan hardware-nya dengan komputer lalu mengembalikan hardware tersebut ke posisi semula. Hal ini memiliki tingkat resiko yang cukup tinggi dikarenakan pada saat penghubungan dengan komputer bisa dilakukan perubahan-perubahan yang tidak diinginkan oleh pihak yang tidak bertanggung jawab. Meminimalisir resiko tindak kejahatan yang dilakukan oleh pihak-pihak tidak bertanggung jawab serta pengurangan biaya produksi adalah salah satu alasan dibutuhkannya suatu sistem yang dapat berkomunikasi antarmuka jaringan yang dapat diterapkan pembaruan pada sistem. Sistem ini biasanya disebut sebagai Over the Air (OTA). Dengan menggunakan saluran komunikasi yang ada maka pembaruan antarmuka khusus dapat dihilangkan yang menyebabkan pengurangan biaya produksi dan biaya pemeliharaan yang lebih kecil karena pembaruan dapat dipicu dari jarak jauh [2]. Mekanisme update OTA berbeda secara umum dengan pemrograman OTA. Pemrograman OTA memerlukan bootloader yang secara penuh agar dapat menjalankan fungsi OTA. Pada update OTA hanya membutuhkan bootloader yang memiliki fungsi untuk melakukan self-flashing atau programming dan download OTA-nya dilakukan di program utamanya. Hal ini mengharuskan mikrokontroler harus memiliki program utama

terlebih dulu yang mana telah diinput secara manual terlebih dahulu [4]. Pengertian pembaruan Over the Air (OTA) sendiri adalah proses mengunggah firmware ke modul ESP menggunakan koneksi Wi-Fi. Sistem tersebut menjadi sangat berguna jika terdapat kondisi akses yang terbatas terhadap modul. OTA dapat dilakukan dengan menggunakan Arduino IDE, Web Browser, HTTP Server [3]. Berdasarkan latar belakang yang ada maka akan dilakukan pengimplementasian over the air update pada perangkat IoT. Over the air update akan digunakan untuk melakukan update firmware pada perangkat IoT dari jaringan melalui media mobile application berbasis website. Mobile application tersebut merupakan sebagai media interface pengguna untuk melakukan update file firmware dan akan melakukan monitoring hasil pengunggahan yang telah dibaca melalui IoT platform.

II. PERANCANGAN SISTEM

Dalam penelitian ini hanya membuat perancangan sistem *Over The Air (OTA) update* menggunakan NodeMCU ESP8266 sebagai mikrokontroler yang kemudian parameter-parameter yang akan dilakukan pembaruan adalah *ssid* dan *password* yang output terakhirnya akan ditampilkan pada *mobile application*.

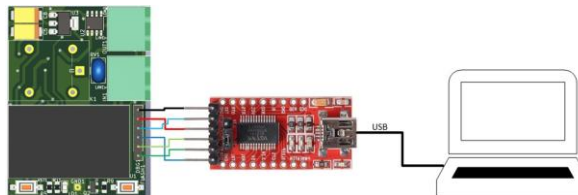
A. Blok Diagram



Gambar 1 Blok Diagram

Gambar diatas merupakan blok diagram dari sistem yang ingin dibuat, *user* melakukan *input* pada *mobile application*, data yang telah diinput kemudian akan disimpan pada server. Perangkat kemudian akan mengambil data dari server dan setelah data terbaru maka akan dilakukan data sensing yang kemudian data diolah pada IoT Platform. Hasil dari pengolahan akan ditampilkan pada *mobile application*.

B. Desain Perangkat Keras



Gambar 2 Desain Perangkat Keras

Perangkat IoT yang menggunakan mikrokontroler NodeMCU ESP8266 digunakan untuk mendukung sistem kerja alat ini. Sensor arus ACS712 sebagai input untuk membaca kuat arus listrik. NodeMCU ESP8266 dipilih sebagai mikrokontroler yang sekaligus digunakan untuk melakukan update OTA. FTDI digunakan untuk membantu komunikasi serialnya ketika akan melakukan *hard-code*.

1. NodeMCU ESP8266



Gambar 3 NodeMCU ESP8266

Mikrokontroler ini terhubung dengan ESP8266 dan memiliki 13 pin digital yang memiliki power input dc 5V. Data dari sensor akan diproses oleh mikrokontroler dan dikirimkan ke dalam IoT platform yang selanjutnya akan diolah lalu ditampilkan pada *mobile application* dan juga modul ini akan dilakukan pembaruan *firmware* melalui OTA dalam bentuk *ssid* dan *password*. Data yang akan ditampilkan pada *mobile application* adalah berupa hasil sensing daya yang akan dibaca oleh sensor arus ACS712.

2. Sensor ACS712

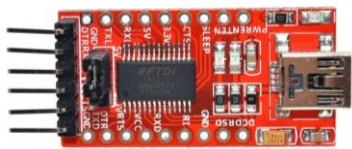


Gambar 4 Sensor Arus ACS712

Sensor arus ACS712 akan melakukan data sensing kuat arus listrik. Hasil dari data sensing kemudian akan dilakukan pengolahan di IoT platform yang kemudian akan ditampilkan pada *mobile application* kepada *user*.

Prinsip kerja sensor ini adalah melakukan pembacaan pada arus listrik yang melewati kabel tembaga didalamnya yang akan menghasilkan sebuah medan magnet dan akan ditangkap oleh integrated Hall IC kemudian diubah menjadi sebuah tegangan proporsional.

3. FTDI FT232RL

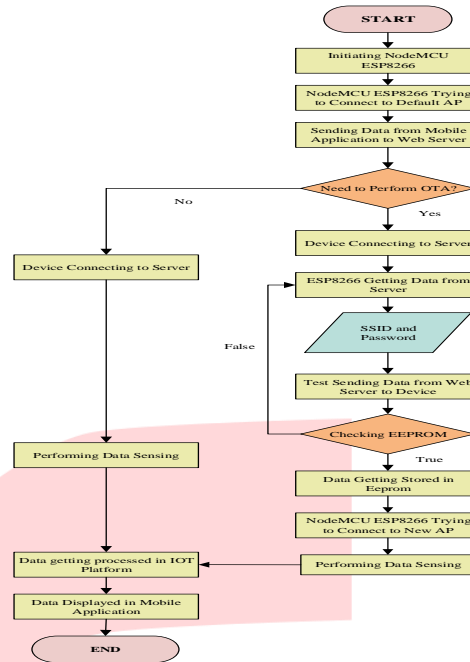


Gambar 5 FTDI FT232RL

FTDI merupakan sebuah modul IC yang berfungsi sebagai konverter sinyal USB-to-UART. Modul FTDI akan melakukan komunikasi serial yang berlangsung antara NodeMCU ESP8266 dengan komputer jinjing.

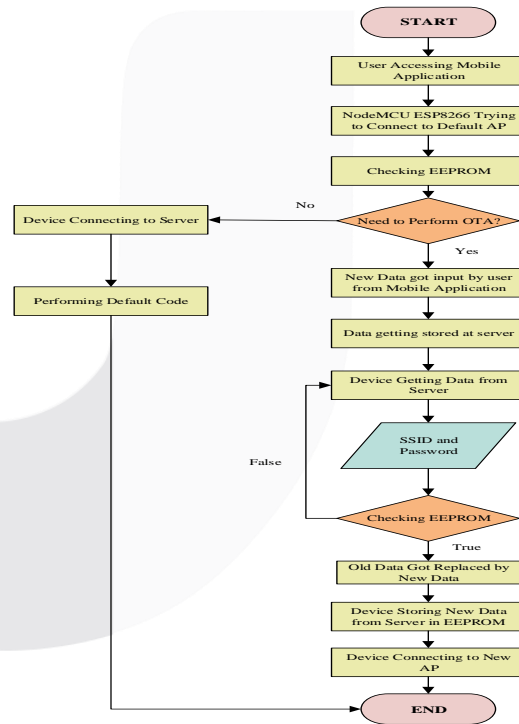
C. Desain Perangkat Lunak

Rancangan sistem perangkat lunak perangkat IoT akan dirancang menggunakan Arduino IDE untuk dapat memprogram mikrokontroler NodeMCU ESP8266.



Gambar 6 Desain Perangkat Lunak

1. Mekanisme OTA



Gambar 7 Mekanisme OTA

User mengakses *mobile application* dahulu. Perangkat akan terhubung dengan *default AP*. Perangkat akan melakukan *checking* pada eeprom, jika tidak dibutuhkan pembaruan OTA maka akan melakukan proses pembacaan data oleh sensor arus

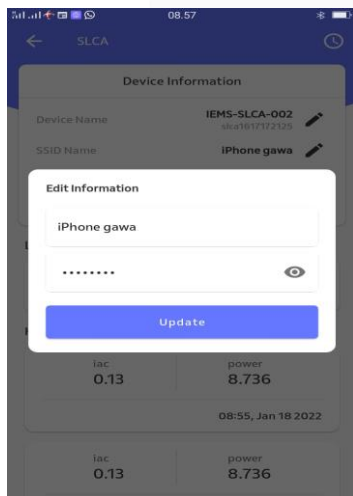
dan data tersebut akan diolah pada IOT Platform yang akan ditampilkan pada *mobile application*. Pada saat dibutuhkan pembaruan OTA maka perlu dilakukan input data baru pada *interface mobile application* oleh *user*. Data tersebut akan disimpan pada *database* server. Perangkat akan mengambil data tersebut dari *database* dan akan disimpan pada *eeprom*. Dilakukan konfirmasi apakah data tersebut merupakan data *ssid* dan *password*, jika tidak maka perangkat akan mencoba mengambil data tersebut kembali. Data baru yang telah diambil kemudian akan mengganti data lama. Data baru tersebut akan disimpan di *eeprom* dan akan dijadikan sebagai *default AP* terbaru. Perangkat kemudian akan mencoba menghubungkan pada AP tersebut.

III. PEMBAHASAN

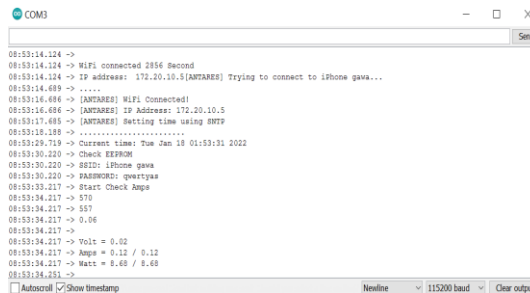
Percobaan akan dilakukan dengan melakukan pengujian untuk mendapatkan tingkat keberhasilan dari OTA yang telah dilakukan serta performa perangkat dalam melakukan OTA.

A. Tingkat keberhasilan perangkat dalam melakukan OTA

Pengujian ini menggunakan lima kali percobaan untuk mendapatkan tingkat keberhasilan dari OTA yang telah dilakukan.



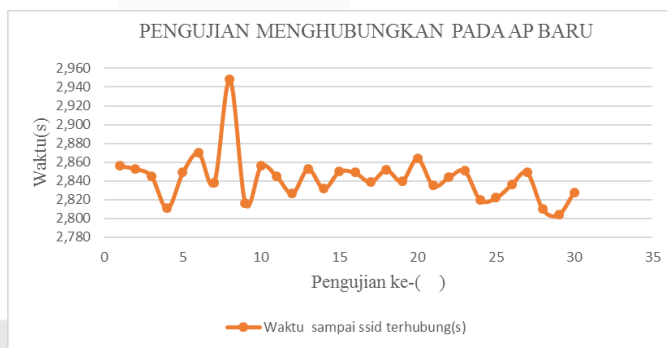
Gambar 8 Penggantian Nama Ssid Dan Password Di Server



Gambar 9 Perangkat Menghubungkan Pada AP Baru

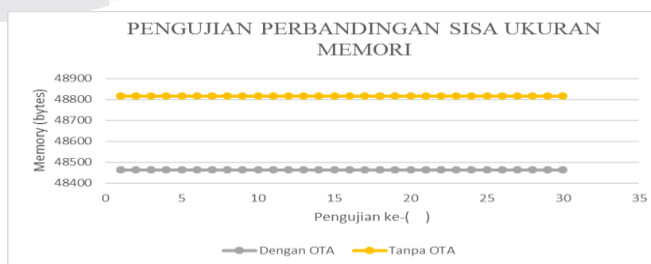
B. Pengujian Pengaruh OTA Pada Performa Mikrokontroler

Pengujian pada perangkat ini memiliki tujuan untuk mendapatkan rata-rata kecepatan menghubungkan pada AP baru setelah dilakukan pembaruan OTA. Pada tahap pengujian sistem ini akan dilakukan percobaan sebanyak tiga puluh (30) kali. Untuk batasan pengukuran waktu yang digunakan dimulai saat perangkat mencoba menghubungkan pada access point dan berhenti saat telah terhubung dengan antares.



Gambar 10 Pengujian Menghubungkan Pada AP Baru

C. Pengujian perbedaan ukuran memori



Gambar 11 Pengujian Perbedaan Ukuran Memori

Hasil pengujian perbandingan ukuran memori yang tersisa dengan menggunakan OTA hasil yang didapat adalah 48464 bytes sedangkan untuk percobaan yang tidak menggunakan OTA adalah 48816 bytes. Dari perbandingan hasil tersebut maka penggunaan OTA dalam kodingan memakan memori lebih banyak dibandingkan tanpa menggunakan OTA.

Dari perbandingan kedua sisa memori yang telah digunakan dapat dikatakan bahwa walaupun terdapat perbedaan memori yang tersisa namun hal itu tidak mempengaruhi performa dari perangkat karena perbedaan yang tidak signifikan

IV. KESIMPULAN

Berdasarkan hasil pengujian yang dilakukan sebanyak lima kali proses OTA yang didapat berikut beberapa kesimpulan dari pengujian yang dilakukan:

1. Percobaan yang telah dilakukan untuk mendapatkan tingkat keberhasilan sebanyak lima kali dalam melakukan OTA mendapatkan hasil tingkat keberhasilan sebesar 100%.
2. Pengujian waktu yang dibutuhkan untuk perangkat dapat terhubung pada access point yang baru rata-rata waktu yang didapat adalah 2,843 detik
3. Pengujian perbandingan ukuran memori yang tersisa saat menggunakan OTA hasil yang didapat adalah 48464 bytes sedangkan untuk percobaan yang tidak menggunakan OTA adalah 48816 bytes. Dari hasil tersebut maka penggunaan OTA memakan lebih banyak memori dibanding saat tidak menggunakan OTA tapi tidak terdapat pengaruh yang signifikan yang diakibatkan oleh perbedaan memori pada keduanya.

V. SARAN

Proses Over The Air update (OTA) yang dilakukan dapat dikembangkan dengan menambahkan parameter-parameter lain seperti tingkat keamanan dalam proses OTA ataupun kompleksitas firmware yang akan diperbarui

VI. REFERENSI

- [1] D. Lund and et.al, "Worldwide And Regional Internet Of Things (Iot) 2014– 2020 Forecast: A Virtuous Circle Of Proven Value And Demand," International Data Corporation (IDC), 2014.
- [2] S. Reißmann and C. Pape, "An Over the Air Update Mechanism for ESP8266 Microcontrollers," ICSNC 2017 Twelfth Int. Conf. Syst. Networks Commun. An, no. October, pp. 11–17, 2017.
- [3] ESP8266, "OTA Update · ESP8266 Arduino Core," Available: http://esp8266.github.io/Arduino/versions/2.0.0/doc/ota_updates/ota_updates.html, [diakses 28 september 2020].
- [4] Eru Puspita, "Teknik Update Program Secara On-line (Over The Air, OTA) Pada Sistem Berbasis Mikrokontroler Menggunakan Modem GPRS," In: IES, 2012
- [5] Xinchu He, "A Blockchain-Based Framework to Secure Over-The-Air Firmware Updates for Internet of Things," ProQuest Dissertations Publishing, 2019.
- [6] N. Nikolov, "Research Firmware Update Over the Air from the Cloud," Proc. XXVII International Scientific Conference Electronics - ET2018, Hal. 1-4, 2018.
- [7] R. Fielding and et.al, "Hypertext Transfer Protocol -- HTTP/1.1," The Internet Society, Hal. 1, 1999.
- [8] Allegro MicroSystems, Inc., "Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kVRMS Voltage Isolation and a Low-Resistance Current Conductor," ACS712, 2012, Available: <https://www.allegromicro.com › acs712datasheet>, [diakses 23 desember 2021].
- [9] K. S. Budi and Y. Pramudya, "Pengembangan Sistem Akuisisi Data Kelembaban Dan Suhu Dengan Menggunakan Sensor Dht11 Dan Arduino Berbasis Iot," Prosiding Seminar Nasional Fisika (E-Journal) Snf2017, Hal. 3, 2017.
- [10] Turban, E., King, D., Lee, J., Liang, T.P., Turban, D. "A Managerial and Social Networks Perspective," Electronic Commerce 2012, (2012).
- [11] Saleha, B., Nasution, S. M., and Prasasti, A. L. "Design of IOT-Based Smart Laundry Applications Using Fuzzy Algorithms," 2020 International Conference on Information Technology Systems and Innovation, 2020.
- [12] B.B. Brown, "Over-the-Air (OTA) Updates in Embedded Microcontroller Applications: Design TradeOffs and Lessons Learned," Analog Dialogue 52-11, November 2018, 2018
- [13] K. C. Okafor, G. C. Ononiwu, and et.al, "Development of Arduino Based IoT Metering System for On-Demand," Vol. 7(23), Jan. 2017.
- [14] Darko Cvetkovski, and et.al, "A Low-Cost DSP Development system featuring freescale's MC56F8322," INFOTEH-JAHORINA Vol. 10, Ref. E-II-9, p.567-570, 2011.