

Pemanfaatan Dua Mikrokontroler Platform IoT Dalam Pengembangan Sistem Parkir

Muhammad Adnan¹, Ratnasari Nur Rohmah²

Teknik Elektro

Universitas Muhammadiyah Surakarta

Surakarta, Indonesia

Email: ¹d400160004@student.ums.ac.id, ²Ratnasari.Rohmah@ums.ac.id

Abstraksi— Kebanyakan tempat parkir di Indonesia masih menggunakan sistem konvensional sehingga sering menyulitkan pengguna parkir dalam mencari tempat parkir kosong. Mereka diharuskan berkeliling untuk mencari tempat parkir kosong yang dapat digunakan. Hal ini disebabkan tidak adanya sistem informasi tempat parkir yang bekerja secara real time. Penelitian ini bertujuan mengimplementasikan prinsip *mechine to mechine* menggunakan 2 mikrokontroler saling berkomunikasi melalui platform IoT yaitu Antares.id. Mikrokontroler yang digunakan yaitu NodeMCU ESP8266. NodeMCU A sebagai miniatur area parkir, terdapat LDR yang terletak di setiap tempat parkir, NodeMCU B sebagai miniatur palang parkir, berisikan Motor Servo sebagai palang pintu, Buzzer indikator Bergeraknya palang, dan LCD untuk display informasi. NodeMCU A sebagai master, sedangkan NodeMCU B sebagai slave. Data NodeMCU A dikirim dan tersimpan pada data logger di Antares.id, lalu diambil oleh NodeMCU B. Namun apabila tempat parkir pada NodeMCU A penuh maka palang pintu NodeMCU B tidak bergerak. NodeMCU A juga mengirimkan data ke blynk agar dapat diakses pada smartphone. Peneliti melakukan pengujian pengiriman dan pengambilan pada Antares.id, menggunakan 2 metode yaitu regular dan legacy, sehingga peneliti dapat menerapkan dalam perancangan sistem parkir dengan delay seminimal mungkin, dari hasil pengujian prototype telah bekerja secara baik dengan akurasi posisi kendaraan sebesar 100%, pada pengujian pengiriman dan pengambilan data pada Antares.id. Pengiriman menggunakan metode legacy lebih cepat dibandingkan pengiriman regular memiliki selisih sebesar 4709 millisecond. Pengambilan data legacy lebih cepat dibanding metode regular selisih dalam pengambilan data sebesar 4873 millisecond. Mikrokontroler pada penelitian ini yaitu NodeMCU ESP8266. Sensor dan aktuator yang digunakan adalah LDR (*Light Dependt Resistor*), Motor Servo SG90, LCD (*Liquid Crystal Display*), Multiplexer MUX74HC4067, dan platform IoT yaitu Antares.id dan Blynk.

Kata Kunci— area parkir; IoT; NodeMCU, Servo SG90, Blynk

I. PENDAHULUAN

Seiring berjalannya waktu, penggunaan kendaraan roda 4 (mobil pribadi) semakin meningkat pesat. Menurut data dari

Badan Pusat Statistik (BPS) yang dilansir bps.go.id, pada tahun 2017 angka pengguna mobil di Indonesia sebanyak 15.423.968. Sedangkan pada tahun 2018 terdapat lonjakan sebanyak 1.017.019 menjadi sebesar 16.440.987. Namun peningkatan penggunaan kendaraan roda 4 ini tidak diiringi dengan peningkatan fasilitas. Salah satunya pada sektor area parkir. Mayoritas sistem parkir masih menerapkan sistem parkir konvensional pada area parkir, dimana sistem parkir konvensional masih memiliki kelemahan, salah satunya yaitu tidak adanya sistem informasi kepada pengguna parkir tentang tempat parkir yang kosong. Hal itu menyebabkan pengguna parkir mengalami kesulitan saat memarkir kendaraan, sehingga banyak membuang waktu dan tenaga untuk mencari lokasi parkir yang kosong. Beberapa peneliti telah mengembangkan berbagai sistem untuk mengatasi adanya kelemahan dalam sistem parkir konvensional tersebut. Penelitian yang telah ada dilakukan dengan memanfaatkan sistem elektronik yang dikembangkan khusus untuk mengatasi kelemahan sistem parkir konvensional. Beberapa penelitian telah mengembangkan sistem dengan konsep IoT (*Internet of Things*). *Internet of Thing* (IoT) adalah sebuah konsep di mana objek memiliki kemampuan untuk mentransfer data melalui jaringan dengan memanfaatkan prinsip M2M (*machine to machine*). Konsep ini membantu memperlancar beberapa aspek kehidupan manusia dimana suatu peralatan dapat berinteraksi dengan peralatan lain tanpa mengikut sertakan campur tangan manusia. Pemanfaatan IoT dapat menjadi solusi dalam proses monitoring, controlling, maupun eksekusi, dalam suatu sistem. Salah satu piranti elektronik yang digunakan dalam pengembangan sistem berbasis IoT adalah MCU ESP8266. Penelitian yang dilakukan oleh

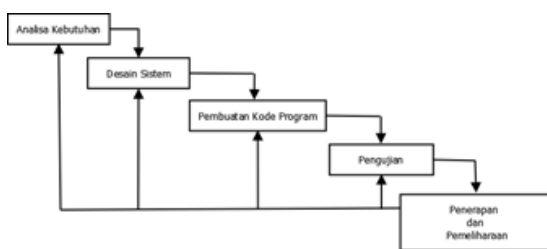
Yuliansyah [1] menunjukkan bahwa pengiriman data dari mikrokontroler menuju komputer server dapat dilakukan secara wireless. Pengiriman data pada penelitian ini menggunakan modul ESP8266. NodeMCU ESP8266 mampu mengirim data dengan kecepatan maksimum 5271 data per menit. Limantara [2] melakukan penelitian untuk mengembangkan sistem parkir, parkir berbasis sensor ultrasonic dan internet of thing, menggunakan chip ESP8266 sebagai mikrokontroler. Sistem ini dilengkapi sensor ultrasonic HC-SR04 yang digunakan untuk mengukur suatu object. Informasi tersebut akan diproses dan diolah pada chip ESP8266 yang menghasilkan informasi tentang tempat parkir yang akan dikirim dan akan diterima oleh access point dan akan diteruskan ke server manajemen yang berbasis web. Sedangkan data informasi akan berada

pada cloud yang akan dikirim berupa software di smartphone pangemudi. Penelitian ini menghasilkan prototype pemodelan sistem parkir kosong menggunakan sensor ultrasonic yang dapat diakses melalui gadget pengguna parkir. Hardiyanto [3] melakukan penelitian dengan mengembangkan sistem parkir. Sistem parkir yang dikembangkan dengan membuat bangunan sistem parkir yang dapat menghitung jumlah mobil secara otomatis, menggunakan Atmega 8535 sebagai mikrokontroler dan buzzer, SRF04 ultrasonic sebagai aktuator dan sensor, serta LCD sebagai penampil informasi. Sensor ultrasonic diletakkan pada pintu masuk dan keluar palang pintu. Ketika sebuah mobil telah masuk dan melewati pintu palang parker, sensor akan mendeteksi dan banyak mobil pada tampilan LCD pun akan menambah 1, jika mobil keluar dan melewati palang pintu keluar maka banyak mobil akan berkurang 1. Buzzer berfungsi sebagai indikator mobil masuk dan keluar tempat parkir. Penelitian ini menghasilkan prototype sistem parkir dengan system parkir yang dapat menghitung jumlah mobil diarea parkir.

Berdasarkan penelitian-penelitian yang telah dilakukan maka peneliti memiliki gagasan untuk membuat sistem tempat parkir mengimplementasikan prinsip M2M (machine to machine) yang dilakukan oleh 2 mikrokontroler yang saling terhubung dengan platform IoT, dan sistem tersebut dapat diakses melalui smartphone. Sistem ini dapat memberikan informasi kepada pengguna parkir terkait tempat parkir yang dapat digunakan sehingga dapat membantu pengguna parkir, selain itu peneliti ingin menguji prinsip M2M dengan memanfaatkan 2 mikrokontroler yang saling berhubungan dengan media parkir sebagai penerapannya karena belum adanya penelitian tentang itu, sehingga dalam perkembangannya dapat memperluas tempat parkir tanpa harus membongkar/merubah tempat parkir yang ada tinggal menghubungkan dengan system yang ada, selain itu system parkir yang dikembangkan dapat diakses melalui smartphone sehingga dapat mempermudah pengguna parkir, selain itu peneliti juga meneliti perbandingan waktu yang diperlukan dalam mengirim maupun mengambil data pada platform IoT, menggunakan 2 metode yaitu regular dan legacy, sehingga peneliti mampu mengoptimalkan penggunaan platform IoT dalam merancang sistem parkir yang digunakan.

II. METODE PENELITIAN

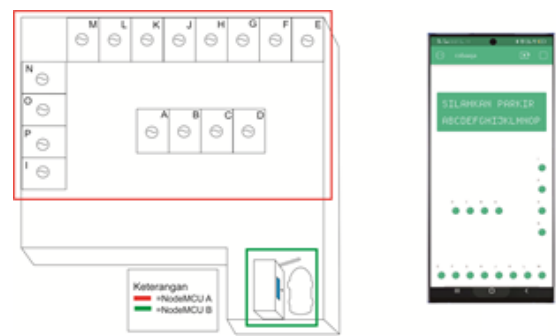
Model metode yang digunakan pada penelitian ini menggunakan model eksperimen simulasi model yang menghasilkan *prototype*. Metode yang digunakan dalam proses pelaksanaan menggunakan model *System Development Life Cycle (SDLC)* dengan pendekatan *waterfall* yang dapat dilihat pada Gambar 1.



Gambar 1. Metode *Waterfall*

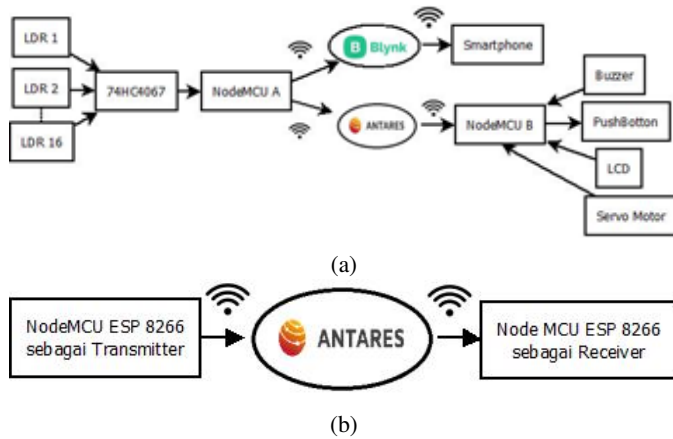
A. Analisis Kebutuhan

1) *Prinsip Kerja Alat*: Alat yang akan dibuat dirancang dengan kemampuan untuk memberi kemudahan pengguna parkir dalam memberikan informasi tentang tempat parkir kosong yang ingin digunakan secara real time, sehingga dapat mempermudah pengguna parkir dalam mencari tempat parkir. Alat ini menggunakan prinsip M2M dengan memanfaatkan 2 mikrokontroler yang saling berkomunikasi. Mikrokontroler yang digunakan adalah NodeMCU ESP8266 yaitu NodeMCU A dan NodeMCU B. Gambar 2 desain prototype alat dan UI pada smartphone dapat menunjukkan bahwa NodeMCUA berisikan sensor LDR yang berada pada zona parkir. NodeMCU B menjadi miniatur palang informasi yang berisikan push botton, motor servo, buzzer, dan LCD. LDR pada NodeMCU A berfungsi sebagai penanda keberadaan mobil. Terdapat 16 sensor LDR pada tempat parkir yang dapat digunakan yaitu A,B,C,D,E,F,G,H,I,J,K,L,M,N,O, dan P. Apabila tidak ada mobil disalah satu tempat parker, maka nilai LDR pada area tempat parkir tersebut 0000000000000000 (A,B,C,D,E,F,G,H,I,J,K,L,M,N,O). Begitu pula sebaliknya, apabila terdapat mobil disalah satu tempat, maka slot tersebut akan bernilai 1. Misalnya tempat parkir yang digunakan adalah A, maka nilai LDR dari area parkir tersebut 1000000000000000 begitupun seterusnya.



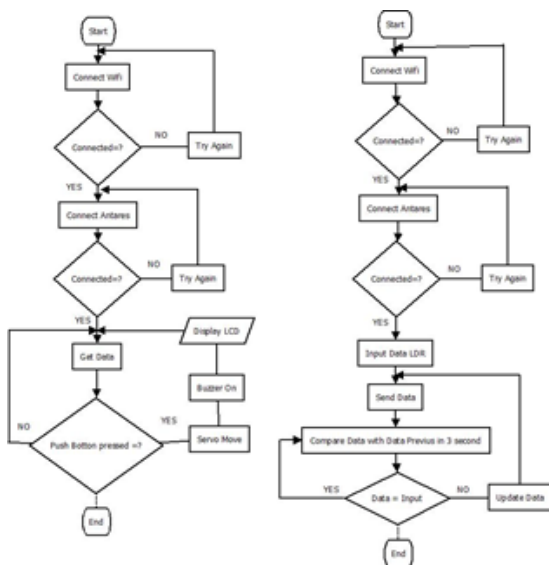
Gambar 2. Desain *Prototype* Alat dan Tampilan UI pada *Smartphone*

NodeMCU A akan membaca hasil sensor LDR pada area parkir setiap 3 ms sekali dan akan mengirim hasil pembacaan ke platform IoT yaitu *Antares.id* dan *blynk*. Data yang dikirim ke *Antares.id* akan akan diambil oleh NodeMCU B lalu akan ditampilkan pada LCD. Push botton pada NodeMCU B sebagai *trigger* motor *servo*, yang bertindak sebagai palang pintu, dan *buzzer*, serta menampilkan informasi parkir terbaru pada LCD. Selain itu, data yang dikirim ke *Blynk* dapat di akses melalui smartphone. Ketika ada mobil yang ingin masukkan menekan *pushbotton*, lalu pada LCD akan menampilkan informasi tempat parkir kosong parkir yang dapat digunakan kosong dan motor *servo* akan menggerakkan palang pintu untuk membuka, dan *buzzer* akan berbunyi. Setelah mobil akan masuk ke area parkir dan mengisi salah satu tempat parkir, nilai LDR pada area parkir tersebut akan berubah dan akan diproses pada NodeMCU A. Data kondisi parkir ini akan dikirim ke *Antares.id* untuk memperbarui informasi pada NodeMCU B, sehingga informasi pada NodeMCU B akan selalu *update* secara *real time*. Selain itu, NodeMCU A akan mengirim ke *Blynk* guna memperbarui informasi yang ada pada smartphone, seperti terlihat pada Gambar 2. Dalam pengujian kecepatan pengiriman



Gambar 3. Aktifitas Antares (a) Diagram Blok Prototype Sistem Parkir (b) Diagram Blok Pengujian Pengiriman dan Pengambilan Data

dan pengambilan data pada platform IoT Antares.id, digunakan 2 mikrokontroler yaitu NodeMCU ESP8266 yang memiliki fungsi yang berbeda. NodeMCU pertama berfungsi sebagai transmitter sedangkan nodeMCU kedua berfungsi sebagai receiver. NodeMCU pertama akan mengirim data random yang diberi range antara 0 sampai 100 yang akan dikirim ke platform IoT, sedangkan nodeMCU kedua akan mengambil data tersebut. Peneliti memanfaatkan perhitungan pada fungsi `millis()` pada pemrograman mikrokontroler untuk mengetahui waktu yang dibutuhkan dalam mengirim maupun menerima data.

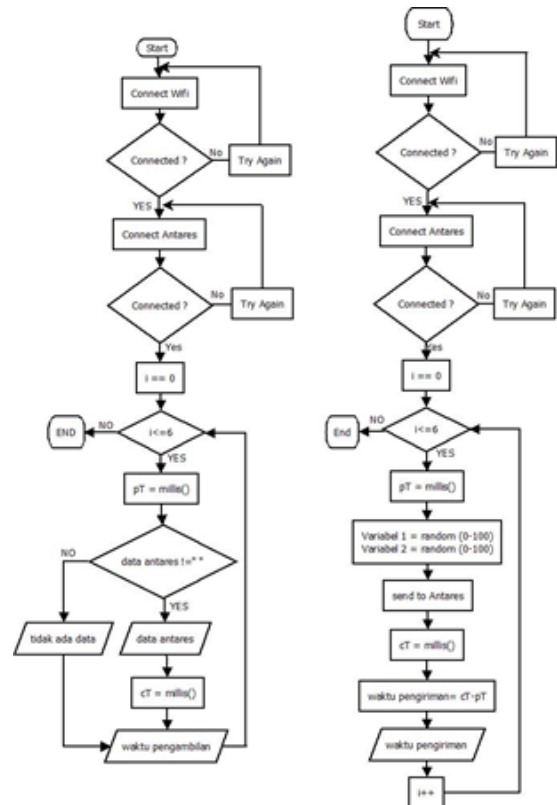


Gambar 4. Diagram alur pemrograman pada prototype sistem parkir

2) **Kebutuhan Perangkat Keras:** Desain sistem yang dikembangkan di atas akan memerlukan beberapa perangkat keras, diantaranya mikrokontroler, sensor, motor *servo*, *push botton*, *LCD*, *buzzer*, dan modul I2C. Mikrokontroler dalam sistem ini menggunakan NodeMCU ESP8266 karena sudah tersedia modul wifi yang sudah terpasang secara *build in* dengan *board*. Sensor yang digunakan pada system yaitu LDR (*Light Dependt Resistor*) sebagai pertanda keberadaan mobil. Motor servo yang digunakan pada sistem ini SG90 yang berfungsi sebagai penggerak pintu masuk dan keluar. Sedangkan modul i2c yang digunakan yaitu *multiplexer* 74HC4067. Penggunaan

multiplexer ini digunakan untuk mengurangi penggunaan pin I/O yang berlebihan pada NodeMCU sehingga penggunaan pin I/O dapat efisien. LCD pada sistem ini digunakan sebagai display informasi. Pada percobaan pengujian pengiriman dan pengambilan data, peneliti memanfaatkan 2 mikrokontroler NodeMCU ESP8266 sebagai *transmitter* dan *receiver*.

3) **Kebutuhan Perangkat Lunak:** Teks editor yang digunakan adalah Arduino IDE. Hal ini dikarenakan Arduino IDE memiliki UI (User Interface) yang bersifat *User Friendly* sehingga mempermudah dalam penggunaan dan eksekusi yang melibatkan *library*.



Gambar 5. Diagram alur pengujian pengiriman dan pengambilan data Antares.id

B. Perancangan Alat

Perancangan Alat prototipe yang digunakan mengacu pada diagram blok yang dapat dilihat pada Gambar 3 (a), sedangkan pada pengujian uji kecepatan pengiriman dan pengambilan data dapat dilihat pada Gambar 3 (b).

1) **Perancangan Perangkat Keras:** Model perancangan perangkat keras yang digunakan memiliki 16 buah LDR sebagai luaran tempat mobil yang akan diproses dan dikirim oleh NodeMCU A ke Antares.id. Data tersebut akan diambil dan diproses NodeMCU B. Proses tersebut menghasilkan informasi tentang tempat parkir yang kosong dan ditampilkan pada LCD di NodeMCU B. *Pushboton* untuk menggerakkan Motor *Servo* yang berperan sebagai palang pintu. Selain mengirim ke *Antares.id*, NodeMCU A juga mengirim data ke *blynk* sehingga dapat di akses melalui *smartphone*, Selain itu peneliti juga menguji kecepatan pengiriman dan pengambilan data secara *legacy* dan *reguler* pada platform IoT *Antares.id* menggunakan

2 NodeMCU ESP8226, NodeMCU pertama sebagai *transmitter* dan NodeMCU kedua sebagai *receiver*.

2) *Perancangan Perangkat Lunak*: Model perancangan lunak pemrograman arduino yang digunakan yaitu teks editor Arduino IDE dengan bahasa C dan C++. Diagram alir yang digunakan dalam pemrograman mikrokontroler pada prototype dapat dilihat pada Gambar 4, sedangkan diagram alir yang digunakan dalam pengujian pengiriman dan pengambilan pada Antares.id dapat dilihat pada Gambar 5.

III. HASIL PENELITIAN

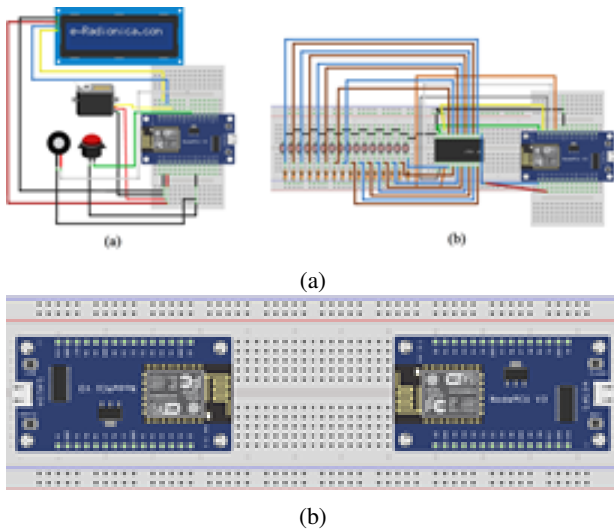
A. Skematik Rangkaian

Skematik Rangkaian yang digunakan pada penelitian ini dapat dilihat pada Gambar 6 (a) sebelah kiri untuk Skematik NodeMCU A yang berisikan 16 buah LDR sebagai inputan, 1 buah Multiplexer untuk penghubung antara LDR dengan mikrokontroler, dan 1 Buah Mikrokontroler NodeMCU sebagai controller, sedangkan pada Gambar 6 (a) sisi kanan untuk Skematik NodeMCU B yang berisikan 1 motor servo sebagai palang pintu parkir, 1 push botton, 1 buah buzzer dan 1 LCD untuk display informasi, dan 1 buah NodeMCU sebagai controller, sedangkan pada pengujian pengiriman dan pengambilan data pada Antares.id, peneliti menggunakan 2 nodeMCU ESP8266 yang bekerja sebagai transmitter dan receiver seperti yang terlihat dalam Gambar 6 (b).



Gambar 7. Model Perangkat

penelitian yang telah dilakukan yaitu berupa model miniatur prototype area parkir yang terbuat dari *yellow board paper* yang di atasnya akan diberi LDR kemudian akan ditutupi oleh mobil yang akan parkir dan di atasnya akan diberi tiang untuk lampu sebagai pentrigger untuk LDR sendiri. Sedangkan pada NodeMCU B akan menjadi miniatur palang pintu informasi. Penjelasan tersebut dilihat pada Gambar 8 (a) untuk prototype dari tempat parkir, dan selanjutnya luaran akan ditampilkan dalam *smartphone*. sedangkan untuk pengujian pengiriman dan pengambilan data dapat dilihat pada Gambar 8 (b).



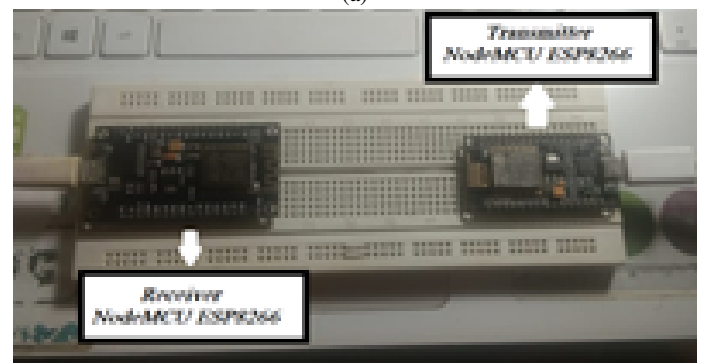
Gambar 6. Skema rangkaian (a) NodeMCU A dan Node MCU B (b) Skema penelitian pengiriman dan pengambilan data

B. Prototype Alat

Pada model prototype penelitian yang telah dilakukan terdapat 2 kontroller yang saling berkomunikasi melalui platform IoT Antares.id yaitu NodeMCU A dan NodeMCU B. NodeMCU A terdapat 16 buah LDR sebagai input dan modul multiplexer yang menghubungkan LDR dengan NodeMCU A dan NodeMCU B yang menjadi miniatur gebang parkir yang berisikan LCD sebagai display informasi, sedangkan motor servo sebagai palang pintu, buzzer sebagai penanda bunyi, push botton sebagai pentrigger motor servo, dan LCD. Bagian-bagian tersebut dapat dilihat pada Gambar 7. Hasil akhir dari



(a)



(b)

Gambar 8. (a) Prototype Hasil Penelitian (b) Percobaan Pengujian Pengiriman dan Pengambilan data



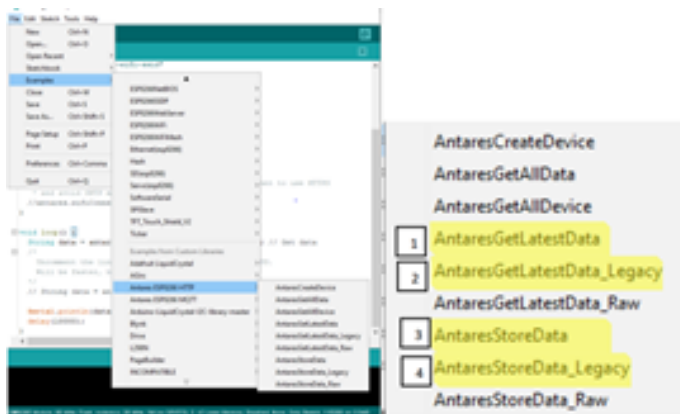
Gambar 9. Tampilan display sistem LCD

C. Hasil Penelitian dan Pembahasan

1) *Pengujian Pengiriman Data:* Bahwasanya tipe pengiriman menuju Platform Antares dilakukan secara *legacy* dengan cara *unformatable* sehingga dapat mempercepat pengiriman data. Proses tersebut dilakukan setiap 3 detik. Untuk tampilan pada Serial Monitor NodeMCU A sebagai pengirim input dan untuk pengiriman pada platform IoT *Antares.id*, selain itu, NodeMCU A juga mengirimkan data ke *Blynk*.

```
Blynk.run();
delay(3000);
```

2) *Pengujian Pengambilan Data:* Pengambilan data yang dilakukan dengan tipe penerimaan *legacy*, pengambilan data dilakukan dengan metode GET dapat dilihat pada Serial Monitor dan data yang diambil akan diolah dan ditampilkan pada papan informasi yang di tampilkan di LCD dapat dilihat pada Gambar 9. Selain itu, data juga dapat di akses ataupun dimonitoring menggunakan *smartphone* melalui *Blynk* (dapat dilihat pada Gambar 10).



Gambar 10. Penggunaan Library Antares.id pada Arduino IDE

3) *Pengujian Kecepatan Pengiriman Data:* Pada Penelitian ini peneliti membandingkan kecepatan pengiriman dan pengambilan data pada platform *Antares.id*, menggunakan dua metode yaitu *legacy* dan reguler, dengan memanfaatkan penggunaan *library* pada Arduino IDE dapat dilihat pada Gambar 10, penggunaan *library* untuk pengiriman dan pengambilan data secara reguler (1 dan 3) dan *legacy* (2 dan 4), dan untuk nilai yang digunakan dalam pengiriman data peneliti menggunakan fungsi *random* yang bernilai 0 sampai 100 seperti yang terlihat dalam Gambar 11 (a), sedangkan untuk mengetahui waktu yang digunakan dalam pengiriman dan pengambilan data peneliti memanfaatkan fungsi *millis()* pada Arduino IDE seperti yang terlihat pada Gambar 11) (b). Terdapat dua tipe

pengujian kecepatan pengiriman data yaitu pengujian reguler dan pengujian *legacy*.

```
// Variables
int varA = random(0, 100);
int varB = random(0, 100);
```

(a)

```
previousTime = millis();
Serial.println("Percobaan ke"+String(i));
// Variables
int varA = random(0, 100);
int varB = random(0, 100);
// Add Variable data to Storage buffer
antares.add("Variable A", varA);
antares.add("Variable B", varB);
// Send from buffer to Antares
antares.send(projectName, deviceName);
currentTime = millis();
delayTime = previousTime - currentTime;
Serial.println("Kecepatan Pengiriman menggunakan
```

(b)

Gambar 11. Kode pemrograman (a) Penggunaan fungsi *random* dalam penelitian (b) Penggunaan fungsi *millis* dalam penelitian

Untuk metode reguler, peneliti melakukan pengujian dan pengambilan data menggunakan metode reguler sebanyak 6 kali pengujian, pada pengiriman dan pengambilan data pada metode reguler ini memiliki kesamaan menggunakan pemodelan *request sent*, sedangkan untuk tampilan dalam dilihat dalam *Antares.id*. Sedangkan untuk metode *Legacy* maka penelitian dilakukan dengan menggunakan uji coba sebanyak 6 kali percobaan untuk pengambilan dan pengiriman data menggunakan metode *legacy*, pada metode *legacy* menggunakan pemodelan POST untuk pengiriman data dan GET untuk pengambilan data serta untuk tampilan data ada pada *Antares.id*.

Pengujian sistem parkir dilakukan dengan cara memberi informasi tentang tempat parkir yang kosong untuk pengguna. Berikut beberapa hasil percobaan simulasi pada penelitian yang telah dilakukan pada sistem parkir (dapat dilihat pada Tabel I), Sedangkan Hasil penelitian perbandingan kecepatan pengiriman dan pengambilan data pada platform IoT *Antares.id* menggunakan dua metode yaitu *legacy* dan reguler dapat dilihat pada Tabel II. Menurut Tabel I. Peneliti melakukan

Tabel I. Percobaan Simulasi

Percobaan	Jumlah Kendaraan Simulasi	Jumlah tempat parkir yang digunakan	Jumlah tempat parkir kosong yang dinarakan	Presentase Keberhasilan
Percobaan 1	1	1	15	100%
Percobaan 2	5	5	11	100%
Percobaan 3	9	9	7	100%
Percobaan 4	16	16	0	100%

percobaan sebanyak 4 kali percobaan, pada percobaan pertama menggunakan sebuah kendaraan yang berada pada area parkir, maka sistem informasi akan memberitahu bahwasanya ada 15 tempat parkir yang dapat digunakan. Pada percobaan kedua

terdapat 5 buah kendaraan yang berada pada area parkir, maka sistem parkir akan menghasilkan rekomendasi tempat parkir kosong yang dapat digunakan, sebanyak 11 tempat begitu pula pada percobaan 3 dan percobaan 4. Pada 4 percobaan tersebut memiliki persentase keberhasilan sebesar 100%.

Tabel II. Percobaan Pengiriman dan Pengambilan

Percobaan	Speed Average ping WiFi=171ms			
	Reguler		Legacy	
	Pengiriman data ke Antares.id (millisecond)	Pengambilan data ke Antares.id (millisecond)	Pengiriman data ke Antares.id (millisecond)	Pengambilan data ke Antares.id (millisecond)
Percobaan 1	49050	48880	1223	149
Percobaan 2	49004	49132	3839	158
Percobaan 3	48930	48922	1289	159
Percobaan 4	48749	49114	1276	175
Percobaan 5	48739	49226	1255	152
Percobaan 6	48235	48094	1239	174
Rata-rata	48785	48895	1687	161

Berdasarkan Tabel II. Peneliti melakukan percobaan sebanyak 6 percobaan dengan kecepatan rata-rata ping WiFi sebesar 171 ms, dapat dilihat dari percobaan di atas rata-rata waktu yang dibutuhkan dalam pengiriman data ke Antares dengan metode reguler selama 48785 ms, sedangkan untuk pengambilan datanya selama 48895 ms, hal ini jauh lebih lama dibanding pengiriman dan pengambilan data yang dilakukan secara legacy, untuk pengiriman data secara legacy hanya memerlukan waktu 1687 ms, sedangkan untuk pengambilan data hanya memerlukan waktu 161 ms, sehingga selisih pengiriman data yang dilakukan secara reguler dan legacy sebesar 4709 ms, sedangkan untuk pengambilan data memiliki selisih sebesar 48734 ms.

IV. PENUTUP

Setelah melakukan penelitian, perancangan, dan pengujian alat tersebut dapat diambil kesimpulan sebagai berikut. Tingkat

Keberhasilan pada hasil uji coba simulasi dengan membandingkan jumlah dan posisi kendaraan pada area parkir mencapai 100%. Implementasi IoT serta prinsip M2M pada sistem parkir cerdas sangat memungkinkan untuk dilakukan, sehingga informasi tempat parkir kosong dapat diketahui secara cepat. Perbedaan pengiriman pada platform Antares menggunakan reguler dan legacy yaitu pada legacy menggunakan data dengan format JSON, sedangkan pada reguler tidak menggunakan format JSON. Hal ini yang menyebabkan legacy lebih cepat dalam pengiriman data maupun pengambilan data, untuk pengiriman data secara legacy memiliki rata-rata waktu pengiriman sebesar 1687 ms sedangkan pengiriman data yang dilakukan secara reguler sebesar 48785 ms, sehingga memiliki selisih sebesar 4709 ms. Pengambilan data pada Platform IoT menggunakan legacy yang menggunakan format JSON hal ini menyebabkan pengambilan data secara legacy lebih cepat dibanding reguler, untuk pengambilan data yang dilakukan secara legacy memiliki rata-rata waktu sebesar 161 ms, Sedangkan untuk reguler sebesar 48895 ms, sehingga memiliki selisih sebesar 48734 ms.

DAFTAR PUSTAKA

- [1] H. Yuliansyah, "Uji kinerja pengiriman data secara wireless menggunakan modul esp8266 berbasis rest architecture," *ELECTRICIAN Jurnal Rekayasa dan Teknologi Elektro*, vol. 10, no. 2, Mei 2016.
- [2] A. D. Limantara, Y. C. S. Purnomo, dan S. W. Mudjanarko, "Pemodelan sistem parkir kosong berbasis sensor ultrasonic dan internet of things (iot) pada lahan parkir diluar jalan," *PROSIDING SEMNASTEK 2017*, 2017.
- [3] R. Hardiyanto, A. Rochim, dan I. Pertiwi, "Pembuatan penghitung jumlah mobil otomatis berbasis mikrokontroler atmega 8535 menggunakan sensor ultrasonik," *Jurnal Teknologi dan Sistem Komputer*, vol. 3, p. 185, 04 2015.