

Implementasi Internet of Things Pada Sistem Pencarian Parkir Berbasis Mikrokontroler Node-MCU

Aqmarina Nur Alifiani Subingat ^{*1}, Nataniel Dengen², Anton Prafanto³, Medi Taruk⁴

^{1,2,3,4}Program Studi Informatika, Fakultas Teknik, Universitas Mulawarman

e-mail: ^{*1} aqmarinanas@gmail.com, ²ndengen@gmail.com, ³antonprafanto@fkti.unmul.ac.id,
⁴meditaruk@gmail.com

Abstrak

Penelitian ini bertujuan untuk memudahkan pengguna yang sedang mencari parkir untuk melakukan pencarian parkir dari luar lokasi pada berbagai lokasi yang ingin dituju. Tujuan penelitian tersebut dapat dicapai dengan mengimplementasikan Internet of Things pada sistem pencarian parkir berbasis NodeMCU yang nantinya sistem yang di hasilkan berupa website untuk monitoring dari berbagai lokasi yang terdaftar. Untuk membangun sistem pencarian parkir berbasis NodeMCU ini perlu menyambungkan antara perangkat yang berupa rangkaian sensor dan mikrokontroler dengan perangkat lunak yang dibangun menggunakan Web-server yang mendukung. Web-server yang digunakan adalah Thinger.io dimana model implementasi yang digunakan adalah REST API. Perangkat keras nantinya akan mengirim data ke Web-server menggunakan wi-fi yang terdapat pada NodeMCU dan sistem nantinya mendapatkan data dari Web-server dengan mendaftarkan Api dan memasukkan token. Pada penelitian ini objek akan terdeteksi jika objek berjarak 5 cm dari sensor dan objek diam selama 5 detik. Sementara dari hasil pengujian dan implementasi Internet of Things yang telah dilakukan mikrokontroler NodeMCU dapat di implementasi kan dengan baik dan mampu memudahkan pengguna yang sedang mencari parkir karena sistem akan selalu diperbarui selama 5 detik sekali dan perangkat akan membaca adanya objek dalam waktu 5 detik sekali dan sistem dapat diakses dari luar lokasi (online).

Kata kunci— Internet of Things, Web-server, REST API, Mikrokontroler, NodeMCU

1. PENDAHULUAN

Kecenderungan peningkatan kepemilikan kendaraan pribadi dan semakin berkurangnya penggunaan angkutan umum. Berdasarkan kajian Masyarakat Transportasi Indonesia, terdapat kecenderungan penurunan penggunaan angkutan umum sebesar 1% per tahun di kota-kota besar di Indonesia. Bahkan di Jakarta terdapat penurunan lebih besar dibanding rata-rata, yaitu 3% per tahun [1]. Hal tersebut secara tidak langsung juga berimbas terhadap permintaan akan kepemilikan kendaraan pribadi. Masyarakat cenderung ingin instant dalam bepergian langsung dari rumah menuju tempat tujuannya tanpa perlu pergi ketempat pemberhentian kendaraan umum. Menjamurnya kendaraan pribadi mulai berdampak pada ketersediaan lahan parkir di sejumlah lokasi [2]. Permasalahan muncul ketika dalam area gedung atau mall yang luas pengendara tidak tahu dimana parkir yang kosong atau pengendara bahkan tidak tahu apakah parkir sudah penuh..

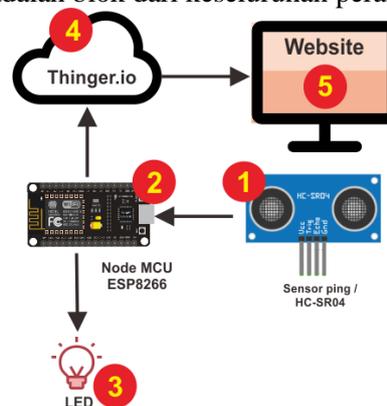
Mobil merupakan salah satu objek parkir yang perlu perhatian khusus dalam hal parkir, karena mobil dalam ukuran lebih besar dari motor roda dua dan dimensi dalam penempatan dalam area parkir lebih banyak memakan tempat [3]. Sehingga membutuhkan penataan yang rapi dan tertata. Saat ini di kawasan sebagian besar wilayah di Samarinda sendiri tempat parkir yang tersedia masih menggunakan metode manual dan kurang efisien sehingga diperlukannya pengembangan dalam hal penyampaian informasi terhadap pengendara mobil. Untuk mengatasi atau mengembangkan penyampaian informasi maka menggunakan komunikasi serial sebagai

jalur informasi. Dengan mengadopsi teknologi yang saat ini sedang gencarnya di kembangkan yakni Internet of Things atau biasa disingkat menjadi IoT dapat menjadi salah satu solusi untuk mengatasi pencarian parkir yang masih manual saat ini di Samarinda, Dari segi pemanfaatannya, IoT memiliki banyak peluang untuk pengembangan kota yang lebih maju dan menciptakan kota yang bersifat Smart City [4], [5], [6]. Untuk mempermudah kondisi kesulitan pengguna mobil untuk mencari tempat parkir, maka diperlukan suatu aplikasi yang dapat digunakan untuk mengetahui dan mencari jalur ke tempat parkir tersebut. Implementasi Internet of Things sistem pencarian parkir berbasis NodeMCU dapat diangkat karena melihat banyaknya penggunaan kendaraan roda 4, banyaknya pengguna smartphone maupun website, sehingga mempermudah pengendara dalam mengetahui lokasi lahan parkir tanpa harus sulit mencari lahan parkir [7].

2. METODE PENELITIAN

2.1 Blok Diagram Sistem

Perancangan perangkat keras meliputi keseluruhan perancangan, yaitu sensor HC-SR04 sehingga nantinya menghasilkan keluaran berbentuk perubahan warna kolom table pada website. Gambar dibawah ini adalah blok dari keseluruhan perancangan.



Gambar 1 Blok Diagram Sistem

Node MCU berfungsi sebagai pengolahan data antara HC-SR04, lampu LED, untuk berkomunikasi dengan baik dan berfungsi pula sebagai mengirimkan data ke thinger.io berupa jarak dan status dari tempat parkir [7].

Sensor HC-SR04 berfungsi untuk membaca jarak mobil yang akan terparkir pada petak parkir data yang dibaca oleh sensor inilah yang akan dikirim ke thinger.io [8].

Lampu LED berfungsi sebagai penanda pada petak apabila tempat parkir sedang terisi atau kosong. Lampu led disini mendapatkan data dari HC-SR04 apabila status tempat parkir berubah.

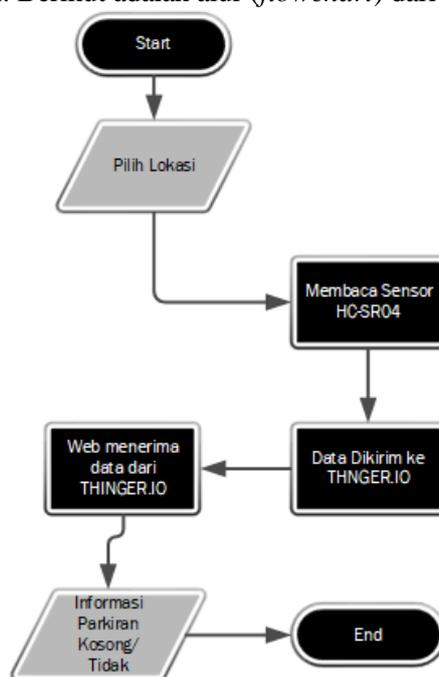
Thinger.io berfungsi sebagai perantara atau perangkat untuk mengkonfersi data dari node MCU menjadi j.son agar data dapat dibaca di berbagai bahasa pemrograman lain. Thinger i.o menyediakan api untuk membaca data dari alat yang dimiliki setiap akunnya [9].

Website berfungsi untuk menerima data dari i.o agar bisa tersampaikan ke pengguna. Pada website ini nantinya akan muncul informasi parkir dari berbagai lokasi dan dimana tiap lokasi akan muncul parkir mana mana saya yang telah terisi dan mana yang belum

2. 2 Flowchart Sistem

Sebelum pembuatan *prototype*, maka diperlukan suatu teknik perancangan yang mempunyai urutan proses kegiatan yang jelas. Diagram alur digunakan sebagai langkah awal pembuatan program. Dengan adanya *flowchart* urutan kegiatan proses akan lebih jelas. Setelah

flowchart selesai disusun, selanjutnya programmer menerjemahkannya ke bentuk program dengan bahasa pemrograman. Berikut adalah alur (*flowchart*) dari *prototype* yang akan dibuat.



Gambar 2 Alur Diagram Pencarian Parkir

Pada alur diagram yang telah di gambarkan diatas dapat dijelaskan bahwa langkah awal adalah pengguna memilih data lokasi yang diinginkan ke aplikasi yang nantinya akan dibuat. Kemudian sistem akan mencari jika data data lokasi yang diinginkan ada maka akan dilanjutkan ke tahap pembacaan sensor. Setelah sistem membaca sensor maka data yang telah dibaca oleh sensor tadi akan di serahkan ke server kemudian server mengirimkan ke aplikasi. Output dari sistem sendiri berupa informasi parkir mana yang kosong dan mana yang terisi dari lokasi yang telah dipilih.

2. 3 Kerangka Konseptual

Kerangka konseptual merupakan suatu bentuk kerangka berpikir yang dapat digunakan sebagai pendekatan dalam memecahkan masalah. Biasanya kerangka penelitian ini menggunakan pendekatan ilmiah dan memperlihatkan hubungan antar variabel dalam proses analisisnya. Adapun kerangka konseptual yang digunakan oleh penulis adalah sebagai berikut.



Gambar 3 Kerangka Konseptual

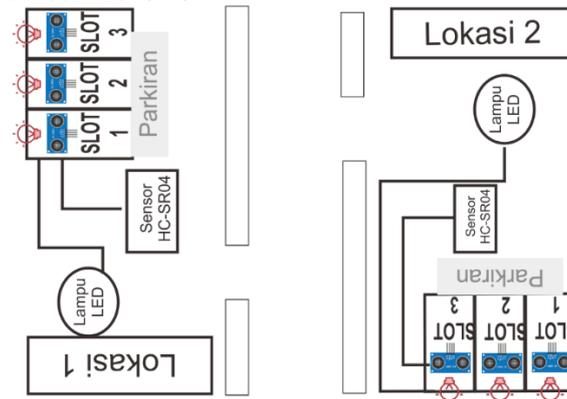
Proses penelitian diawali dengan melakukan identifikasi masalah dan pengumpulan data. Kemudian penelitian ini diawali dengan mempelajari teori-teori yang berhubungan dengan proses perancangan dan pembuatan sistem pencarian parkir menggunakan mikrokontroler. Pustaka yang digunakan adalah yang berhubungan dengan pengembangan program yang sedang dilakukan rujukan dalam hal ini dapat berupa jurnal ilmiah, text book, maupun informasi berupa artikel yang tertera pada internet.

Setelah itu melakukan perancangan desain prototype. Prototype yang telah di desain akan dilengkapi dengan modul Mikrokontroler NodeMCU 8266, Sensor HC-SR04, Kabel Jumper,

Breadboard, LED, dan Kabel USB. Jika desain prototype telah dibuat maka hal yang dilakukan setelahnya adalah membuat aplikasi, rangkaian komponen dan penulisan program. Penulisan program ini menggunakan compiler Arduino IDE dan memakai bahasa C. Jika rangkaian komponen dan penulisan program telah dibuat, kemudian akan diuji dan diimplementasikan kepada sebuah prototype yang memiliki 6 buah LED, 6 buah Sensor HC-SR04, dan 2 buah Node MCU.

2. 4 Denah Peletakan Prototype

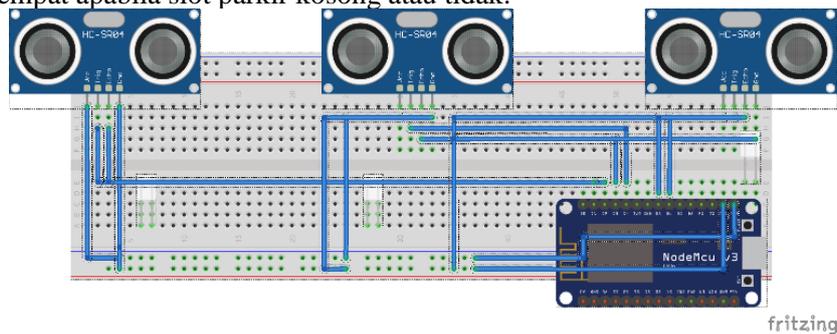
Pada gambar dibawah ini merupakan gambaran denah peletakan prototype pada parkir yang dibuat. Yang dimana terdapat 6 sensor HC-SR04 yang di letakkan di 2 lokasi berbeda dan masing masing lokasi memiliki 3 slot parkir dan terdapat 6 Lampu LED Yang masing masing di letakkan di belakang di sensor HC-SR04.



Gambar 4 Denah Prototype

2. 5 Model Perancangan Pencarian Parkir

Pada gambar dibawah ini merupakan gambaran dari model sistem pencarian parkir. Yang dimana terdapat dua rangkaian dan masing masing rangkaian nya identik yakni masing masing menggunakan 1 NodeMCU ESP8266 sebagai mikrokontroler dimana data yang di dapat akan dikirimkan ke thinger.io, 3 sensor HC-SR04 sebagai pembaca sensor dan lampu led sebagai penanda di tempat apabila slot parkir kosong atau tidak.



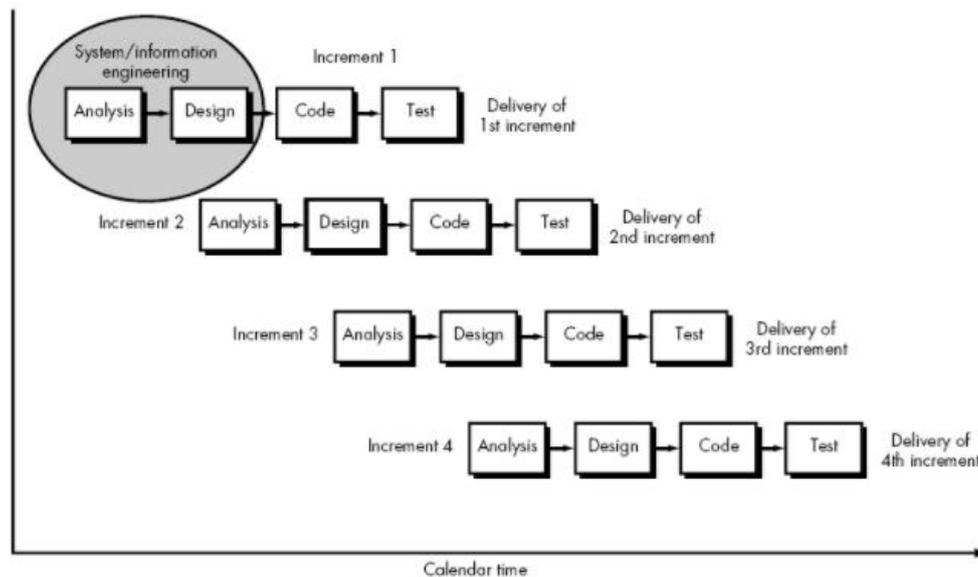
Gambar 5 Model Perancangan Pencarian Parkir

2. 6 Metode Pengembangan Sistem

Pada pengembangannya, sistem ini menggunakan metode incremental adalah model pengembangan sistem pada software engineering berdasarkan requirement software yang dipecah menjadi beberapa fungsi atau bagian sehingga model pengembangannya secara increment /bertahap.

Incremental model termasuk kategori evolutionary software process models karena bersifat iteratif/mengandung perulangan [8]. Hasil proses berupa produk yang makin lama makin lengkap atau bertambah baik sampai versi terlengkap dihasilkan sebagai produk akhir dari proses yang dilakukan. Tahapan awal yang dilakukan pada incremental model adalah penentuan

kebutuhan requirement). Setelah dilakukan analisis kebutuhan, maka dilakukan spesifikasi (specification) dengan menggunakan analisis kebutuhan tersebut sebagai acuannya. Tahap selanjutnya adalah perancangan arsitektur software (architecture design) yang terbuka agar dapat diterapkan sistem pembangunan per-bagian pada tahapan selanjutnya. Apabila desain telah dibuat, maka tahap selanjutnya adalah pengkodean (coding) kemudian dilakukan uji coba (testing). Tahapan-tahapan tersebut dilakukan secara berurutan. Setiap bagian yang sudah selesai dilakukan testing, kemudian dikirim ke pemakai untuk langsung dapat digunakan. Setelah pengembangan suatu increment dimulai, kebutuhan dibekukan dulu hingga increment berikutnya dimulai. Pada incremental model, tiga tahapan awal harus diselesaikan terlebih dahulu sebelum tahap pembangunan tiap modul/increment. Berikut adalah contoh model rekayasa perangkat lunak secara incremental pada Gambar 6.



Gambar 6 Incremental Life Cycle

2. 7 Komunikasi Antar Perangkat

Pada penelitian ini akan menggunakan Thinger sebagai cloud server. Thinger akan menerima berbagai request data melalui REST:API (Representational State Transfer : Application Programming Interface) yang telah disediakan oleh Thinger.io.



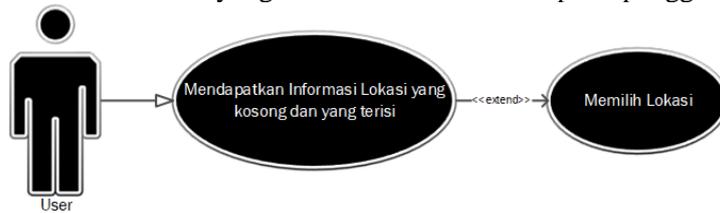
Gambar 7 Device Communication

Informasi yang telah dikumpulkan oleh NodeMCU akan dikirimkan ke Channel Thinger dalam rentang waktu tertentu melalui Metode POST. dan Perangkat Android akan melakukan request data melalui metode GET sebelum menampilkan informasi ke Aplikasi Android. Metode POST dan GET adalah metode yang telah disediakan oleh REST:API Thinger untuk berkomunikasi antara perangkat IoT dan perangkat pengguna.

2. 8 Usecase Diagram

Menggambaran alur proses yang dilakukan oleh pengguna memperlihatkan proses sistem yang akan dirancang. Diagram ini memperlihatkan himpunan use-case dan aktor (suatu

jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.



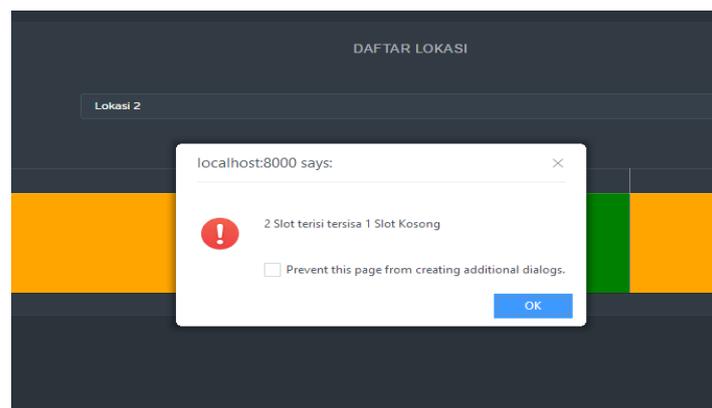
Gambar 8 Usecase Diagram

3. HASIL DAN PEMBAHASAN

Pada bagian ini ditampilkan hasil dan pembahasan terhadap system dan peran. Tahapan ini dilakukan setelah perancangan selesai. Setelah diimplementasikan maka dilakukan pengujian terhadap sistem dan dilihat kekurangan-kekurangan pada - aplikasi untuk pengembangan sistem selanjutnya. Pengujian dilakukan setelah semua rangkaian komponen NodeMCU dan perangkat lunak diintegrasikan menjadi satu sistem. Tahap ini bertujuan untuk mengetahui apakah komponen Sistem Pencarian Parkir berjalan dengan baik dan sesuai yang diharapkan.

3. 1 Implementasi Perangkat Lunak

Tahap implementasi merupakan tahap dimana suatu sistem telah dirancang dioperasikan dalam keadaan yang sebenarnya. Sehingga melalui tahap ini sistem dapat diketahui kelayakannya dalam penggunaan. Berikut ini merupakan implementasi dari pencarian parkir. Pada bagian ini akan muncul slot parkir dengan status parkir yang ada serta kode warna yang menandakan kosong dan terisi. Akan muncul berapa slot yang terisi dan sisa slot yang kosong kemudian warna background slot parkir yang di modelkan akan terlihat berwarna hijau apabila slot parkir terisi dan akan berwarna jingga apabila slot parkir berstatus kosong. Halaman hasil ini akan muncul setelah pengguna memilih lokasi yang ada pada halaman utama dan data yang ditampilkan merupakan data yang real-time. Halaman hasil dapat dilihat pada Gambar berikut.

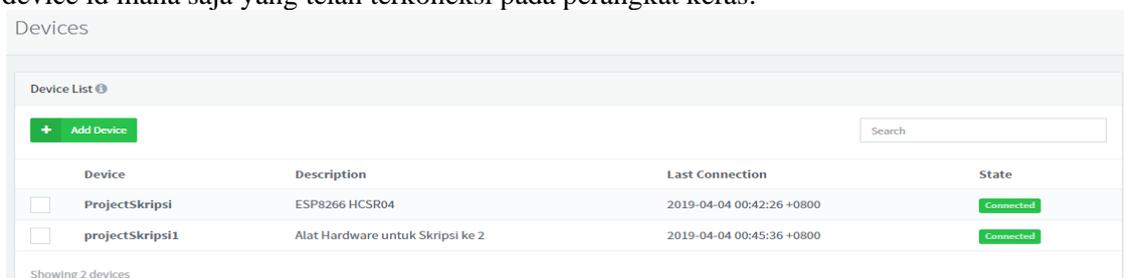


Gambar 10 Implementasi Halaman Hasil

3. 2 Implementasi Koneksi Hardware dan Software

Implementasi juga di terapkan pada koneksi hardware dan software dimana pada poin ini akan dijelaskan bagaimana perangkat keras dan perangkat lunak dapat mengirim data dan menerima data satu sama lain menggunakan Rest API. Method yang digunakan pun ada dua yakni POST dan GET. Perlu dijelaskan bahwa disini penulis menggunakan beberapa tahap untuk mengkoneksikan perangkat keras dan perangkat lunak.

Tahap pertama untuk koneksi Web-server adalah membuat device pada menu device di Web-server Thinger.io pada form pembuatan akan diminta untuk membuat nama device id dan device kredit tersendiri. Device id dan device kredit inilah yang dicantumkan pada saat koneksi perangkat keras dilakukan. Sehingga setelah koneksi perangkat keras dilakukan Web-server akan menerima data yang dikirim oleh perangkat keras. Web-server akan memberikan info device id mana saja yang telah terkoneksi pada perangkat keras.



Gambar 11 Koneksi Web Server

3. 3 Hasil Pengujian Sensor HC-SR04

Pengujian sensor HC-SR04 bertujuan untuk mengetahui kemampuan sensor dalam mendeteksi jarak pada parkir. Pengujian dimulai dari pengujian sensor pada lokasi 1 dan lokasi 2.

Tabel 1 Hasil Uji Coba Parkiran Lokasi 1

No	Jarak	Status	Website	Delay (s)	Lampu
1	<= 5cm	Terisi	Terkirim	7	On
2	<= 5cm	Kosong	Tidak Terkirim	2	Off
3	>5 Cm	Kosong	Terkirim	7	Off
4	>5 Cm	Kosong	Tidak Terkirim	2	Off

Pada tabel 1 menunjukkan keaktifan dari sensor HC-SR04 Pada slot parkir dengan jarak kurang dari 5 cm yang diam dalam waktu lebih 5 detik dan kemudian mengirimkan data ke sistem bahwa slot terisi. Kemudian apabila pada kondisi parkir kosong mulanya kemudian terdapat objek yang melewati sensor dalam waktu kurang dari 5 detik dalam jangkauan jarak kurang dari 5 cm maka objek tersebut tidak akan terbaca dan data tidak terkirim ke Web-server. Objek yang terbaca pada jangkauan sensor lebih dari 5 cm dan dalam waktu lebih dari 5 detik maka status parkir akan menjadi kosong, informasi yang terkirim berupa status parkir yang kosong. Apabila objek menjauhi sensor lebih dari 5 cm dalam waktu kurang dari 5 detik maka data tidak akan terbaca.

Tabel 2 Hasil Uji Coba Parkiran Lokasi 2

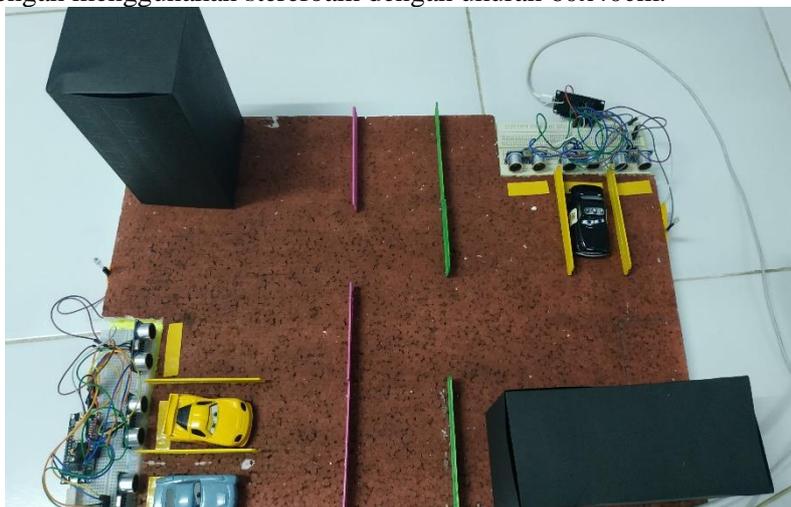
No	Jarak	Status	Website	Delay (s)	Lampu
1	<= 5cm	Terisi	Terkirim	7	On
2	<= 5cm	Kosong	Tidak Terkirim	2	Off
3	>5 Cm	Kosong	Terkirim	7	Off
4	>5 Cm	Kosong	Tidak Terkirim	2	Off

Pada tabel 2 menunjukkan kesamaan pengujian pada lokasi 2, yaitu keaktifan dari sensor HC-SR04 Pada slot parkir dengan jarak kurang dari 5 cm yang diam dalam waktu lebih 5 detik dan kemudian mengirimkan data ke sistem bahwa slot terisi. Kemudian apabila pada kondisi parkir kosong mulanya kemudian terdapat objek yang melewati sensor dalam waktu kurang dari 5 detik dalam jangkauan jarak kurang dari 5 cm maka objek tersebut tidak akan terbaca dan data tidak terkirim ke Web-server. Objek yang terbaca pada jangkauan sensor lebih dari 5 cm dan dalam waktu lebih dari 5 detik maka status parkir akan menjadi kosong,

informasi yang terkirim berupa status parkir yang kosong. Apabila objek menjauhi sensor lebih dari 5 cm dalam waktu kurang dari 5 detik maka data tidak akan terbaca.

3. 4 Model Prototype

Gambar 13 dan gambar 14 merupakan tampilan model prototype dan hasil akhir yang telah dibuat dengan menggunakan sterefoam dengan ukuran 60x40cm.



Gambar 12 Model Prototype

3. 5 Pembahasan

Mikrokontroler NodeMCU berhasil di implementasikan pada monitoring sistem pencarian parkir dengan lancar pada sistem yang telah dibuat dengan catatan perangkat lunak dan perangkat keras harus saling terkoneksi pada internet satu sama lain. Perantara antara perangkat lunak dan perangkat keras adalah web server thinger.io. Thingier.io yang penulis gunakan tidak berbayar sehingga kapasitas untuk satu akun yang terdaftar maksimal hanya dua device yang dapat digunakan. Setiap device yang dibuat memiliki device id dan device credential yang dapat diisi sendiri. Pada saat pengaturan pada perangkat keras yang dilakukan pada aplikasi Arduino IDE perlu mendaftarkan device yang telah dibuat pada akun thingier yang nantinya perangkat keras akan mengirim data ke web server dan perangkat lunak akan mengambil data dari web server. Untuk interaksi antara perangkat lunak, perangkat keras dan Web-server inilah yang dinamakan Rest API. Rest API merupakan suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dimana data yang dikirim dari perangkat keras ke Web-server menggunakan method post dan perangkat lunak mendapatkan data dari Web-server menggunakan method get.

Sedangkan hasil pengujian yang telah dilakukan pada sensor HC-SR04 pada prototype menerangkan bahwa data akan di delay selama 5 detik. Dengan delay tersebut dapat meminimalisir error dimana jika ada objek yang mendekati sensor dengan durasi tidak diam ditempat selama kurang dari 5 detik maka sistem tidak akan membaca adanya objek. Namun apabila objek berdiam diri selama lebih dari 5 detik maka sistem akan membaca adanya objek pada slot parkir yang tersedia.

4. KESIMPULAN

NodeMCU dapat menjadi alat yang sangat murah dan dapat digunakan sebagai sistem pencarian parkir. NodeMCU juga dapat mengirimkan data ke Web-server iot menggunakan module wi-fi yang dimiliki.

Hasil implementasi dan pengujian sistem yang telah dibangun dapat memberikan informasi berupa monitoring pencarian parkir pada lokasi tertentu yang terdaftar dengan delay 5

detik dan dapat dikatakan data tersebut real-time karena data akan update tiap 5 detik sekali. Objek Akan terbaca sensor jika objek berjarak 5 cm atau kurang dari sensor. Aplikasi ini dapat diakses dari luar lokasi karena Web-server yang digunakan sebagai media komunikasi pertukaran data sudah online. Sistem juga melakukan interaksi kepada pengguna yang sedang mencari parkir berupa memilih lokasi yang ingin dituju parkirannya dari luar lokasi dan untuk setiap perubahan data yang terjadi pada suatu lokasi berupa pemberitahuan berbentuk pop up untuk berapa slot yang terisi dan berapa slot yang tersisa sehingga dapat memudahkan pengguna dalam pencarian parkir mobil karena pengguna akan mendapatkan informasi real-time.

5. SARAN

Berdasarkan hasil evaluasi pada Implementasi Internet of Things pada sistem pencarian parkir ini, terdapat beberapa saran untuk pengembangan aplikasi selanjutnya sebagai berikut, pada pengembangan sistem selanjutnya, diharapkan adanya manipulasi data lokasi untuk parkir yang ingin dituju. Dalam pengujian prototype slot port yang dapat digunakan hanya berkisar 9 port sehingga sensor yang dapat dihubungkan terbatas dengan keterbatasan port tersebut diharapkan adanya penelitian yang mengangkat menggunakan mikroncontroller dengan port yang lebih banyak pada penelitian selanjutnya.

Pada pengembangan sistem selanjutnya, diharapkan adanya teori pengaruh delay yang di atur pada hardware dan software. Serta adanya perangkat yang dapat menunjang NodeMCU agar pin yang digunakan dapat lebih banyak untuk menerima data sensor.

DAFTAR PUSTAKA

- [1] S. Aminah, "Transportasi Publik dan Aksesibilitas Masyarakat Perkotaan," *Tek. Sipil*, vol. 9, no. 1, 2018.
- [2] P. D. Saputra and N. Najid, "Pengendalian Penggunaan Kendaraan Pribadi Dengan Strategi Parkir Dan Erp Di Sudirman – Thamrin (Dki Jakarta)," *JMTS J. Mitra Tek. Sipil*, vol. 1, no. 2, p. 9, 2018, doi: 10.24912/jmts.v1i2.2611.
- [3] S. Maulana, Sumarni, and U. M. Barus, "Peranan Dinas Perhubungan Terhadap Penguasaan Sementara Atas Kendaraan Roda Empat Dalam Hal Pelanggaran Parkir Di Kota Medan," *J. Darma Agung*, vol. 29, no. 3, pp. 318–331, 2021.
- [4] A. Camero and E. Alba, "Smart City and information technology: A review," *Cities*, vol. 93, no. March 2018, pp. 84–94, 2019, doi: 10.1016/j.cities.2019.04.014.
- [5] P. J. Basford, F. M. J. Bulot, M. Apetroaie-Cristea, S. J. Cox, and S. J. J. Ossont, "LoRaWan for smart city IoT deployments: A long term evaluation," *Sensors (Switzerland)*, vol. 20, no. 3, 2020, doi: 10.3390/s20030648.
- [6] A. T. Mauludy, D. C. Khrisne, and K. O. Saputra, "Rancang Bangun Aplikasi Pencarian Slot Empat Dengan Pendekatan Computer Vision Parkir Kosong Untuk Kendaraan Roda Empat Dengan Pendekatan Computer Vision," *J. SPEKTRUM*, vol. 7, no. 1, pp. 36–40, 2020.
- [7] Z. Wan, Y. Song, and Z. Cao, "Environment dynamic monitoring and remote control of greenhouse with ESP8266 NodeMCU," *Proc. 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, no. Itnec, pp. 377–382, 2019, doi:

10.1109/ITNEC.2019.8729519.

- [8] Mutinda Mutava Gabriel, “Arduino Uno, Ultrasonic Sensor HC-SR04 Motion Detector with Display of Distance in the LCD,” *Int. J. Eng. Res.*, vol. V9, no. 05, pp. 936–942, 2020, doi: 10.17577/ijertv9is050677.
 - [9] A. L. Bustamante, M. A. Patricio, and J. M. Molina, “Thingier.io: An open source platform for deploying data fusion applications in IoT environments,” *Sensors (Switzerland)*, vol. 19, no. 5, 2019, doi: 10.3390/s19051044.
 - [10] Y. Zhou, E. J. van Kampen, and Q. P. Chu, “Incremental model based online dual heuristic programming for nonlinear adaptive control,” *Control Eng. Pract.*, vol. 73, no. November 2017, pp. 13–25, 2018, doi: 10.1016/j.conengprac.2017.12.011.
-