



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IS184853

***RANCANG BANGUN SISTEM SMART PARKING
BERBASIS ARDUINO UNTUK LAHAN PARKIR
BERTINGKAT
DESIGN AND DEVELOPMENT OF ARDUINO-BASED
SMART PARKING SYSTEM FOR MULTI-STOREY
PARKING LOT***

**LABIB IZZATUR RAHMAN
NRP. 0521154000108**

**Dosen Pembimbing
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.**

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019**



TUGAS AKHIR - IS184853

RANCANG BANGUN SISTEM SMART PARKING BERBASIS ARDUINO UNTUK LAHAN PARKIR BERTINGKAT

**LABIB IZZATUR RAHMAN
NRP. 0521154000108**

**Dosen Pembimbing
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.**

**DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

Halaman ini sengaja dikosongkan



ITS
Institut
Teknologi
Sepuluh Nopember

UNDERGRADUATE THESIS - IS184853

DESIGN AND DEVELOPMENT OF ARDUINO-BASED SMART PARKING SYSTEM FOR MULTI-STOREY PARKING LOT

LABIB IZZATUR RAHMAN
NRP. 05211540000108

Supervisor
Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Information and Communication Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM SMART PARKING BERBASIS ARDUINO UNTUK LAHAN PARKIR BERTINGKAT.

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

LABIB IZZATUR RAHMAN
NRP. 05211540000108

Surabaya, 11 Juli 2019



Mahendrawathi Er, S.T., M.Sc., Ph.D.
NIP. 19761011 200604 2 001

Halaman ini sengaja dikosongkan

LEMBAR PERSETUJUAN

RANCANG BANGUN SISTEM SMART PARKING BERBASIS ARDUINO UNTUK LAHAN PARKIR BERTINGKAT

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

LABIB IZZATUR RAHMAN
NRP. 05211540000108

Disetujui Tim Penguji : Tanggal Ujian : 12 Juli 2019
Periode Wisuda : September 2019

Dr.Eng. Febriliyan Samopa, S.Kom, M.Kom.

(Pembimbing I)



Bekti Cahyo Hidayanto, S.Si, M.Kom

(Penguji I)

Hatma Suryotrisongko, S.Kom, M.Eng.

(Penguji II)

Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM SMART PARKING BERBASIS ARDUINO UNTUK LAHAN PARKIR BERTINGKAT

Nama : Labib Izzatur Rahman
NRP : 05211540000108
Departemen : Sistem Informasi ITS
Pembimbing I : Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

ABSTRAK

Dewasa ini, jumlah kendaraan yang berada di jalan semakin banyak. Hal ini menciptakan masalah-masalah baru seperti kurangnya lahan parkir. Kejadian yang lumrah terjadi adalah pengendara akan berputar-putar area parkir untuk mencari lot parkir yang kosong, terkadang pengendara tidak menemukan lahan parkir yang kosong, sehingga membuang waktu pengendara sia-sia. Penelitian ini mengusulkan sebuah purwarupa sistem parkir cerdas berbasis IoT yang menggabungkan sensor, web server, dan mobile application. Sistem ini akan membantu pengendara mobil dalam mencari tempat parkir kosong. Sistem ini akan dipasang di tempat parkir bertingkat. Sistem ini akan menggunakan sensor ultrasonik yang bergantung dengan gelombang suara untuk mendeteksi kondisi lot parkir, apakah lot tersebut tersedia atau tidak. Apabila lot parkir yang tersedia sama dengan jumlah mobil yang memasuki area parkir, maka sistem akan menutup gerbang sehingga mencegah kejadian pengendara tidak menemukan lahan parkir. Sistem parkir cerdas ini menggunakan aplikasi web untuk membantu pengendara mencari lot parkir. Aplikasi akan memvisualisasikan lot parkir kosong menggunakan denah berbentuk tabel. Dengan ini, harapannya secara garis besar sistem ini dapat mengurangi waktu yang terbuang untuk mencari parkir.

Kata Kunci :
Sistem Parkir Cerdas, IoT, Sensor Ultrasonik

DESIGN AND DEVELOPMENT OF ARDUINO-BASED SMART PARKING SYSTEM FOR MULTI-STOREY PARKING LOT

Name : Labib Izzatur Rahman
NRP : 05211540000108
Department : Information Systems ITS
Supervisor I : Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom.

ABSTRACT

Today, there are more and more vehicles on the road. This raises new problems such as decreasing in availability of parking space. A common occurrence is that the driver will circulate the parking area to find an empty parking lot, sometimes the driver doesn't find an empty parking lot, resulting in a waste of time. This study propose an IoT-based smart parking system prototype that utilize sensors, web servers and Arduino. This system will help motorists find empty parking lots. This system will be able to operate in a multi-storey parking lot. This system utilize ultrasonic sensors that rely on sound waves to monitor parking lot, whether the lot is available or not. If a lot of parking is available equal to the number of cars that enters parking area, the system will close the gate so as to prevent the occurrence of the driver not finding parking space. The smart parking system uses a web application to help motorists find free parking lot. The application will visualize parking lots conditions as a table. With this system, it is expected that system can reduce wasted time in looking for parking.

Keywords : Smart Parking, Internet of Things (IoT), Ultrasonic Sensor

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan serta hidayah-Nya kepada penulis sehingga penulis dapat menyelesaikan tugas akhir ini yang merupakan salah satu syarat kelulusan di Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada :

1. Sege nap keluarga besar terutama kedua orang tua dan adik penulis, Bapak Fatkhul Arifin, Ibu Umi Zayanah, Nabil Apta Maulana, Mumtaz Achmad Ramadhan, Danastri Verina Almaghfiroh, yang senantiasa mendoakan, memberikan motivasi dan semangat, sehingga penulis mampu menyelesaikan pendidikan sarjana ini dengan baik.
2. Ibu Mahendrawathi Er, S.T., M.Sc., Ph.D. selaku Kepala Departemen Sistem Informasi ITS, Bapak Nisfu Asrul Sani, S.Kom., M.Sc selaku Ketua Program Studi Sarjana Departemen Sistem Informasi ITS, serta seluruh dosen pengajar beserta staf dan karyawan Departemen Sistem Informasi ITS selama penulis menjalani perkuliahan.
3. Bu Renny Pradina K., ST, MT, SCJP sebagai dosen wali selama menempuh pendidikan di Departemen Sistem Informasi ITS.
4. Bapak Dr.Eng. Febriliyan Samopa, S.Kom., M.Kom. selaku dosen pembimbing yang telah banyak meluangkan waktu untuk membimbing, mengarahkan, dan mendukung dengan memberikan ilmu, petunjuk, dan motivasi dalam penyelesaian tugas akhir ini.

5. Bapak Bekti Cahyo Hidayanto, S.Si., M.Kom. dan Bapak Hatma Suryotrisonoko, S.Kom, M.Eng. selaku dosen penguji yang telah memberikan kritik dan saran dalam penyempurnaan tugas akhir ini.
6. Teman-teman Sistem Informasi angkatan 2015 (Lannister) yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.
7. Teman-teman kopi EX6 yang senantiasa menemani dan memberikan motivasi bagi penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.
8. Vidia Widya, Ara, Nadira Firinda, yang senantiasa mau memberikan panduan dan saran dalam mengerjakan buku tugas akhir.
9. Seluruh pihak-pihak lainnya yang tidak dapat disebutkan satu per satu, yang telah membantu penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

Penyusunan laporan tugas akhir ini masih jauh dari kata sempurna, sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, Juli 2019

Penulis

Labib Izzatur Rahman

DAFTAR ISI

LEMBAR PENGESAHAN	ERROR! BOOKMARK NOT DEFINED.
LEMBAR PERSETUJUAN	ERROR! BOOKMARK NOT DEFINED.
ABSTRAK	IX
ABSTRACT	XI
KATA PENGANTAR	XIII
DAFTAR ISI	XV
DAFTAR GAMBAR	XVII
DAFTAR TABEL	XIX
DAFTAR KODE	XXI
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan	3
1.5. Manfaat	3
1.6. Relevansi	4
BAB II TINJAUAN PUSTAKA	5
2.1. Studi Literatur	5
2.2. Dasar Teori	6
2.2.1 Arduino	6
2.2.2 Rangefinder Ultrasonik	6
2.2.3 Arduino IDE	7
2.2.4 Motor Servo	8
BAB III METODOLOGI	9
3.1 Studi Literatur	9
3.2 Analisis dan Desain	9
3.3 Perancangan Sistem	10
3.4 Implementasi Aplikasi	10
3.5 Testing Aplikasi	10
3.6 Dokumen Tugas Akhir	11
BAB IV PERANCANGAN	13
4.1 Desain <i>Use Case</i>	13
4.2 Desain Sistem	14

4.2.1	Desain Rangkaian Perangkat Keras.....	14
4.2.2	Desain Sistem	16
4.2.3	Desain Jaringan Sistem.....	17
4.2.4	Desain Basis Data.....	17
4.2.5	Desain Uji black-box.....	19
BAB V IMPLEMENTASI		21
5.1.	Lingkungan Implementasi.....	21
5.2.	Implementasi Sistem	22
5.2.1	Implementasi Rangkaian Perangkat Keras	22
5.2.2	Implementasi Kode Program Perangkat Keras Lot Parkir.....	22
5.2.3	Implementasi Kode Perangkat Keras Gerbang.....	27
5.3	Implementasi Basis Data.....	34
5.4	Implementasi Server.....	35
5.5	Implementasi Uji black-box	38
5.6	Implementasi <i>Deployment</i>	40
BAB VI HASIL DAN PEMBAHASAN.....		41
6.1	Dashboard Admin	41
6.2	Model Lot Parkir	42
6.3	Lot Parkir	43
6.4	Reservasi Lot Parkir	43
6.5	Gerbang Parkir	44
6.6	Refresh sensor	45
6.7	Login	46
BAB VII KESIMPULAN DAN SARAN		47
7.1	Kesimpulan	47
7.2	Saran.....	47
DAFTAR PUSTAKA.....		49
BIODATA PENULIS.....		51

DAFTAR GAMBAR

Gambar 2. 1. Sensor Ultrasonik HC-SR04.....	7
Gambar 2. 2. Cara kerja sensor ultrasonik.....	7
Gambar 3. 1. Bagan metodologi.....	9
Gambar 4. 1 Desain Use Case Sistem Pengguna admin.....	13
Gambar 4. 2 Use Case pengguna user.....	14
Gambar 4. 3 Desain Arduino Lot.....	14
Gambar 4. 4 Design Arduino Gerbang.....	15
Gambar 4. 5 Desain Sistem.....	16
Gambar 4. 6 Desain Jaringan Sistem.....	17
Gambar 4. 7 Desain Database.....	18
Gambar 5. 1 Implementasi Perangkat Keras.....	22
Gambar 5. 2 State Diagram Arduino Gerbang pintu masuk	30
Gambar 5. 3 State diagram Arduino gerbang pintu keluar	33
Gambar 5. 4 Daftar Tabel Sistem.....	34
Gambar 5. 5 GET request ke arduino.....	36
Gambar 6. 1. Dashboard Admin.....	41
Gambar 6. 2. Model lot parkir.....	42
Gambar 6. 3 Lot Parkir.....	43
Gambar 6. 4. Reservasi lot parkir.....	43
Gambar 6. 5. Gerbang parkir buka.....	44
Gambar 6. 6. Gerbang parkir ketika kapasitas maksimal	45
Gambar 6. 7. Tombol Refresh Reservation.....	45

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3. 1 Jadwal Kegiatan.....	11
Tabel 4. 1. Rancangan Uji black-box.....	19
Tabel 5. 1 Lingkungan Implementasi	21
Tabel 5. 2 Teknologi pengembangan	21
Tabel 5. 3. Hasil uji black-box.....	39

Halamanan ini sengaja dikosongkan

DAFTAR KODE

Kode 5. 1 Library lot	23
Kode 5. 2 Inisiasi Variabel.....	23
Kode 5. 3 Ekspos endpoint variabel.....	24
Kode 5. 4 Inisiasi endpoint fungsi	25
Kode 5. 5 Inisiasi sensor jarak.....	25
Kode 5. 6 Pengambilan Jarak.....	26
Kode 5. 7 Penentuan status lot level A.....	26
Kode 5. 8 Baris program LED.....	27
Kode 5. 9 Library Arduino Gerbang	28
Kode 5. 10 Inisiasi variabel arduino gerbang	28
Kode 5. 11 State Arduino Gerbang.....	29
Kode 5. 12 Baris kode Arduino Gerbang State VIP	31
Kode 5. 13 State Arduino Gerbang pintu Keluar	32
Kode 5. 14 State Arduino Gerbang Keluar untuk VIP	34
Kode 5. 15 GET request ke Arduino.....	35
Kode 5. 16 Pendaftaran file ke Kernel.....	36
Kode 5. 17 Set reservasi	37
Kode 5. 18 fungsi refreshReservation()	37
Kode 5. 19 fungsi Index()	38
Kode 5. 20 Fungsi vipIn() dan vipOut().....	38

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Pada bagian ini akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bagian ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir ini dapat dipahami.

1.1. Latar Belakang

Perkembangan kendaraan roda empat di Indonesia bertumbuh cukup pesat. Di Jawa Timur, rata-rata pertumbuhan kendaraan bermotor adalah 5 – 10% [1]. Dengan tahun 2017 mencapai total 138.556.669 yang termasuk sepeda motor, mobil penumpang, mobil bis, dan mobil barang. Pulau Jawa sendiri pada tahun 2015 menempati peringkat pertama dalam jumlah kendaraan di Indonesia yaitu sebesar 51,24% dari total kendaraan bermotor di Indonesia. Pertumbuhan ini menyebabkan masalah-masalah yang dapat ditemui sehari-hari seperti masalah parkir. Hal ini terutama di kota-kota besar seperti Jakarta dan Surabaya memiliki masalah parkir yang lebih kritis dikarenakan dapat menyebabkan kemacetan.

Terdapat dua macam jenis ruang parkir, yaitu parkir di badan jalan, dan parkir di luar badan jalan [2]. Parkir jenis di luar badan jalan menggunakan tempat parkir yang khusus untuk parkir kendaraan, parkir jenis ini dapat berupa open-space atau closed-space. Untuk closed-space dapat berupa lahan datar atau lahan bertingkat (multi-level). Untuk parkir, pengendara mencari tempat (lot) parkir yang kosong untuk memarkirkan mobil mereka. Persoalan yang timbul adalah pada saat pencarian lot parkir yang masih kosong, dimana pengendara akan berputar-putar atau naik-turun untuk mencari lot parkir yang kosong tersebut. Menurut survey yang dilaksanakan di kota-kota besar di Eropa, dalam setahun, penduduk Jerman

menghabiskan 44 jam dalam setahun untuk mencari parkir[3]. Masalah ini membuat penduduk Jerman menghasilkan kerugian sebesar 40.4 miliar euro. Untuk mengatasi masalah ini, dibutuhkan sebuah sistem parkir yang dapat memberitahu posisi lot parkir yang kosong, untuk mengurangi waktu yang terbuang dalam mencari lot parkir.

Internet of Things (IoT) merupakan bidang baru yang sedang naik trennya, baik di kalangan praktisi atau akademis. Meskipun istilah Internet of Things semakin sering dibicarakan, belum ada definisi pastinya[4]. Namun pada dasarnya Internet of Things memungkinkan 2 atau lebih alat yang tidak bisa berkomunikasi satu sama lain, menjadi bisa saling bertukar informasi melalui jaringan internet.

Pengaplikasian dari Internet of Things sangat luas dan beragam, seiring dengan ditemukannya solusi berbasis IoT baru seiring dengan masalah baru yang muncul[4]. Salah satu pengaplikasian IoT yang terkait penelitian ini adalah IoT dapat menghubungkan sebuah sistem berbasis sensor dengan smartphone kita sehari-hari melalui internet.

Oleh karena itu, menggunakan IoT, penulis berinisiatif melakukan penelitian mengenai pembuatan sistem Smart Parking. Sistem ini bertujuan untuk membantu pengendara dalam mencari tempat parkir agar mengurangi waktu yang dihabiskan untuk mencari parkir dalam kesehariannya, melalui smartphone.

1.2. Rumusan Masalah

Beberapa masalah yang diselesaikan pada Tugas Akhir ini adalah :

1. Bagaimana membuat sistem parkir cerdas yang dapat membantu pengendara untuk mencari lot parkir yang kosong?
2. Bagaimana memvisualisasikan informasi yang diterima sistem agar dapat membantu pengendara?

1.3. Batasan Masalah

Berdasarkan permasalahan yang telah diuraikan sebelumnya, adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Kendaraan yang diterima oleh sistem adalah kendaraan roda empat.
2. Sistem akan dibuat dengan model miniatur yang mampu merepresentasikan lahan parkir bertingkat.
3. Visualisasi informasi akan berbentuk denah tabel.
4. Rancang bangun dibangun dengan Arduino.
5. Rancang bangun tidak melingkupi sistem tiket.
6. Sistem hanya dapat diakses melalui web.
7. Sistem membutuhkan peran manusia sebagai administrator.

1.4. Tujuan

Beberapa tujuan yang ingin dicapai dari Tugas Akhir ini antara lain:

1. Membuat sistem parkir yang dapat memberitahu lot parkir yang kosong.
2. Membuat interface untuk pengendara untuk menampilkan lot parkir yang kosong.
3. Mencegah pengendara tidak menemukan lot parkir pada saat sudah di dalam area parkir.

1.5. Manfaat

Tugas Akhir ini diharapkan dapat memberikan manfaat-manfaat sebagai berikut:

1. Menjadi referensi untuk kalangan akademik dalam melakukan pengembangan penelitian dalam ruang lingkup Internet of Things
2. Menjadi referensi model untuk kalangan praktisi dalam melakukan pengembangan penelitian dalam ruang lingkup Smart Parking

1.6. Relevansi

Tugas Akhir ini disusun untuk memenuhi syarat kelulusan pada Departemen Sistem Informasi. Topik yang diangkat dalam penelitian tugas akhir ini adalah pembuatan rancang bangun sistem Smart Parking untuk lahan parkir bertingkat. Keterkaitan dengan perkuliahan yang dipelajari adalah Internet untuk Segala (IuS).

BAB II TINJAUAN PUSTAKA

Pada bagian ini akan menjelaskan mengenai studi literatur terhadap penelitian sebelumnya dan dasar teori yang dijadikan acuan atau landasan dalam pengerjaan tugas akhir ini.

2.1. Studi Literatur

Dalam proses pengerjaan dari pengembangan perangkat lunak pada tugas akhir ini, dilakukan pencarian penelitian-penelitian yang sudah dilakukan sebagai referensi dalam pengerjaan dari pengembangan perangkat lunak pada tugas akhir ini, seperti pada Tabel 2.1.

Tabel 2. 1 Studi Literatur

1) Purwarupa sistem parkir cerdas berbasis arduino sebagai upaya mewujudkan smart city.
Penulis; Tahun R.Sindhuja dan B.Krithiga; 2017 [9]
Pembahasan Penelitian ini membuat sistem smart parking berbasis Arduino di area parkir tidak bertingkat menggunakan visualisasi LED untuk membantu pengendara mencari lot parkir[5].
Keterkaitan Penelitian ini membuat sistem basis data untuk parkir lot. Basis data meliputi informasi status lot, reservasi lot, dan tipe mobil.
2) Car Parking System Using IR Sensor
Penulis; Tahun Divyansh Thakur, Yugal Kumar, Arvind Kumar, Pradeep Kumar, dan Vijendra Singh; 2018 [10]
Pembahasan Penelitian ini membuat sistem smart parking menggunakan microcontroller dibantu dengan sensor inframerah untuk

mendeteksi lot parkir yang kosong. Informasi tersebut kemudian ditampilkan ke layar LCD[6].

Keterkaitan

Penelitian tugas akhir yang dilakukan berkaitan dengan pengembangan *Real Time Monitoring* menggunakan *wireless module* untuk mengirim data dari sensor ke aplikasi yang akan ditampilkan kepada pengguna.

2.2. Dasar Teori

2.2.1 Arduino

Arduino adalah board microcontroller dengan port Universal Serial Bus (USB) untuk menghubungkan komputer dan sejumlah socket yang terhubung dengan elektronik eksternal seperti motors, relay, sensor cahaya, dioda laser, loudspeaker, microphone, dan lain-lain[7]. Alat-alat tersebut dapat diberi daya melalui USB dari komputer, dari baterai, atau dari sebuah power supply. Alat-alat ini dapat dikendalikan oleh komputer atau diprogram oleh komputer, lalu diputus sambungannya dengan komputer sehingga dapat bekerja secara independen.

Design dari board Arduino adalah open-source[7]. Artinya, siapapun boleh membuat board arduino sendiri. Hal ini membuat Arduino mempunyai harga yang rendah.

Untuk pemrograman Arduino, minimal yang dibutuhkan adalah komputer, kabel USB, dan board Arduino. IDE Arduino tersedia untuk Windows, Mac, dan Linux.

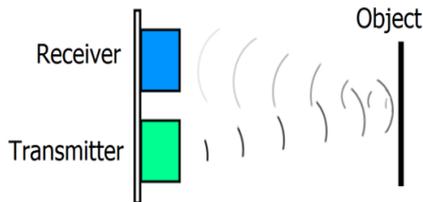
2.2.2 Rangefinder Ultrasonik

Prinsip dari Rangefinder Ultrasonik adalah untuk mengukur waktu yang ditempuh oleh sinyal yang ditembak oleh Transmitter sampai diterima oleh Receiver[8]. Kecepatan dari sinyal sudah diketahui sebelumnya, sehingga dapat mengetahui jarak objek dengan Rangefinder menggunakan rumus jarak.

Rangefinder Ultrasonik yang dipertimbangkan dalam paper ini adalah HC-SR04.



Gambar 2. 1. Sensor Ultrasonik HC-SR04



Gambar 2. 2. Cara kerja sensor ultrasonik

2.2.3 Arduino IDE

Arduino integrated development environment (IDE) adalah aplikasi cross-platform (Windows, macOS, Linux) yang dibuat dengan bahasa pemrograman Java. IDE ini digunakan untuk menulis program dan mengupload ke board Arduino.

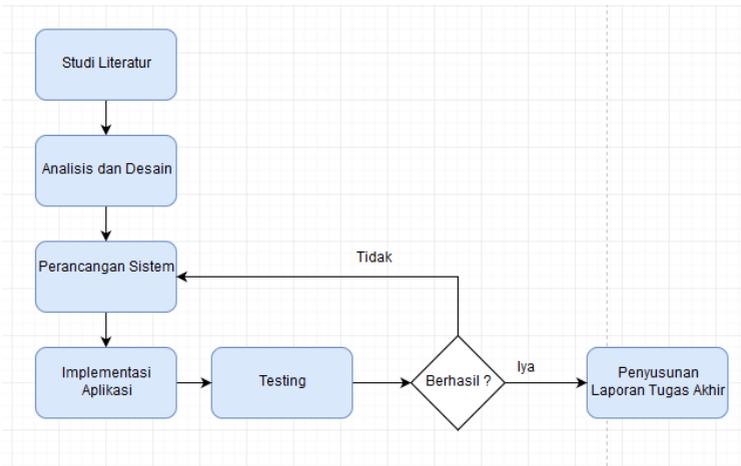
Arduino IDE mendukung bahasa C dan C++[9]. Arduino layaknya IDE yang lain menyediakan fitur library yang mana terdiri dari class-class memiliki tujuan tertentu yang dapat ditulis oleh siapa saja. Program Arduino yang ditulis oleh pengguna minimal terdiri dari 2 fungsi, fungsi setup() sebagai persiapan/inisialisasi pin-pin yang akan ditancapkan ke Arduino, dan fungsi loop() dimana fungsi loop() ini akan menjalankan perintah program di dalamnya sampai board Arduino tersebut dimatikan[10].

2.2.4 Motor Servo

Motor Servo adalah akutator putar atau aktuator linier yang mampu mengontrol posisi, kecepatan, dan percepatan sudut putar secara tepat[11]. Terdiri dari motor penggerak dan sensor untuk mengetahui posisi saat ini. Motor servo digerakkan pengontrol khusus yang dapat digerakkan dengan modul khusus untuk motor servo tersebut. Penggunaan dari motor servo antara lain : robotika, mesin numerik, dan mesin otomasi. Dalam penelitian ini, motor servo dapat digunakan untuk menggerakkan gerbang parkir cerdas.

BAB III METODOLOGI

Pada bab ini dijelaskan mengenai metodologi yang akan digunakan dalam penyelesaian tugas akhir yang akan dilakukan.



Gambar 3. 1. Bagan metodologi

3.1 Studi Literatur

Pada tahap ini peneliti melakukan pengumpulan literatur yang mendukung dalam menyelesaikan tugas akhir ini. Literatur yang diambil adalah terkait konsep-konsep atau penelitian sebelumnya yang pernah dilakukan dan didokumentasikan dalam buku, jurnal, maupun website. Keluaran dari proses ini adalah pemahaman konsep dan knowledge gap pada penelitian sebelumnya.

3.2 Analisis dan Desain

Pada tahap ini dilakukan analisis dan desain sistem yang akan dibuat, yaitu bagaimana data dapat diambil menggunakan

arduino, bagaimana data dapat dikirim dari arduino ke server, dan bagaimana server juga mengirim informasi ke arduino. Pada tahap analisis des

ain sistem ini yang harus dilakukan adalah menganalisis kebutuhan apa saja agar bisa menjalankan fungsi yang dibutuhkan, fungsi yang dibutuhkan adalah :

- Arduino dapat mengambil data dari sensor
- Server dapat mengambil data dari Arduino
- Data dari server siap diambil oleh aplikasi mobile

3.3 Perancangan Sistem

Pada tahap ini dilakukan perancangan sistem sehingga sistem parkir cerdas dapat melakukan pengambilan data dan mengirim data tersebut ke server. Hal yang akan dilakukan pada tahap ini adalah :

- Pembuatan model miniatur parkir
- Perakitan sensor, motor servo, dan Arduino
- Pemrograman Arduino sehingga dapat mengirim data sensor ke server
- Pembuatan Web Server dan database

Hasil akhir dari tahap ini adalah model sistem parkir cerdas dengan sensor-sensor yang telah terhubung ke server.

3.4 Implementasi Aplikasi

Pada tahap ini dilakukan penampilan data yang didapatkan arduino ke web melalui endpoint API yang terdapat di arduino. Data yang ditampilkan akan berupa denah area parkir dilengkapi dengan kondisi kekosongan lot parkir.

3.5 Testing Aplikasi

Pada tahap ini akan dilakukan pengujian terhadap sistem yang telah dibuat secara black box, adapun hal-hal yang perlu diuji adalah :

- Akurasi sensor ultrasonik membaca kondisi lot parkir dibandingkan dengan kondisi sebenarnya secara langsung
- Interval update sensor pembaca lot parkir
- Perbandingan data yang divisualisasikan dengan kondisi sebenarnya

Apabila terdapat hasil uji yang tidak berhasil, maka peneliti akan kembali ke tahap perancangan sistem atau implementasi aplikasi sesuai hasil uji.

3.6 Dokumen Tugas Akhir

Pada tahap ini, laporan tugas akhir akan dibuat yang akan mendokumentasikan setiap langkah yang telah dilakukan, hasil yang dihasilkan, kesimpulan serta saran untuk penelitian kedepannya. Laporan tugas akhir akan berbentuk buku tugas akhir yang disusun sesuai format yang telah ditentukan.

3.7. Jadwal Kegiatan

Tabel 3. 1 Jadwal Kegiatan

No	Kegiatan	Februari				Maret				April				Mei				Juni			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
1	Studi Literatur	■	■	■	■																
2	Analisis dan Desain					■	■	■	■												
3	Perancangan Sistem									■	■	■	■								
4	Impementasi Aplikasi													■	■	■	■				
5	Testing Aplikasi																	■	■	■	■
	Penyusunan Laporan					■	■	■	■	■	■	■	■	■	■	■	■				
6	Tugas Akhir																				

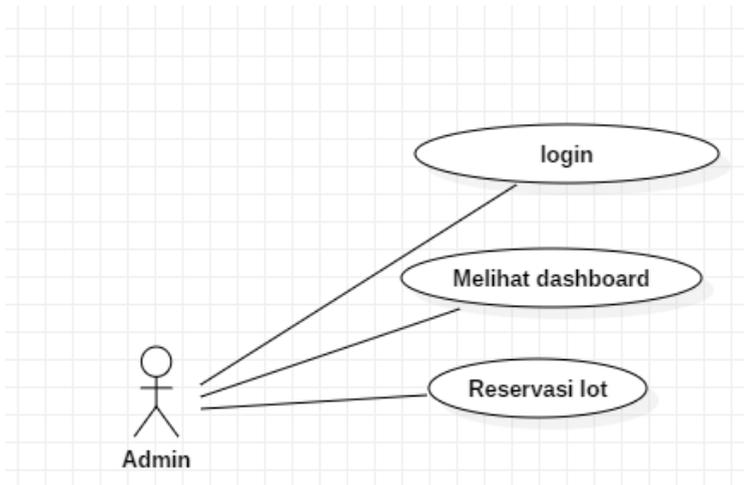
Halaman ini sengaja dikosongkan

BAB IV PERANCANGAN

Pada bagian ini akan menjelaskan mengenai proses perancangan terhadap sistem yang akan digunakan sebagai acuan dalam proses implementasi.

4.1 Desain Use Case

Pada sub bagian ini akan dijelaskan mengenai desain *use case* diagram pada sistem yang dapat digunakan oleh pengguna. Jenis pengguna ada 2, yaitu *Admin* dan *User*. *Admin* adalah pengguna yang dapat mengelola sistem, sedangkan *user* hanya bisa melihat status lot parkir. Berikut perancangan atau desain dari *use case* untuk sistem secara keseluruhan yang akan diperlihatkan pada gambar 4.1.



Gambar 4. 1 Desain Use Case Sistem Pengguna admin

Kemudian berikut adalah use case untuk pengguna



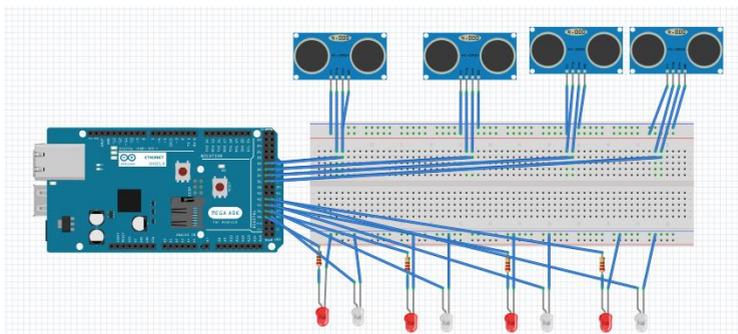
Gambar 4. 2 Use Case pengguna *user*

4.2 Desain Sistem

Pada desain sistem akan dijelaskan bagaimana gambaran dari sistem yang akan dibuat serta aliran informasi berupa data dari Arduino hingga dapat dimunculkan dalam aplikasi *client* berbasis *web*.

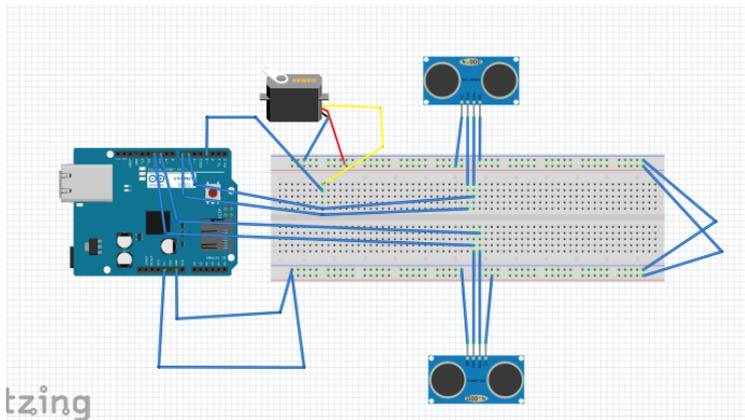
4.2.1 Desain Rangkaian Perangkat Keras

Desain Arduino akan dibagi menjadi dua, yaitu Arduino yang membaca data lot parkir menggunakan sensor ultrasonik dan memberikan respon berupa nyala LED, yang selanjutnya akan disebut sebagai Arduino Lot dan Arduino yang bertugas untuk menutup gerbang menggunakan servo apabila kapasitas lot parkir penuh, yang selanjutnya akan disebut dengan Arduino Gerbang.



Gambar 4. 3 Desain Arduino Lot

Gambar 4.3 menggambarkan desain Arduino Lot untuk empat lot parkir. Setiap kutub positif LED akan disambungkan ke resistor 220 ohm lalu dihubungkan ke pin arduino dengan kabel, sedangkan kutub negatif LED akan dihubungkan ke GND melalui breadboard. Setiap sensor ultrasonik membutuhkan 4 pin, yaitu pin GND, pin VCC, pin TRIGGER, dan pin ECHO. Pin GND dihubungkan ke GND arduino melalui breadboard. Pin VCC dihubungkan ke 5V arduino melalui breadboard juga. Sedangkan pin TRIGGER dan pin ECHO akan dihubungkan ke pin digital arduino,

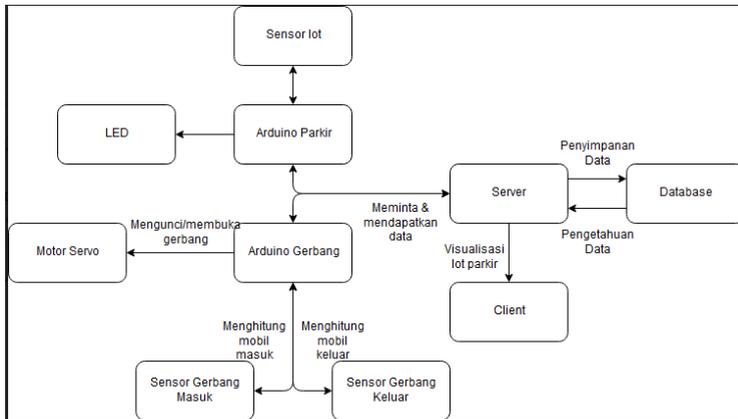


Gambar 4. 4 Design Arduino Gerbang

Gambar 4.4 menggambarkan desain Arduino Gerbang yang bertanggung jawab untuk menghitung jumlah kendaraan yang masuk dan keluar, lalu menutup gerbang apabila kapasitas parkir penuh. Arduino Gerbang terdiri dari arduino, shield ethernet, 2 sensor ultrasonik, motor servo. Untuk pemasangan sensor ultrasonik sama dengan Arduino Lot. Untuk motor servo memiliki 3 pin, yaitu pin GND, pin VCC, dan pin PULSE. Pin PULSE akan dihubungkan ke pin digital Arduino. Pin Pulse

berguna untuk mengatur rotasi dan kecepatan sudut dari motor servo.

4.2.2 Desain Sistem

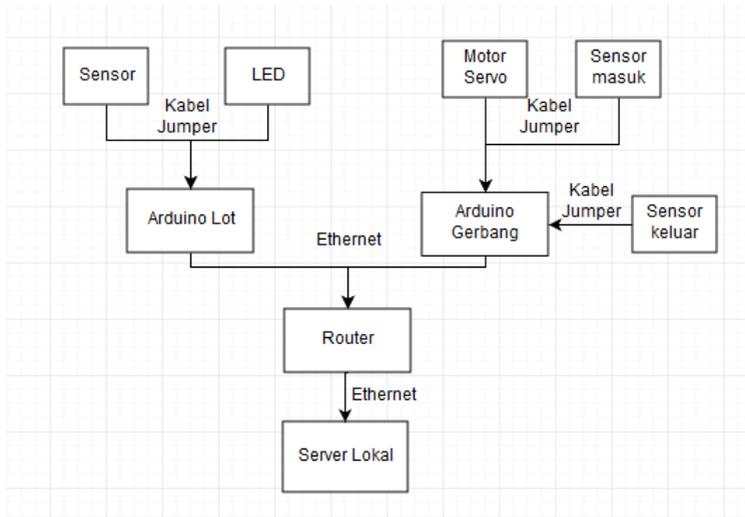


Gambar 4. 5 Desain Sistem

Gambar 4.5 menunjukkan desain sistem. Bagaimana sistem secara keseluruhan akan bekerja berikut dengan elemen-elemen di dalamnya. Arduino akan mengambil kondisi dari lot parkir menggunakan sensor ultrasonik yang dipasang di setiap lot parkir. Apabila lot parkir tersebut terisi, maka LED akan berwarna merah, apabila lot parkir tersebut tersedia maka LED akan berwarna hijau. Sistem juga akan mempertimbangkan mobil yang dapat masuk area parkir, apabila jumlah lot parkir yang tersedia dibandingkan dengan mobil yang masuk ke area parkir adalah sama, maka arduino akan memberi sinyal ke servo motor sebagai pengendali gerbang untuk mengunci gerbang masuk agar tidak terjadi over capacity. Selama itu, arduino akan mengirim data ke server terkait info lot parkir setiap interval 1 menit sekali. Selanjutnya, server akan menerima data dari arduino untuk disimpan di database, untuk persiapan agar data dapat diambil oleh aplikasi klien untuk divisualisasi.

4.2.3 Desain Jaringan Sistem

Pada sub bagian ini digambarkan desain jaringan sistem.



Gambar 4. 6 Desain Jaringan Sistem

Sensor dan Motor Servo akan terhubung dengan Arduino melalui kabel jumper, lalu terhubung dengan server lokal dengan kabel ethernet. Server lokal yang terpasang modem akan mengirimkan data melalui interface yang disediakan oleh web server, yang lalu akan disimpan ke database. Aplikasi android akan meminta data melalui API di web server.

4.2.4 Desain Basis Data

Pada sub bagian ini akan menjelaskan mengenai desain dari basis data yang akan digunakan oleh sistem. Untuk desain dari basis data dijelaskan pada Gambar 4.7

The image shows two screenshots of a database schema. The top screenshot is for the 'tugasakhir parking_lot' table, which has a primary key 'id' (int(11)) and fields 'name' (varchar(255)), 'level' (int(10)), 'status' (smallint(255) unsigned), 'reserved' (smallint(255)), 'created_at' (datetime(6)), and 'updated_at' (datetime(6)). A green arrow points to the 'id' field. The bottom screenshot is for the 'tugasakhir users' table, which has a primary key 'id' (int(11)) and fields 'username' (varchar(255)), 'created_at' (datetime(6)), and 'updated_at' (datetime(6)).

tugasakhir parking_lot	
🔑	id : int(11)
📄	name : varchar(255)
#	level : int(10)
#	status : smallint(255) unsigned
#	reserved : smallint(255)
🕒	created_at : datetime(6)
🕒	updated_at : datetime(6)

tugasakhir users	
🔑	id : int(11)
📄	username : varchar(255)
🕒	created_at : datetime(6)
🕒	updated_at : datetime(6)

Gambar 4. 7 Desain Database

Basis data sistem akan terdiri dari dua tabel, yaitu tabel parking_lot dan tabel users. Untuk perincian fungsi dari tabel adalah sebagai berikut :

1. Tabel Parking Lot digunakan untuk menyimpan informasi mengenai lot parkir, yakni seperti nama lot parkir, level (lantai), status lot parkir tersebut, dan apakah lot tersebut dibooking atau tidak.

2. Tabel users digunakan untuk menyimpan informasi user admin yang akan digunakan waktu login.

4.2.5 Desain Uji black-box

Tahap ini merupakan perancangan uji black-box terhadap sistem. Uji black-box dilakukan untuk memastikan fungsional dari sistem. Berikut adalah rancangan uji black-box.

Tabel 4. 1. Rancangan Uji black-box

No.	Test Case	Skenario	Ekspektasi
1	Kondisi Led Parkir dengan kondisi sebenarnya.	Memasukkan mobil ke lot.	LED berubah menjadi warna merah
2	Visualisasi kondisi lot parkir dengan kondisi sebenarnya	Memasukkan mobil ke lot lalu membandingkan dengan hasil visualisasi server	Status lot pada visualisasi menjadi merah (yang berarti terisi)
3	Interval antara visualisasi dengan kondisi sebenarnya	Memasukkan mobil ke lot lalu membandingkan dengan hasil visualisasi server. Hasil server tidak boleh lebih dari 30 detik.	Status lot pada visualisasi berubah menjadi merah pada kurang lebih 30 detik.
4	Memasukkan mobil ke area lot parkir	Mobil menuju gerbang, menunggu gerbang terbuka, lalu memasukkan mobil	Mobil masuk melalui gerbang. Gerbang dapat tertutup kembali dan mampu

			menerima mobil lain.
5	Memasukkan mobil ke area lot parkir saat kapasitas penuh	Mobil menuju gerbang, menunggu gerbang terbuka.	Gerbang tidak mau terbuka.
6	Reservasi lot	Admin melakukan reservasi lot ke lot yang diinginkan	Status lot menjadi "reserved", dan LED lot menjadi merah.

BAB V IMPLEMENTASI

Pada bagian ini akan menjelaskan mengenai proses implementasi terhadap sistem Smart Parking sesuai dengan *output* dari proses perancangan yang telah dibuat.

5.1. Lingkungan Implementasi

Pengembangan sistem smart parking menggunakan komputer dengan spesifikasi yang tertera pada Tabel 5.1.

Tabel 5. 1 Lingkungan Implementasi

<i>Processor</i>	Intel(R) Core (TM) i3-5005U
<i>Memory (RAM)</i>	4 GB
Sistem Operasi	Windows 10 Home 64-bit
<i>Graphic Card</i>	Intel(R) HD Graphics 5500

Rancangan sistem dikembangkan dengan bantuan teknologi seperti Arduino IDE, *framework* PHP Laravel, database, *editor*, dan *library* yang terdapat pada Tabel 5.2.

Tabel 5. 2 Teknologi pengembangan

<i>Web Server</i>	Apache
<i>Hardware</i>	Arduino Mega, Arduino Uno, Sensor ultrasonik HC-SR04, LED, motor servo.
Bahasa Pemrograman	PHP ver. 7.1, C/C++
<i>Framework</i>	Laravel 5.5
<i>Database</i>	MySQL
<i>Editor (IDE)</i>	Arduino IDE
<i>Library</i>	NewPing, SPI, aREST, Servo

5.2. Implementasi Sistem

Pengembangan sistem parkir cerdas dibagi menjadi tiga bagian, yaitu implementasi perangkat keras, implementasi server lokal, dan implementasi aplikasi.

5.2.1. Implementasi Rangkaian Perangkat Keras

Untuk dapat mengimplementasi sistem, perlu perangkaian perangkat keras. Perangkaian perangkat keras dilaksanakan dengan desain skematik sebagai cetak biru perangkaian. Perangkaian menggunakan kayu untuk membangun model yang dapat merepresentasikan lahan parkir bertingkat. Gambar 5.1 adalah tampilan depan dari model lahan parkir.



Gambar 5. 1 Implementasi Perangkat Keras

5.2.2. Implementasi Kode Program Perangkat Keras Lot Parkir

Agar arduino dan sensor dapat mengambil data lot parkir, perlu baris program yang ditulis dengan bahasa pemrograman yang dikenal.

```

1 #include <SPI.h>
2 #include <Ethernet.h>
3 #include <aREST.h>
4 #include <NewPing.h>
5 //include avr/wdt.h ???

```

Kode 5. 1 Library lot

Kode 5.1 menunjukkan library yang digunakan untuk arduino lot. Penjelasan singkat adalah sebagai berikut :

1. Ethernet : Digunakan untuk membuka server pada arduino. Hal ini berguna agar server dapat mengambil dan set data terkait ke arduino melalui endpoint yang disediakan oleh arduino.
2. aRest : Digunakan untuk membuat endpoint yang akan diakses oleh server.
3. NewPing : Digunakan untuk melakukan operasi terhadap sensor jarak ultrasonik.

```

//include avr/wdt.h ???
#define SONAR_NUM      8 // Number of sensors.
#define MAX_DISTANCE 100 // Max distance in cm.
//define PING_INTERVAL 33 // Milliseconds between pings.
#define MINIMUM_DISTANCE 7 //Jarak minimum kendaraan ke sensor

int cm_A[SONAR_NUM/2]; // Store ping distances untuk level A.
int cm_C[SONAR_NUM/2]; // Store ping distances untuk level C.
uint8_t currentSensor = 0; // Which sensor is active.
unsigned int statuslot_A[SONAR_NUM/2];
unsigned int statuslot_C[SONAR_NUM/2];
unsigned int counter_A[SONAR_NUM/2];
unsigned int counter_C[SONAR_NUM/2];
byte reservasi_A[SONAR_NUM/2];
byte reservasi_C[SONAR_NUM/2];

```

Kode 5. 2 Inisiasi Variabel

Pada Kode 5.2 dilakukan inisiasi variabel pada Arduino lot AC. SONAR_NUM adalah jumlah sensor jarak yang ditangani oleh arduino. MAX_DISTANCE merupakan jarak maksimal yang dapat diambil oleh sensor. MINIMUM_DISTANCE adalah jarak minimal dari sensor agar dianggap oleh sistem sebagai

terisi. Statuslot_A merupakan array yang menyimpan kondisi lot lantai A, apabila lot terisi akan berisi 1 bila tidak maka 0, begitu juga dengan statuslot_C. Reservasi_A adalah array yang menyimpan status reservasi dari lantai A, apabila terservasi maka 1, apabila tidak maka 0, begitu juga dengan reservasi_C. Variabel cm_A adalah array yang menyimpan jarak yang diterima oleh sensor jarak.

```
rest.variable("statuslotB1", &statuslot_B[0]);
rest.variable("statuslotB2", &statuslot_B[1]);
rest.variable("statuslotB3", &statuslot_B[2]);
rest.variable("statuslotB4", &statuslot_B[3]);

rest.variable("statuslotD1", &statuslot_D[0]);
rest.variable("statuslotD2", &statuslot_D[1]);
rest.variable("statuslotD3", &statuslot_D[2]);
rest.variable("statuslotD4", &statuslot_D[3]);

rest.set_id("002");
rest.set_name("arduino_lot_BD");
```

Kode 5.3 Ekspos endpoint variabel

Kode 5.3 menunjukkan bahwa setiap Arduino gerbang akan menyediakan endpoint-endpoint untuk diakses oleh server lokal. Endpoint “statuslotB1” akan return value 0 apabila status lot B1 sedang kosong dan return value 1 apabila status lot B1 terisi, begitu juga seterusnya untuk lot parkir yang lain.

```

rest.function("reserveC1", reserveC1);
rest.function("reserveC2", reserveC2);
rest.function("reserveC3", reserveC3);
rest.function("reserveC4", reserveC4);

rest.function("reserveA1", reserveA1);
rest.function("reserveA2", reserveA2);
rest.function("reserveA3", reserveA3);
rest.function("reserveA4", reserveA4);

```

Kode 5. 4 Inisiasi endpoint fungsi

Kode 5.4 menunjukkan endpoint yang dibuka yang digunakan untuk mengubah status reservasi pada sistem. Endpoint ini akan dipanggil oleh server. Variabel “rest” merupakan instansiasi dari class aRest yang didapatkan dari library.

```

NewPing sonar_A[SONAR_NUM/2] = { // Sensor object array.
  NewPing(TRIGGER_PIN_A1, ECHO_PIN_A1, MAX_DISTANCE),
  NewPing(TRIGGER_PIN_A2, ECHO_PIN_A2, MAX_DISTANCE),
  NewPing(TRIGGER_PIN_A3, ECHO_PIN_A3, MAX_DISTANCE),
  NewPing(TRIGGER_PIN_A4, ECHO_PIN_A4, MAX_DISTANCE)
};

NewPing sonar_C[SONAR_NUM/2] = { // Sensor object array.
  NewPing(TRIGGER_PIN_C1, ECHO_PIN_C1, MAX_DISTANCE),
  NewPing(TRIGGER_PIN_C2, ECHO_PIN_C2, MAX_DISTANCE),
  NewPing(TRIGGER_PIN_C3, ECHO_PIN_C3, MAX_DISTANCE),
  NewPing(TRIGGER_PIN_C4, ECHO_PIN_C4, MAX_DISTANCE)
};

```

Kode 5. 5 Inisiasi sensor jarak

Kode 5.5 adalah inisiasi variabel sonar_A yang merupakan instansiasi dari class NewPing yang didapatkan dari library. Variabel sonar_A adalah array yang menyimpan sensor-sensor untuk lantai A, begitu juga untuk sonar_C, menyimpan sensor-sensor untuk lantai C.

```

271 // PERULANGAN LEVEL A
272 for (uint8_t i = 0; i < SONAR_NUM/2; i++) { // Loop through each s:
273     delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms
274     cm_A[i] = sonar_A[i].ping_cm();
275     if(cm_A[i] == 0){ // work around untuk distance
276         delay(50);
277         cm_A[i] = sonar_A[i].ping_cm();
278         delay(50);
279         cm_A[i] = sonar_A[i].ping_cm();
280         delay(50);
281     }

```

Kode 5. 6 Pengambilan Jarak

Kode 5.6 merupakan baris program yang dijalankan untuk pengambilan jarak. Pada gambar 5.6 dilakukan perulangan untuk level A. Dengan bantuan library NewPing, setiap sonar_A dilakukan fungsi ping_cm() yang akan mengembalikan jarak di depan sensor dalam satuan sentimeter, yang lalu nilai jarak tersebut disimpan ke array cm_A. Pada baris kode 275 – 281 merupakan workaround yang harus dilakukan yang disebabkan oleh sensor HC-SR04 yang digunakan terkadang mengembalikan nilai 0.

```

---
290     if(cm_A[i] <= MINIMUM_DISTANCE)
291         counter_A[i]++;
292     else
293         counter_A[i] = 0;
294
295     if(counter_A[i]>=3)
296         statuslot_A[i] = 1;
297     else
298         statuslot_A[i] = 0;

```

Kode 5. 7 Penentuan status lot level A

Kode 5.7 menunjukkan bagaimana sistem menentukan bahwa lot terisi atau tidak. Pada baris 290 – 291 adalah penambahan nilai “counter” sebanyak satu apabila jarak benda di depan sensor telah di bawah nilai MINIMUM_DISTANCE yang

merupakan variabel global. Baris 295 – 298 menunjukkan bahwa apabila counter untuk lot tersebut melebihi 3, maka lot tersebut dianggap terisi, bila tidak maka lot tersebut dianggap kosong. Sistem menggunakan hitungan “counter” untuk mengurangi bacaan sensor yang salah.

```

334     for (uint8_t i=0; i < SONAR_NUM/2; i++){
335         if(statuslot_C[i] == 1 || reservasi_C[i] == 1)
336             turnOnLedRed("c",i);
337         else
338             turnOnLedGreen("c",i);
339         if(statuslot_A[i] == 1 || reservasi_A[i])
340             turnOnLedRed("a",i);
341         else
342             turnOnLedGreen("a",i);
343     }

```

Kode 5. 8 Baris program LED

Sistem menggunakan LED untuk visualisasi lot parkir. TurnOnLedRed() adalah fungsi untuk menyalakan LED merah, sekaligus mematikan LED hijau. Kebalikannya, TurnOnLedGreen() adalah fungsi untuk menyalakan LED hijau sekaligus untuk mematikan LED merah.

Baris 335 – 338 menjelaskan bagaimana sistem menyalakan LED pada lantai C. LED hanya menyala merah apabila statuslot_C bernilai 1, yang berarti lot tersebut terisi, atau reservasiC bernilai 1 yang berarti lot tersebut dipeservasi. Hal tersebut juga sama halnya untuk lot A yang dituliskan di baris kode 339 – 342.

5.2.3. Implementasi Kode Perangkat Keras Gerbang

Gerbang sistem smart parking dikendalikan juga oleh Arduino. Arduino untuk gerbang, yang selanjutnya akan dipanggil “Arduino Gerbang” membutuhkan baris program yang berbeda dengan arduino lot. Arduino gerbang berperan besar agar sistem parkir cerdas ini dapat disebut cerdas. Gerbang harus tahu kapan

dapat membolehkan mobil masuk, dan kapan tidak. Gerbang tidak boleh membiarkan mobil masuk apabila kapasitas sedang penuh.

```

2 // Libraries
3 #include <SPI.h>
4 #include <Ethernet.h>
5 #include <aREST.h>
6 #include <avr/wdt.h>
7 #include <Servo.h>
8 #include <NewPing.h>

```

Kode 5. 9 Library Arduino Gerbang

Library yang digunakan tidak jauh berbeda dengan arduino lot. Terdapat satu tambahan untuk arduino gerbang yaitu library “Servo” yang digunakan untuk mengendalikan motor servo selaku gerbang sistem smart parking.

```

10 Servo servo;
11 #define SERVO_PIN 2
12 #define BUKA 270 //Derajat untuk membuka pintu gerbang
13 #define TUTUP 90 //Derajat untuk menutup pintu gerbang
14 #define MAX_CAPACITY 16
15 #define MIN_DISTANCE 6

```

Kode 5. 10 Inisiasi variabel arduino gerbang

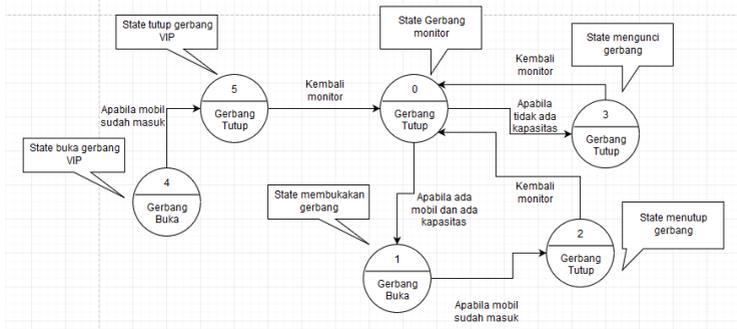
MAX_CAPACITY merupakan variabel yang menyimpan kapasitas maksimal dari sistem. MIN_DISTANCE adalah jarak minimal dari sensor untuk menganggap ada benda di depan sensor.

```
145  switch (state)
146  {
147  case 0 : //state monitor sensor gerbang masuk
148          if(counter_in>=3){
149              if(current_car < MAX_CAPACITY)
150                  state = 1;
151              else
152                  state = 3;
153          }
154          break;
155
156  case 1 : //state mbuka gerbang
157          servo.write(BUKA);
158          if(distance_in>= MIN_DISTANCE)
159              counter_in_no_car++; //counter_in_no_car ada
160          if(counter_in_no_car >= 5)
161              state = 2;
162          break;
163  case 2 : //state nutup gerbang waktu udah mbuka
164          servo.write(TUTUP);
165          current_car++;
166          counter_in_no_car = 0;
167          state=0;
168          break;
169  case 3 : //state kalo uda max_capacity
170          servo.write(TUTUP);
171          state = 0;
172          break;
173  }
```

Kode 5. 11 State Arduino Gerbang

Arduino gerbang menggunakan konsep *state* untuk membantu dalam pengoperasiannya. State 0 adalah state saat gerbang sedang memonitor apakah ada mobil di depan sensor atau tidak, apabila ada dan parkir masih mempunyai sisa lot parkir, maka diteruskan di state 1. State 1 adalah state di mana gerbang membuka pintu. Apabila sudah lebih dari 5 hitungan tidak ada mobil, maka lanjut ke state 2. State 2 adalah ketika gerbang ditutup dan menganggap bahwa mobil telah memasuki area parkir. Maka pada state ini, sistem menambah hitungan mobil di dalam sebanyak satu, yang digambarkan pada baris 165.

Apabila pada saat state 0 kapasitas penuh, maka diteruskan ke state 3 di mana gerbang tidak akan membuka kecuali area smart parking terbebas satu mobil.



Gambar 5. 2 State Diagram Arduino Gerbang pintu masuk

Gambar 5.2 menggambarkan State Machine dari gerbang pintu masuk. Angka “0” menggambarkan angka “state” dan dibawah angka state adalah output dari mesin. Apabila “Gerbang Tutup” maka outputnya adalah menutup gerbang, apabila “Gerbang Buka” outputnya adalah membuka gerbang. State 0 adalah state default dari mesin, di mana gerbang memonitor apakah ada mobil di depan atau tidak. Apabila ada, dan lot parkir masih dapat memasukkan mobil, dia akan masuk ke state 1. State 1 mesin membukakan gerbang untuk mobil. Mesin menunggu hingga mobil sudah tidak di depan sensor lagi. Cara mesin membedakan mobil dengan objek lain adalah dengan cara menarik data selama 4 detik, apabila selama 4 detik sudah tidak ada objek lain, maka mobil sudah melewati gerbang, setelah itu dilanjutkan ke state 2. State 2 adalah state ketika mesin menutup gerbang ketika mobil sudah masuk. Di state 2 dilakukan penambahan variabel “current_car” sebanyak satu, lalu dilemparkan kembali ke state 0, di mana mesin mulai memonitor lagi. Mesin hanya masuk ke state 3 apabila pada saat state 0, lot parkir tidak dapat menampung lebih banyak lagi, sehingga di state 3 yang dilakukan adalah menutup gerbang.

```
186     case 4 : //state mbuka gerbang
187             servo.write(BUKA);
188             if(distance_in>= MIN_DISTANCE)
189                 counter_in_no_car++; //counter_in_no_car adal
190             if(counter_in_no_car >= 5)
191                 state = 5;
192             break;
193     case 5 : //state nutup gerbang waktu udah mbuka (VIP)
194             servo.write(TUTUP);
195             current_vip_car++;
196             counter_in_no_car = 0;
197             state=0;
198             break;
199 }
```

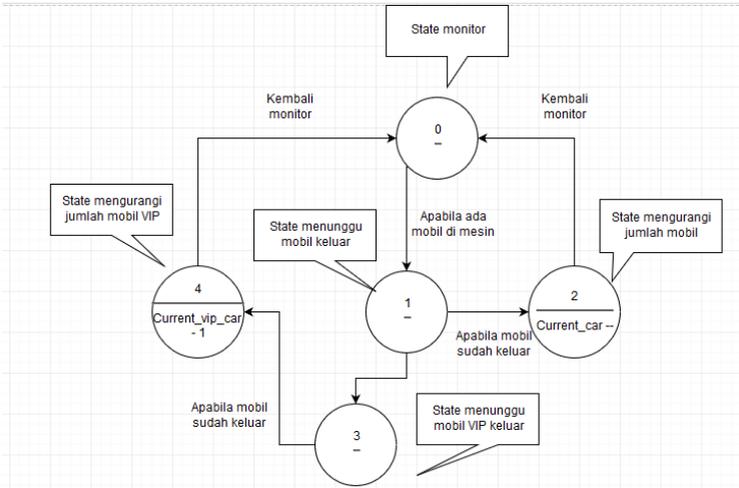
Kode 5. 12 Baris kode Arduino Gerbang State VIP

Kode 5.12 adalah baris kode untuk VIP. Apabila masuk ke switch 4, dimana itu adalah case 4, maka Arduino akan membuka gerbang, dan mulai menghitung apabila mobil sudah tidak di depan sensor. Apabila sudah lima hitungan, ditunjukkan pada baris 190, maka akan lanjut ke state 5. Pada state 5, mesin akan menutup gerbang dan menambahkan jumlah mobil VIP pada sistem, ditunjukkan pada baris 196. State 4 hanya bisa dipanggil melalui input admin.

```
184 switch(state_out){
185     case 0 : //state monitor gerbang keluar
186         if(counter_out>=3)
187             state_out=1;
188             break;
189     case 1 :
190         if(distance_out >= MIN_DISTANCE)
191             counter_out_no_car++;
192             if(counter_out_no_car >= 3)
193                 state_out = 2;
194             break;
195
196     case 2 : //state ngurangi counter mobil dalam parkir
197         current_car--;
198         counter_out_no_car = 0;
199         state_out=0;
200         break;
201
202
203
204 }
205
```

Kode 5. 13 State Arduino Gerbang pintu Keluar

Untuk gerbang keluar memiliki state sendiri seperti pada Kode 5.13. Gerbang keluar akan menunggu 3 hitungan untuk mendeteksi mobil di depan sensor yang ingin keluar, lalu menghitung 3 hitungan lagi saat mobil sudah keluar dari jarak deteksi sensor. Pada saat itu baru mobil di dalam parkir dikurang sejumlah satu.



Gambar 5. 3 State diagram Arduino gerbang pintu keluar

Gambar 5.3 menunjukkan diagram dari state gerbang pintu keluar. State gerbang pintu keluar hampir mirip dengan pintu masuk, hanya saja lebih sederhana dikarenakan tidak ada gerbang di pintu keluar. State 0 adalah state mesin memonitor gerbang. Apabila ada mobil di depan sensor, maka akan masuk ke state 1. State 0 menunggu ada objek selama 4 detik untuk menentukan bahwa yang melewati adalah mobil. State 1 adalah state ketika mesin menunggu mobil keluar meninggalkan area parkir. Apabila mobil sudah tidak ada selama 3 detik, maka akan masuk ke state 2. State 2 adalah state yang berjalan apabila mobil sudah benar-benar meninggalkan area parkir. Mesin melakukan pengurangan jumlah mobil di dalam area parkir di state 2 ini. Apabila operasi state 2 sudah selesai, maka akan kembali masuk ke state 0 di mana mesin memonitor gerbang ulang, menunggu apabila ada mobil di depan.

```

228     case 3 :
229         if(distance_out >= MIN_DISTANCE)
230             counter_out_no_car++;
231         if(counter_out_no_car >= 3)
232             state_out = 4;
233         break;
234
235     case 4 : //state ngurangi counter mobil dalam parkir
236         current_vip_car--;
237         counter_out_no_car = 0;
238         state_out=0;
239         break;
240
241 }
242

```

Kode 5. 14 State Arduino Gerbang Keluar untuk VIP

Kode 5.14 adalah baris kode untuk menghitung keluar mobil VIP. Sama seperti gerbang masuk, state ini hanya bisa dipanggil oleh input admin. State 3 akan menunggu mobil keluar dengan hitungan “counter_out_no_car”. Apabila sudah 3 hitungan, maka berganti ke state 4. State 4 akan mengurangi jumlah mobil VIP yang terdapat pada sistem sebanyak satu, dilakukan pada baris 236.

5.3 Implementasi Basis Data

Implementasi basis data menggunakan MySQL. Berikut adalah daftar tabel yang terdapat pada basis data.

tugasakhir arduino	tugasakhir parking_lot	tugasakhir users
id : int(11)	id : int(11)	id : int(11)
name : varchar(255)	name : varchar(255)	username : varchar(255)
ip_address : varchar(255)	level : int(10)	created_at : datetime(6)
	status : smallint(255) unsigned	updated_at : datetime(6)
	reserved : smallint(255)	
	created_at : datetime(6)	
	updated_at : datetime(6)	

Gambar 5. 4 Daftar Tabel Sistem

Tabel Arduino menyimpan informasi mengenai arduino agar informasi tersebut tidak *hard-coded* pada sistem. Tabel users menyimpan informasi mengenai user. Tabel parking_lot menyimpan informasi tentang lot parkir. Pada tabel parking_lot terdapat kolom “name” untuk menyimpan nama dari lot parkir, seperti A1, B3, dan lain-lain. Level untuk menyimpan lantai berapa lot parkir tersebut berada. Status adalah status dari lot parkir tersebut, apakah terisi atau tidak. Reserved adalah status reservasi dari lot parkir tersebut, apakah tereservasi atau tidak.

5.4 Implementasi Server

Server diperlukan untuk mengambil data dari arduino, set reservasi terhadap arduino, dan visualisasi data yang diambil oleh arduino. Server dibangun di lingkungan PHP dengan framework Laravel.

```

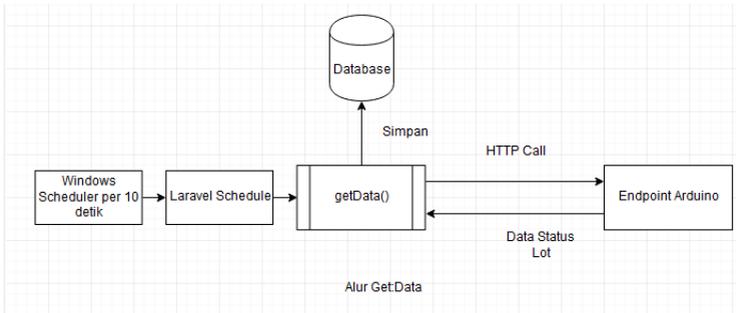
43     public function handle(){
44         $ip_address_ac = "http://192.168.100.161/";
45         $ip_address_bd = "http://192.168.100.162/";
46
47         $client = new \GuzzleHttp\Client();
48
49         $response_ac = $client->request('GET', $ip_address_ac);
50         $response_bd = $client->request('GET', $ip_address_bd); /
51
52         $request_code_ac = $response_ac->getStatusCode(); //ngam
53         $request_code_bd = $response_bd->getStatusCode();
54     }

```

Kode 5. 15 GET request ke Arduino

Dengan bantuan library Guzzle, server mengambil data ke arduino sesuai dengan alamat IP masing-masing. Data yang diambil adalah status lot dari parkir, apakah terisi atau tidak. Setelah data diambil oleh server, maka data tersebut disimpan ke database.

Kode 5.15 merupakan screenshot dari file `GetData.php` yang akan diotomasi oleh server setiap 10 detik sekali. Untuk mengotomasi dilakukan 2 hal, yaitu didaftarkan di `App\Console\Kernel.php`, lalu di automasi menggunakan `Task Scheduler` untuk Windows.



Gambar 5. 5 GET request ke arduino

Gambar 5.6 merupakan gambaran dari Gambar X. Server menggunakan `Windows Scheduler` yang sudah diatur untuk melakukan perintah `Laravel Schedule` per 10 detik. `Laravel Schedule` adalah fitur pada framework `Laravel` untuk membantu operasi yang bersifat rutin. Apabila di jalankan, salah satu yang dipanggil adalah fungsi `getData()`. Fungsi `getData()` merupakan *routine* custom yang digunakan penulis untuk mengambil data ke arduino lot.

```

protected function schedule(Schedule $schedule)
{
    // $schedule->command('inspire')
    //         ->hourly();

    $schedule->command('get:data')
        ->everyMinute()
        ->appendOutputTo('D:\Kuliah\TA\Actual Kerjaan\log.txt');
}
  
```

Kode 5. 16 Pendaftaran file ke Kernel

Task Scheduler diharuskan melakukan call “Schedule:run” ke server yang mana ketika di panggil akan melakukan semua rutinitas yang ada pada server.

```

22     public function reserve($id){
23
24         $parking_lot = ParkingLot::where('id',$id)->first();
25         // dd($parking_lot);
26         if($parking_lot->reserved == 0)
27             $parking_lot->reserved = 1;
28         else{
29             $parking_lot->reserved = 0;
30         }
31         $parking_lot->save();

```

Kode 5. 17 Set reservasi

Fungsi reserve() pada server membutuhkan satu parameter yaitu \$id. Id adalah id dari lot parkir yang akan direserve, lalu di set 1 atau 0. Pada baris 31 data yang sudah di ubah di update di database. Langkah selanjutnya agar arduino dapat mengambil data reservasi adalah dengan mengirimkan data tersebut ke arduino dengan fungsi refreshReservation().

```

36     public function refreshReservation(){
37
38         // dd($this->ip_address_gerbang);
39         $parking_lot_ac = ParkingLot::where('level','1')->orWhere('level','3')->orderBy('name','asc')->get();
40         $parking_lot_bd = ParkingLot::where('level','2')->orWhere('level','4')->orderBy('name','asc')->get();
41
42         $client = new \GuzzleHttpClient(); // init Guzzle client
43
44
45         foreach($parking_lot_ac as $item){
46             $suburl = "reserve" . strtoupper($item->name) . "?params=" . $item->reserved;
47             $url = $this->ip_address_ac . $suburl;
48             $client->get($url);
49         }

```

Kode 5. 18 fungsi refreshReservation()

Fungsi refreshReservation() berguna untuk mengupload data reservasi menuju Arduino yang diinginkan.

```

public function index(){
    $parking_lot = ParkingLot::orderBy('name','asc')->get();
    $parking_lot_A = $parking_lot->where('level','1');
    $parking_lot_B = $parking_lot->where('level','2');
    $parking_lot_C = $parking_lot->where('level','3');
    $parking_lot_D = $parking_lot->where('level','4');
    return view('index')->with(compact('parking_lot_A','parking_lot_B','parking_lot_C','parking_lot_D'));
}

```

Kode 5. 19 fungsi Index()

Selanjutnya adalah fungsi Index() yang berfungsi menampilkan dashboard beserta tabel kondisi dari lot parkir.

```

92     public function vipIn(){
93         $client = new \GuzzleHttp\Client();
94
95         $url = $this->ip_address_gerbang . 'vipIn?params=0';
96         $client->get($url);
97
98
99         return redirect(route('index'));
100     }
101 }
102
103 public function vipOut(){
104     $client = new \GuzzleHttp\Client();
105
106     $response = $client->get($this->ip_address_gerbang . 'vipOut?params=0');
107
108
109     return redirect(route('index'));
110 }

```

Kode 5. 20 Fungsi vipIn() dan vipOut()

Fungsi vipIn() dan vipOut() berfungsi agar server dapat memberi tahu Arduino bahwa vip sudah keluar atau masuk. VipIn. VipIn adalah fungsi yang akan memanggil endpoint “vipIn” pada Arduino. Fungsi ini akan dipanggil apabila admin menekan tombol VIP masuk. Sebaliknya, vipOut adalah fungsi yang akan memanggil endpoint “vipOut” pada Arduino yang akan dipanggil apabila admin menekan tombol VIP keluar jika mobil VIP keluar dari area parkir.

5.5 Implementasi Uji black-box

Metode pengujian pada sistem dilakukan dengan uji *black-box* yang telah dirancang sebelumnya. Implementasi uji black-box dilakukan dengan cara melakukan checklist terhadap setiap

poin yang terdapat pada daftar uji, dan cek apakah hasil luaran sesuai atau tidak. Tabel 5.3 menunjukkan hasil uji black-box.

Tabel 5. 3. Hasil uji black-box

No.	Test Case	Skenario	Ekspektasi	Hasil luaran sesuai
1	Kondisi Led Parkir dengan kondisi sebenarnya.	Memasukkan mobil ke lot.	LED berubah menjadi warna merah	Sesuai
2	Visualisasi kondisi lot parkir dengan kondisi sebenarnya	Memasukkan mobil ke lot lalu membandingkan dengan hasil visualisasi server	Status lot pada visualisasi menjadi merah (yang berarti terisi)	Sesuai
3	Interval antara visualisasi dengan kondisi sebenarnya	Memasukkan mobil ke lot lalu membandingkan dengan hasil visualisasi server. Hasil server tidak boleh lebih dari 30 detik.	Status lot pada visualisasi berubah menjadi merah pada kurang lebih 30 detik.	Sesuai
4	Memasukkan mobil ke area lot parkir	Mobil menuju gerbang, menunggu gerbang terbuka, lalu memasukkan mobil	Mobil masuk melalui gerbang. Gerbang dapat tertutup kembali dan mampu menerima mobil lain.	Sesuai

5	Memasukkan mobil ke area lot parkir saat kapasitas penuh	Mobil menuju gerbang, menunggu gerbang terbuka.	Gerbang tidak mau terbuka.	Sesuai
6	Reservasi lot	Admin melakukan reservasi lot ke lot yang diinginkan	Status lot menjadi "reserved", dan LED lot menjadi merah.	Sesuai

Dari hasil tabel 5.3, didapatkan bahwa hasil uji semuanya memenuhi ekspektasi.

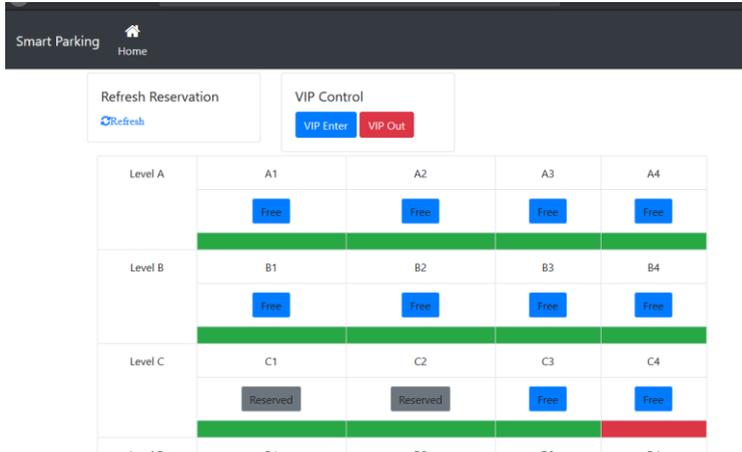
5.6 Implementasi *Deployment*

Tahap deployment merupakan tahap terakhir dari keseluruhan proses pengembangan dari sistem yang ada. Proses deployment menggunakan bantuan framework Laravel untuk memulai server. Menggunakan command "php artisan serve" oleh artisan, maka server terbuka pada alamat <http://127.0.0.1:8000/>

BAB VI HASIL DAN PEMBAHASAN

Pada bagian ini akan menjelaskan mengenai hasil dan pembahasan dari pengembangan sistem parkir cerdas yang telah dibuat dalam pengerjaan tugas akhir ini.

6.1 Dashboard Admin



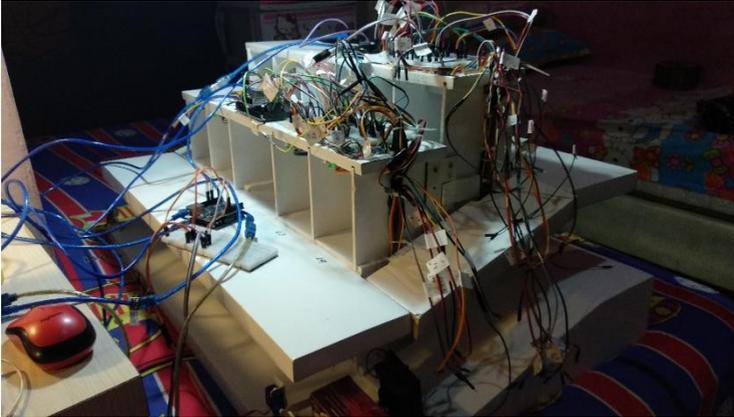
Gambar 6. 1. Dashboard Admin

Gambar 6.1 adalah tampilan Admin. Admin dapat melihat status lot parkir, apakah lot parkir tersebut terisi atau tidak. Garis hijau menandakan bahwa lot parkir tersebut sedang kosong, dan garis merah menandakan bahwa lot tersebut sedang terisi. Di atas baris warna status lot terdapat button berwarna biru dengan tulisan “free” atau button abu-abu bertuliskan “reserved”. Button tersebut akan berwarna biru apabila tidak direserve, dan abu-abu apabila direserve.

Pada Dashboard juga tersedia VIP Control, di mana ia berisi tombol yang dibutuhkan untuk mobil VIP. Setiap kali VIP masuk, admin menekan tombol “VIP Enter”. Tombol ini akan menghitung masuk mobil VIP sebesar satu. Apabila gerbang

sedang tertutup, apabila ditekan maka gerbang akan membuka. Tombol VIP Out akan ditekan oleh admin setiap kali mobil VIP keluar.

6.2 Model Lot Parkir



Gambar 6. 2. Model lot parkir

Sistem dipasangkan pada model kayu yang merepresentasikan area lot parkir. Kapasitas dari model kayu ini adalah 16 petak lot parkir. Pada model terdapat pintu masuk dan pintu keluar. Pintu masuk memiliki sensor dan gerbang, dan pintu keluar memiliki sensor. Arduino dan kabel-kabel yang dibutuhkan komponen dialihkan menuju atap dari model, di mana akan dirakit sedemikian rupa agar terkoneksi ke arduino. Setiap lot parkir terdapat satu sensor jarak, satu LED merah sebagai indikator full atau reservasi, dan satu LED hijau sebagai indikator lot bebas.

6.3 Lot Parkir



Gambar 6. 3 Lot Parkir

Gambar 6.3 menunjukkan hasil dari lot parkir. Lot di gambar 6.3 merupakan lot C4. Sensor terletak di dinding atas model. Sesuai dengan hasil uji test case, LED menyala apabila ada mobil di dalam sensor, pada gambar, mobil direpresentasikan dengan kardus teh.

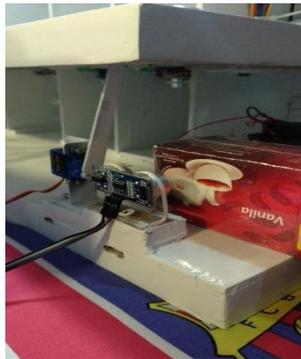
6.4 Reservasi Lot Parkir



Gambar 6. 4. Reservasi lot parkir

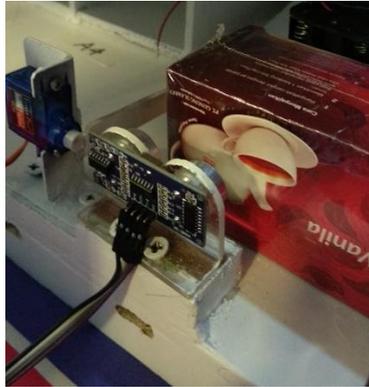
Gambar 6.4 adalah lot parkir ketika di reservasi. Pada gambar 6.4, lot C3 di reservasi oleh admin. LED akan menyala merah walaupun tidak ada mobil di dalam lot parkir, sebagai pencegah mobil lain masuk. Sesuai uji test case, LED menyala ketika di reservasi, walaupun tidak ada mobil di dalam.

6.5 Gerbang Parkir



Gambar 6. 5. Gerbang parkir buka

Gambar 6.5 merupakan gerbang dari sistem smart parking. Sesuai dengan state diagram pada bab 5 dan hasil uji black-box juga pada bab 5, gerbang akan membuka apabila mobil benar-benar di depan sensor dan apabila terdapat kapasitas tersisa untuk satu mobil. Mobil direpresentasikan pada gambar sebagai box teh.

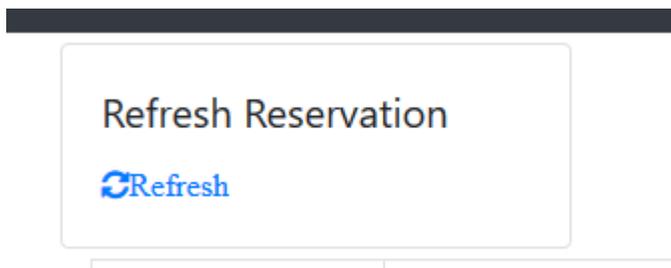


Gambar 6. 6. Gerbang parkir ketika kapasitas maksimal

Gambar 6.6 menunjukkan ketika kapasitas penuh. Apabila kapasitas parkir penuh, maka mobil tidak akan membiarkan mobil masuk lagi. Satu-satunya cara agar membuka adalah apabila ada lot tereservasi dan mobil tersebut adalah VIP sehingga dibukakan oleh admin melalui dashboard sistem.

6.6 Refresh sensor

Untuk melakukan refresh sensor perlu campur tangan dari manusia. Admin perlu menekan tombol refresh untuk mengupload ulang data reservasi ke Arduino. Pada saat ditekan, sesuai dengan implementasi server pada bab V, server akan mengirimkan data-data lot yang tereservasi ke Arduino, agar Arduino dapat mengetahui lot mana yang tereservasi.



Gambar 6. 7. Tombol Refresh Reservation

6.7 Login

Fitur login mengharuskan hanya admin saja yang dapat masuk. Fitur login dibuat sebagai pengamanan terhadap kontrol reservasi ke sistem smart parking. User biasa tidak perlu menjadi admin ataupun login untuk dapat melihat data dari status parkir.

BAB VII

KESIMPULAN DAN SARAN

Pada bagian ini akan menjelaskan mengenai kesimpulan dari pengerjaan tugas akhir ini dan saran untuk pengembangan perangkat lunak selanjutnya yang relevan.

7.1 Kesimpulan

1. Arduino mampu menjadi controller untuk menjadikan data lot parkir menjadi digital.
2. Arduino mampu menjadi controller untuk menjadikan gerbang memiliki kemampuan decision-making melalui input sensor dan output gerbang.
3. Arduino mampu berkomunikasi dengan server.
4. Penggunaan library aREST untuk Arduino sangat memudahkan untuk pengambilan data oleh server ke Arduino
5. Penggunaan sensor HC-SR04 terbukti mampu untuk dijadikan sensor untuk mendeteksi status lot parkir.
6. Meskipun terbukti mampu, perlu adanya penangan tambahan untuk sensor HC-SR04 dikarenakan tidak dapat diandalkan sepenuhnya.

7.2 Saran

Berikut merupakan saran untuk pengembangan lebih lanjut yang dapat dilakukan dalam menyempurnakan pengembangan perangkat lunak.

1. Penggantian sensor ultrasonik dengan sensor tingkat industrial untuk mempersingkat baris program.
2. Penambahan aplikasi mobile untuk memudahkan akses terhadap sistem.
3. Langsung upload data dari Arduino ke cloud.
4. Penggantian koneksi menjadi wireless, dengan memperhatikan sisi keamanan.
5. Memberi rekomendasi parkir kepada pengguna.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] P. Priyambodo, “Analisis Korelasi Jumlah Kendaraan dan Pengaruhnya Terhadap PDRB di Provinsi Jawa Timur,” *War. Penelit. Perhub.*, vol. 30, no. 1, p. 59, 2018.
- [2] T. Sipil and U. Narotama, “Analisis Kinerja Parkir Sepanjang Jalan,” vol. 1, pp. 39–46.
- [3] M. Simmons, “Germans Waste 41 Hours a Year Searching for Parking,” *INRIX*.
- [4] F. Wortmann and K. Flüchter, “Internet of Things: Technology and Value Added,” *Bus. Inf. Syst. Eng.*, vol. 57, no. 3, pp. 221–224, 2015.
- [5] C. Iswahyudi, ... A. P.-P., and undefined 2017, “Purwarupa Sistem Parkir Cerdas Berbasis Arduino Sebagai Upaya Mewujudkan Smart City,” *Jurnal.Unmuhjember.Ac.Id*, no. November, 2017.
- [6] A. Agrawal, “Car Parking System Using IR Sensors,” vol. 4, no. 2, pp. 5–8, 2017.
- [7] S. Monk, *Programming Arduino : getting started with sketches*. McGraw, 2012.
- [8] V. A. Zhmud, N. O. Kondratiev, K. A. Kuznetsov, V. G. Trubin, and L. V. Dimitrov, “Application of ultrasonic

- sensor for measuring distances in robotics,” *J. Phys. Conf. Ser.*, vol. 1015, no. 3, 2018.
- [9] J. J. (Jack J. Purdum, *Beginning C for Arduino : learn C programming for the Arduino.* .
- [10] J. R. Castro, *Building a home security system with Arduino : design, build, and maintain a home security system with Arduino Uno.* 2015.
- [11] D. Sawicz, “Hobby Servo Fundamentals,” 2012.

BIODATA PENULIS

Penulis bernama lengkap Labib Izzatur Rahman, lahir di Madiun, 26 Maret 1997.



Penulis merupakan anak pertama dari empat bersaudara. Penulis saat ini berdomisili di Surabaya, dan pada waktu buku ini ditulis sedang mengambil studi S-1 Institut Teknologi Sepuluh Nopember jurusan Sistem Informasi. Penulis sempat menginjak bangku SD di SD Negeri Kebonsari III Surabaya, SMP Negeri 1 Surabaya, SMA Negeri 2 Surabaya, dan akhirnya masuk menjadi mahasiswa

Institut Teknologi Sepuluh Nopember jurusan Sistem Informasi. Selama perkuliahan, penulis pernah menjadi staf IT-DEV untuk departemen Fakultas Teknologi Informasi dan Komunikasi (FTIK).

Guna menyelesaikan studi dan menggapai gelar Sarjana Komputer (S.Kom), penulis mengambil topik penelitian Internet untuk Segala pada Laboratorium Infrastruktur dan Keamanan Teknologi Informasi (IKTI).

Halaman ini sengaja dikosongkan