



TUGAS AKHIR - IF184802

PERANCANGAN DAN IMPLEMENTASI MINIATUR SISTEM PARKIR SEPEDA MOTOR CERDAS

AHMAD SHIDQI FIRDAUS
05111640000157

Dosen Pembimbing I
Ir. Muhammad Husni, M.Kom.

Dosen Pembimbing II
Tohari Ahmad, S.Kom., MIT., Ph.D.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2020

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - IF184802

PERANCANGAN DAN IMPLEMENTASI MINIATUR SISTEM PARKIR SEPEDA MOTOR CERDAS

AHMAD SHIDQI FIRDAUS
05111640000157

Dosen Pembimbing I
Ir. Muhammad Husni, M.Kom.

Dosen Pembimbing II
Tohari Ahmad, S.Kom., MIT., Ph.D.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2020

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - IF184802

DESIGN AND IMPLEMENTATION OF SMART MOTORCYCLE PARKING MINIATURE

AHMAD SHIDQI FIRDAUS
05111640000157

Supervisor I
Ir. Muhammad Husni, M.Kom.

Supervisor II
Tohari Ahmad, S.Kom., MIT., Ph.D.

DEPARTMENT OF INFORMATICS
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya
2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
PERANCANGAN DAN IMPLEMENTASI MINIATUR
SISTEM PARKIR SEPEDA MOTOR CERDAS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Arsitektur dan Jaringan Komputer
Program Studi S-1 Informatika
Departemen Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:
Ahmad Shidqi Firdaus
NRP: 051116 40000 157

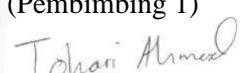
Disetujui oleh Dosen Pembimbing Tugas Akhir:

Ir. Muhammad Husni, M.Kom.
(NIP. 196002211984031001)


(Pembimbing 1)

Tohari Ahmad, S.Kom., MIT., Ph.D.
(NIP. 197505252003121002)




.....
(Pembimbing 2)

SURABAYA
JUNI 2020

[Halaman ini sengaja dikosongkan]

PERANCANGAN DAN IMPLEMENTASI MINIATUR SISTEM PARKIR SEPEDA MOTOR CERDAS

Nama Mahasiswa : Ahmad Shidqi Firdaus
NRP : 051116 40000 157
Departemen : Departemen Teknik Informatika
Fakultas Teknologi Elektro dan
Informatika Cerdas – ITS
Dosen Pembimbing 1 : Ir. Muchammad Husni, M.Kom.
Dosen Pembimbing 2 : Tohari Ahmad, S.Kom., MIT., Ph.D.
.

Abstrak

Permasalahan dalam Tugas Akhir ini diambil dari dunia nyata. Secara umum persoalan yang akan diselesaikan pada penelitian Tugas Akhir ini, dapat direpresentasikan sebagai persoalan peningkatan keamanan pada lahan parkir. Metode penyelesaian yang digunakan adalah pemanfaatan teknologi IoT.

Teknologi IoT diimplementasikan dalam bentuk miniatur, dengan dua alat yang diberi nama Transceiver Motor Cycle Device dan Smart Portal Device. Dilakukan uji coba pada implementasi kedua alat dan kedua alat tersebut telah diasumsikan berhasil memberikan pengamanan lebih terhadap motor yang sedang berada didalam parkiran.

Transceiver Motor Cycle Device dan Smart Portal Device diimplementasikan dengan menggunakan Bahasa pemrograman C yang di compile melalui Arduino IDE dan database MySQL. Dari uji coba yang telah dilakukan, didapatkan kesimpulan bahwa kedua alat telah berhasil diimplementasikan dan berfungsi dengan baik serta dapat dilakukan website monitoring, pengamanan lahan parkir menggunakan teknologi IoT telah sesuai untuk memecahkan persoalan yang ada.

Kata kunci: Teknologi IoT, MySQL, Bahasa Pemrograman C, Arduino IDE

DESIGN AND IMPLEMENTATTION OF SMART MOTORCYCLE PARKING MINIATURE

Student Name	: Ahmad Shidqi Firdaus
Registration Number	: 051116 40000 157
Department	: Department of Informatics Faculty of Intelligent Electrical and Informatics Technology – ITS
First Supervisor	: Ir. Muchammad Husni, M.Kom.
Second Supervisor	: Tohari Ahmad, S.Kom., MIT., Ph.D.

Abstract

The problems in this Final Project are taken from the real world. In general, the problems which will be solved in this Final Project research can be represented as a problem of improving security in parking lots. The settlement method used is the utilization of IoT technology.

IoT technology is implemented in a miniature form, using two devices namely Transceiver Motor Cycle Device and Smart Portal Device. Trials are carried out on the implementation of the two tools and both of the tools have been assumed to have succeeded in providing more security to the motorbikes in the parking lot.

Motor Cycle Device and Smart Portal Device Transceivers are implemented using the C programming language compiled via the Arduino IDE and MySQL database. From the trials that have been carried out, it is concluded that the two tools have been successfully implemented and are functioning properly and website monitoring can be done, securing parking lots using IoT technology is appropriate to solve existing problems.

Keywords: *IoT Technology, MySQL, C Programming Language, Arduino IDE*

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

PERANCANGAN DAN IMPLEMENTASI MINIATUR SISTEM PARKING SEPEDA MOTOR CERDAS

Pengerjaan Tugas Akhir ini dilakukan untuk memenuhi salah satu syarat meraih gelar Sarjana di Departemen Informatika Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini diharapkan apa yang telah dikerjakan penulis dapat memberikan manfaat bagi perkembangan ilmu pengetahuan terutama di bidang teknologi informasi serta bagi orang lain dan diri penulis sendiri.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama penulis mengerjakan Tugas Akhir maupun selama menempuh masa studi antara lain:

1. Terimakasih kepada Allah SWT, di mana penulis masih diberi kesempatan, kesehatan dan umur untuk menempuh kuliah disini dan menjalani hidup dengan baik sehingga mampu menyelesaikan Tugas Akhir ini.
2. Ibu Nani Mustikasari, Bapak R. Triyono Budi Prayitno, Ahmad Lutfhi Firdaus dan Ahmad Iqbal Firdaus selaku keluarga penulis yang selalu memberikan perhatian, do'a, dorongan, dan juga kasih sayang agar lebih semangat menempuh kuliah dan segera menyelesaikan Tugas Akhir ini.
3. Bapak Ir. Muchammad Husni, M.Kom. selaku Dosen Pembimbing yang telah membimbing penulis selama masa kuliah maupun selama penyelesaian Tugas Akhir ini.
4. Bapak Tohari Ahmad, S.Kom., MIT., Ph.D. Selaku Dosen Pembimbing yang telah memberikan ilmu, dan masukan kepada penulis.

5. Asyiah Udhiyah Hajar, yang selalu memberikan doa maupun usaha dalam membantu penulis serta menyemangati penulis untuk menyelesaikan Tugas Akhir ini.
6. Ivanda Zevi Amalia, selaku teman yang selalu membantu penulis dalam penyusunan buku Tugas Akhir ini.
7. Anugrah Arief Yahya Lubis, selaku adik tingkat yang membantu penulis dalam pembuatan poster, video, dan ppt.
8. Keluarga besar Teater Tiyang Alit ITS selaku keluarga nomor dua yang selalu ikut serta membantu dan menjadi penyemangat selama kuliah disini.
9. Teman-teman angkatan 2016 jurusan Teknik Informatika ITS yang telah meneman perjuangan penulis selama masa perkuliahan.
10. Serta pihak-pihak lain yang tidak dapat disebutkan di sini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis mohon maaf apabila masih ada kekurangan pada Tugas Akhir ini. Penulis juga mengharapkan kritik dan saran yang membangun untuk pembelajaran dan perbaikan di kemudian hari. Semoga melalui Tugas Akhir ini penulis dapat memberikan kontribusi dan manfaat yang sebaik-baiknya.

Surabaya, Juni 2020

Ahmad Shidqi Firdaus

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
<i>Abstrak.....</i>	<i>vii</i>
<i>Abstract.....</i>	<i>viii</i>
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xix
1 BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Permasalahan	2
1.3 Batasan Permasalahan	2
1.4 Tujuan Pembuatan Tugas Akhir	2
1.5 Manfaat Tugas Akhir	2
1.6 Metodologi.....	2
1.6.1 Penyusunan Proposal Tugas Akhir	3
1.6.2 Studi Literatur.....	3
1.6.3 Implementasi Perangkat Lunak.....	3
1.6.4 Pengujian dan Evaluasi.....	3
1.6.5 Penyusunan Buku Tugas Akhir	4
1.7 Sistematika Penulisan.....	4
2 BAB II TINJAUAN PUSTAKA	7
2.1 <i>Hardware Tools</i>	7
2.1.1 WeMos D1 R1	7
2.1.2 RFID Module RC522	7
2.1.3 Motor Servo SG90.....	9
2.1.4 Sensor Load Cell	9
2.1.5 Modul HX711.....	10
2.1.6 <i>Buzzer</i>	11
2.1.7 <i>Green LED</i>	11
2.1.8 <i>Jumper Cable</i>	12
2.2 <i>Software Tools</i>	12
2.2.1 Arduino IDE	12

2.2.2 XAMPP	14
2.2.3 CodeIgniter 3	14
2.2.4 Sublime Text 3	14
2.2.5 Navicat Premium 12	14
3 BAB III PERANCANGAN	16
3.1 Perancangan Sistem	16
3.2 Perancangan <i>Database</i>	22
3.3 Perancangan Alat	24
3.3.1 <i>Transceiver Motorcyle Device</i>	24
3.3.2 <i>Smart Portal Device</i>	25
3.3.3 Miniatur	27
3.4 Perancangan Sistem Monitor dan Kontrol.....	27
4 BAB IV IMPLEMENTASI.....	30
4.1 Implementasi Database	30
4.2 Implementasi Perangkat	33
4.2.1. <i>Transceiver Motorcyle Device</i>	33
4.2.2. <i>Smart Portal Device</i>	39
4.2.3. Miniatur	57
4.3.1. CodeIgniter	59
4.3.2. Melihat Sepeda Motor Masuk dan Keluar	60
4.3.3. Mengelola Sepeda Motor	61
4.3.4. Mengelola Pengendara	63
5 BAB V UJICOBA DAN EVALUASI	66
5.1 Lingkup Uji Coba.....	66
5.2 Sistem Parkir Sepeda Motor Cerdas	66
5.3 Performa.....	76
6 BAB VI KESIMPULAN	80
6.1 Kesimpulan	80
6.2 Saran	80
DAFTAR PUSTAKA	83
LAMPIRAN.....	87
1. Kode Program SQL	87
2. Kode Program Transceiver Motorcyle Device	89

3.	Kode Program <i>Smart Portal Device</i> Masuk	101
4.	Kode Program <i>Smart Portal Device</i> Keluar	110
5.	Kode Program <i>Controller Motor</i>	121
6.	Kode Program <i>View Index Motor</i>	125
7.	Kode Program <i>View Input Motor</i>	131
8.	Kode Program <i>Controller Pengendara</i>	134
9.	Kode Program <i>View Index Pengendara</i>	138
10.	Kode Program <i>View Input Pengendara</i>	143
11.	Kode Program <i>Model SQL</i>	145
BIODATA PENULIS		149

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Wemos D1 R1	7
Gambar 2.2 RFID Module RC522	8
Gambar 2.3 Motor Servo SG90.....	9
Gambar 2.4 Sensor Load Cell	10
Gambar 2.5 Modul HX711.....	10
Gambar 2.6 <i>Buzzer</i>	11
Gambar 2.7 <i>Green LED</i>	11
Gambar 2.8 <i>Jumper Cable</i>	12
Gambar 3.1 Alur Interaksi Device.....	16
Gambar 3.2 <i>Flowchart</i> Sistem Parkir Cerdas	18
Gambar 3.3 Alur Parkir Cerdas Masuk – 1	19
Gambar 3.4 Alur Parkir Cerdas Masuk – 2	20
Gambar 3.5 Alur Parkir Cerdas Masuk – 3	21
Gambar 3.6 <i>Smart Portal Device</i>	26
Gambar 3.7 Rancang Miniatur Sistem Parkir Cerdas	27
Gambar 3.8 Use Case Diagram Sistem Monitor.....	28
Gambar 4.1 Kode SQL SMP_MOTOR.....	30
Gambar 4.2 Kode SQL SMP_PENGENDARA	31
Gambar 4.3 Kode SQL SMP_ ACCESS	31
Gambar 4.4 Kode SQL SMP_ PORTAL.....	32
Gambar 4.5 Kode SQL Foreign Key	32
Gambar 4.6 Hasil Eksekusi SQL.....	33
Gambar 4.7 Rangkaian Arduino TMD	34
Gambar 4.8 TMD Define Pin	35
Gambar 4.9 TMD Konfigurasi	36
Gambar 4.10 TMD Setup	36
Gambar 4.11 TMD Loop.....	38
Gambar 4.12 Rangkaian Arduino RFC522.....	40
Gambar 4.13 Rangkaian Arduino SG90.....	41
Gambar 4.14 Rangkaian Arduino Load HX711	42
Gambar 4.15 Rangkaian Arduino Buzzer LED	43
Gambar 4.16 Rangkaian Arduino	44

Gambar 4.17 SPD Masuk Library	45
Gambar 4.18 SPD Masuk Define PIN	45
Gambar 4.19 SPD Masuk Konfigurasi	46
Gambar 4.20 SPD Masuk Setup	47
Gambar 4.21 SPD Masuk Fungsi ConnectPin	47
Gambar 4.22 SPD Masuk Fungsi ConnectWifi	48
Gambar 4.23 SPD Masuk Fungsi Loop	49
Gambar 4.24 SPD Masuk Fungsi Loop-1	50
Gambar 4.25 SPD Masuk Fungsi Loop-2	51
Gambar 4.26 SPD Masuk Fungsi Loop-3	52
Gambar 4.27 SPD Keluar Fungsi Loop-1	55
Gambar 4.28 Implementasi Miniatur-1	57
Gambar 4.29 Implementasi Miniatur-2	57
Gambar 4.30 Implementasi Miniatur-3	58
Gambar 4.31 Implementasi Miniatur-4	58
Gambar 4.32 Implementasi CodeIgniter SQL	59
Gambar 4.33 SMP Controller	61
Gambar 4.34 Motor Controller	61
Gambar 4.35 Motor Controller-2	62
Gambar 4.36 Motor Controller-3	62
Gambar 4.37 Motor Controller-4	62
Gambar 4.38 Pengendara Controller	63
Gambar 4.39 Pengendara Controller-2	63
Gambar 4.40 Pengendara Controller-3	64
Gambar 4.41 Pengendara Controller-4	64
Gambar 5.1 Registrasi Pengendara	67
Gambar 5.2 Daftar Pengendara	67
Gambar 5.3 Edit Pengendara	68
Gambar 5.4 Registrasi Motor	68
Gambar 5.5 Daftar Motor	69
Gambar 5.6 Edit Motor	69
Gambar 5.7 Asosiasi Masuk TMD	70
Gambar 5.8 Asosiasi Masuk TMD IDE	70
Gambar 5.9 Asosiasi Masuk TMD Monitor	70

Gambar 5.10 SMD Masuk Tap	71
Gambar 5.11 SMD Masuk <i>Database</i>	71
Gambar 5.12 SMD Masuk Monitor.....	72
Gambar 5.13 Asosiasi Keluar TMD	73
Gambar 5.14 Asosiasi Keluar TMD IDE.....	73
Gambar 5.15 Asosiasi Keluar TMD Monitor	73
Gambar 5.16 SMD Keluar Tap	74
Gambar 5.17 SMD Keluar <i>Database</i>	74
Gambar 5.18 SMD Keluar Monitor.....	75
Gambar 5.19 SMP_ACESS Finished	75
Gambar 5.20 SMP_PORTAL Finished	75

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 <i>Library</i> Arduino	13
Tabel 3.1 Alat Sistem Parkir Cerdas.....	17
Tabel 3.2 Deskripsi Tabel <i>Database</i>	22
Tabel 3.3 Detail Tabel SMP_MOTOR	22
Tabel 3.4 Detail Tabel SMP_PENGENDARA	23
Tabel 3.5 Detail Tabel SMP_ACCESS.....	23
Tabel 3.6 Detail Tabel SMP_PORTAL	24
Tabel 4.1 Rangkaian Arduino TMD	34
Tabel 4.2 TMD Loop	39
Tabel 4.3 Rangkaian Arduino RFC522.....	41
Tabel 4.4 Rangkaian Arduino SG90.....	42
Tabel 4.5 Rangkaian Arduino Load HX711	43
Tabel 4.6 Rangkaian Arduino Buzzer LED	44
Tabel 4.7 SPD Masuk Fungsi Loop-1.....	52
Tabel 4.8 SPD Masuk Fungsi Loop-2.....	53
Tabel 4.9 SPD Keluar Fungsi Loop-1.....	56
Tabel 5.1 Lingkup Spesifikasi	66
Tabel 5.2 Uji Coba TMD – 4 Meter	76
Tabel 5.3 Uji Coba TMD – 4 Meter Tap Respon	77
Tabel 5.4 Uji Coba TMD – 15 Meter	77
Tabel 5.5 Uji Coba TMD – 15 Meter Tap Respon	78

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan masalah, batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan buku Tugas Akhir ini.

1.1 Latar Belakang

Keamanan lahan parkir adalah aspek penting dalam menghindari tindak kriminal di saat pengendara sedang tidak berada di kendaraanya. terutama bagi pengguna sepeda motor, karena banyaknya tingkat kriminalitas terhadap pencurian sepeda motor yang ada, hal ini dapat membantu agar sepeda motor tetap aman tanpa khawatir adanya kriminalitas di dalam lahan parkir sepeda motor, akan tetapi terdapat masalah yaitu keamanan parkir tidak bisa mengetahui identitas pengendara dan sepeda motor yang keluar masuk lahan parkir tersebut, dikarenakan keamanan parkiran hanya sebatas membuka dan menutup portal atau pagar parkiran tersebut. Hal ini menjadi permasalahan utama yang ada dari Sistem Keamanan Parkir.

Perancangan Miniatur Sistem Parkir Sepeda Motor Cerdas ini adalah sebuah miniatur yang menggunakan beberapa perangkat keras tambahan yang disematkan pada sepeda motor dengan memadukan konsep *Internet of Things* (IoT), berfungsi sebagai identitas dari kendaraan tersebut, dengan sistem buka tutup portal menggunakan *double tap* sistem, dimana pengendara akan melakukan *tap* pada sepeda motor sebelum *tap* ke portal, sebagai syarat untuk memasuki lahan parkir, *model* miniatur ini akan dibuat dalam bentuk maket dengan menggunakan *WeMos D1*, *Servo Motor*, dan *RC522* sebagai *hardware* tambahan yang nantinya akan digunakan dalam miniatur. Yang disebut sebagai *TMD (Transceiver Motorcyle Device)* dan *SPD (Smart Portal Device)*.

Dengan adanya sistem tersebut, diharap nantinya membantu sistem keamanan lahan parkir sepeda motor dalam men-antisipasi tindak kriminal yang ada, dengan adanya informasi yang valid berupa identitas dari pengendara dan sepeda motor yang keluar masuk parkiran, yang nantinya akan tercatat secara rinci dan mampu di lakukan monitoring via website.

1.2 Rumusan Permasalahan

Rumusan masalah yang terdapat pada tugas akhir ini adalah sebagai berikut:

- a) Bagaimana merancang Miniatur Sistem Parkir Sepeda Motor Cerdas?
- b) Bagaimana mendapatkan informasi mengenai identitas pengendara dan sepeda motor yang masuk ke dalam halaman parkir?

1.3 Batasan Permasalahan

Permasalahan pada Tugas Akhir ini, memiliki batasan masalah sebagai berikut:

- a) Diasumsikan *Transceiver Motorcycle Device* pada sepeda motor sudah teridentifikasi.

1.4 Tujuan Pembuatan Tugas Akhir

Tujuan dari pembuatan Tugas Akhir ini adalah untuk merancang Sistem Parkir Sepeda Motor Cerdas.

1.5 Manfaat Tugas Akhir

Manfaat dari Tugas Akhir ini diharapkan dapat membantu Sistem Keamanan Parkir Sepeda Motor dengan adanya data mengenai identitas pengendara serta data mengenai sepeda motor yang masuk dan keluar parkiran.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal Tugas Akhir ini merupakan pendahuluan dalam menyelesaikan Tugas Akhir. Proposal ini selain berisi ringkasan tugas akhir juga meliputi latar belakang dari masalah, rumusan masalah yang akan diselesaikan, batasan masalah, tujuan dan manfaat dalam tugas akhir yang dibuat, tinjauan pustaka dalam mengerjakan tugas akhir serta metodologi yang berisi tahapan-tahapan dalam menyusun tugas akhir.

1.6.2 Studi Literatur

Studi literatur yang dipakai adalah dengan pengumpulan informasi mengenai bahan-bahan yang digunakan dalam pembuatan miniatur yang dijadikan sebagai referensi dalam penggerjaan Tugas Akhir. Mengumpulkan informasi mengenai IoT(*Internet of Things*), sensor, *servo motor*, *wemos* dan Arduino. Informasi didapatkan dari buku, paper, jurnal dan materi-materi kuliah yang berhubungan dengan topik Tugas Akhir.

1.6.3 Implementasi Perangkat Lunak

Implementasi yang akan dilakukan yaitu miniatur perancangan sistem berdasarkan studi literatur dan pengumpulan informasi yang telah dilakukan. Tahap ini suatu bentuk awal perancangan yang akan diimplementasikan. Yang dibuat adalah merupakan miniatur versi awal perancangan sistem parkir sepeda motor cerdas.

1.6.4 Pengujian dan Evaluasi

Pengujian dilakukan melalui dua cara yaitu:

1. Pengujian *hardware*

Pengujian *hardware* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat keras. Pengujian ini dilakukan untuk menguji apakah sistem keamanan kendaraan berjalan atau tidak.

2. Pengujian *software*

Pengujian *software* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak. Pengujian ini dilakukan untuk menguji apakah perangkat lunak sistem parkir sepeda motor cerdas berjalan atau tidak

1.6.5 Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini merupakan laporan secara lengkap mengenai Tugas Akhir yang telah dikerjakan baik dari sisi teori, rancangan, maupun implementasi sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkannya lebih lanjut. Sistematika penulisan buku Tugas Akhir secara garis besar antara lain:

Bab I Pendahuluan

Bab ini berisi penjelasan latar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan Tugas Akhir. Selain itu, metodologi penggerjaan dan sistematika penulisan laporan Tugas Akhir juga dijelaskan di dalamnya.

Bab II Tinjauan Pustaka

Pada bagian ini berisi kajian teori atau penjelasan dari metode, algoritma, *library*, dan *tools* yang digunakan dalam penyusunan Tugas Akhir ini.

Bab III Perancangan Sistem

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun.

Bab IV Implementasi

Pada bagian ini menjelaskan implementasi yang berbentuk kode program dari proses modifikasi, proses pengujian, serta hardware dan maket.

Bab V Pengujian dan Evaluasi

Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menjelaskan kesimpulan dari hasil ujicoba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

[*Halaman ini sengaja dikosongkan*]

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan mengenai teori-teori dasar yang berkaitan dengan pengimplementasian perangkat lunak dan perangkat keras serta penunjangnya. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap *Internet of Things*, tools, serta definisi yang digunakan dalam pembuatan Tugas Akhir.

2.1 *Hardware Tools*

2.1.1 WeMos D1 R1

WeMos D1 merupakan module development board yang berbasis WiFi dari ESP8266 yang dimana dapat diprogram menggunakan *software* IDE Arduino, perangkat inilah yang nantinya akan menghubungkan perangkat dengan *internet* melalui WiFi.[17]



Gambar 2.1 Wemos D1 R1.

2.1.2 RFID Module RC522

RFID RC522 (Radio Frequency Identification) merupakan suatu teknologi yang memanfaatkan frekuensi

radio sebagai pengidentifikasi terhadap suatu objek. *RFID* mempunyai 2 bagian komponen utama, yaitu: [14]

a) *RFID Tag*

Merupakan sebuah perangkat yang akan diidentifikasi oleh *RFID reader* yang dapat berupa perangkat pasif maupun aktif yang berisi suatu data atau informasi. Perangkat pasif tidak menggunakan catudaya, sedangkan perangkat aktif wajib menggunakan catudaya. Pada *RFID Tag* terdapat 2 jenis yaitu *Read-Write* dan *Only Read*. *RFID Tag* mempunyai 2 komponen utama, antara lain:

1. IC (Integrated Circuit) : berfungsi sebagai pemroses informasi, modulasi serta demodulasi sinyal RF, yang beroperasi dengan catudaya DC.
2. ANTENNA : mempunyai fungsi untuk mengirim maupun menerima sinyal RF.

b) *RFID Reader*

Berfungsi untuk membaca data dari *RFID Tag*. *RFID Reader* dapat berupa pasif yaitu hanya bisa membaca data dari *RFID Tag* aktif, atau aktif yang hanya dapat membaca data *RFID Tag* pasif.



Gambar 2.2 RFID Module RC522.

2.1.3 Motor Servo SG90

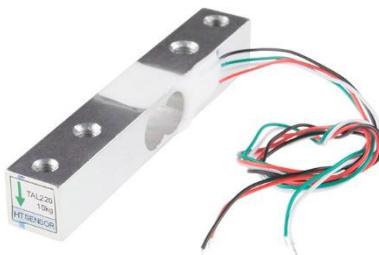
Motor Servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer. [13]



Gambar 2.3 Motor Servo SG90.

2.1.4 Sensor Load Cell

Load cell merupakan sensor transduser berat dapat merubah tekanan (force) menjadi sinyal listrik. Pada sistem timbangan digital ini digunakan strain gauge load cell. Keluaran dari sensor ini berupa tegangan dalam orde mV yang terbaca terhadap perubahan nilai resistansi yang merepresentasikan berat benda. [10]



Gambar 2.4 Sensor Load Cell.

2.1.5 Modul HX711

Modul HX711 adalah modul yang memudahkan membaca *load cell* dalam pengukuran berat. Modul ini berfungsi untuk menguatkan sinyal keluaran dari sensor dan mengonversi data analog menjadi data digital. Dengan menghubungkannya ke mikrokontroler, kita dapat membaca perubahan resistansi dari *load cell*. Setelah proses kalibrasi kita akan memperoleh pengukuran berat dengan keakuratan yang tinggi.



Gambar 2.5 Modul HX711.

2.1.6 *Buzzer*

Buzzer adalah pengubah sinyal listrik menjadi getaran suara yang berfungsi untuk menimbulkan suara beep yang digunakan penulis sebagai penanda pada alat.



Gambar 2.6 *Buzzer*.

2.1.7 *Green LED*

Green LED adalah lampu yang dipasangkan pada Arduino yang mengubah sinyal listrik menjadi cahaya.



Gambar 2.7 *Green LED*.

2.1.8 *Jumper Cable*

Kabel jumper adalah kabel yang digunakan untuk menghubungkan Arduino dengan sensor atau modul lainnya, terdapat tiga jenis kabel jumper, *Female to Male*, *Female to Female*, *Male to Male*, dimana masing masing memiliki fungsi yang sama hanya berbeda pada ujung kabelnya saja.



Gambar 2.8 *Jumper Cable*.

2.2 *Software Tools*

2.2.1 *Arduino IDE*

Arduino IDE merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran, IC mikrokontroler Arduino telah ditanamkan suatu program bernama *Bootlader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. [16]

2.2.1.1 Library

Dalam pembuatan Tugas Akhir ini digunakan beberapa *Library* untuk mendukung berjalannya pemograman pada Arduino, beberapa *Library* yang digunakan adalah:

Tabel 2.1 *Library* Arduino

Nama Library	Deskripsi
MFRC522.h	<i>Library</i> yang digunakan untuk membaca dan menulis berbagai jenis kartu yang memiliki <i>Radio-Frequency IDentification</i> (RFID) pada Arduino dengan menggunakan pembaca berbasis RC522 yang terhubung melalui antarmuka <i>Serial Peripheral Interface</i> (SPI).
Servo.h	<i>Library</i> yang memungkinkan Arduino untuk menggerakan <i>Servo Motor</i> .
MySQL_Connection.h MySQL_Cursor.h	<i>Library</i> yang memungkinan Arduino untuk berinteraksi secara langsung dengan <i>database</i> menggunakan <i>internet</i> .
ESP8266WiFi.h	<i>Library</i> yang digunakan untuk menghubungkan ESP8266 pada Arduino dengan <i>internet</i> .
ESP8266HTTPClient.h WiFiClient.h	<i>Library</i> yang memungkinan untuk ESP8266 pada Arduino mengakses sebuah <i>Hypertext Transfer Protocol</i> (HTTP).
HX711.h	<i>Library</i> yang digunakan oleh Arduino untuk membaca modul HX711.

2.2.2 XAMPP

XAMPP adalah sebuah paket perangkat lunak (software) komputer yang sistem penamaannya diambil dari akronim kata Apache, MySQL (dulu) / MariaDB (sekarang), PHP, dan Perl. Sementara imbuhan huruf “X” yang terdapat pada awal kata berasal dari istilah *cross platform* sebagai simbol bahwa aplikasi ini bisa dijalankan di empat sistem operasi berbeda, seperti OS Linux, OS Windows, Mac OS, dan juga Solaris. XAMPP pada Tugas Akhir ini akan digunakan sebagai *WebServer* serta penggunaan *database* MySQL yang telah tersedia pada XAMPP. [9]

2.2.3 CodeIgniter 3

CodeIgniter 3 adalah sebuah *framework* berbasis PHP yang digunakan pada pembuatan *website* untuk melakukan monitor pada sistem parkir sepeda motor cerdas. [15]

2.2.4 Sublime Text 3

Sublime Text 3 adalah sebuah editor yang digunakan penulis dalam melakukan pemrograman untuk membangun *website* monitor sistem parkir sepeda motor cerdas. [6]

2.2.5 Navicat Premium 12

Sebuah *database client* yang digunakan penulis untuk mempermudah dalam berinteraksi dengan *database*. [7]

[*Halaman ini sengaja dikosongkan*]

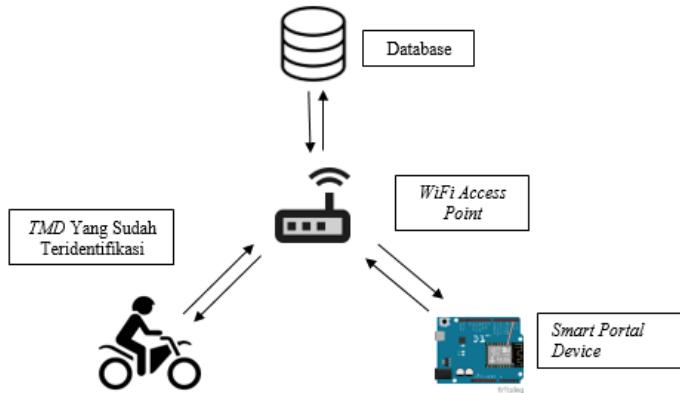
BAB III

PERANCANGAN

Pada bagian ini akan dijelaskan rancangan *hardware* serta sistem yang akan dibangun oleh penulis pada Tugas Akhir ini.

3.1 Perancangan Sistem

Penelitian ini akan membuat Miniatur Sistem Parkir Sepeda Motor Cerdas, dimana arsitektur jaringan seperti pada Gambar 3.1 Alur Interaksi *Device*.



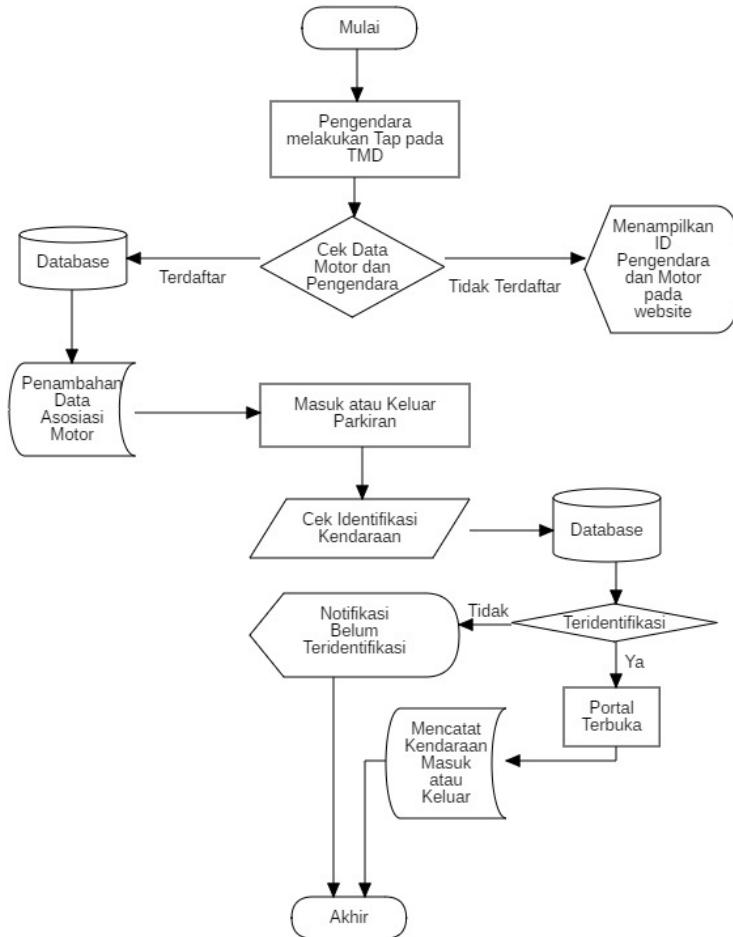
Gambar 3.1 Alur Interaksi *Device*.

untuk mengimplementasikan miniatur ini dibutuhkan beberapa alat seperti pada Tabel 3.1 Alat Sistem Parkir Cerdas.

Tabel 3.1 Alat Sistem Parkir Cerdas.

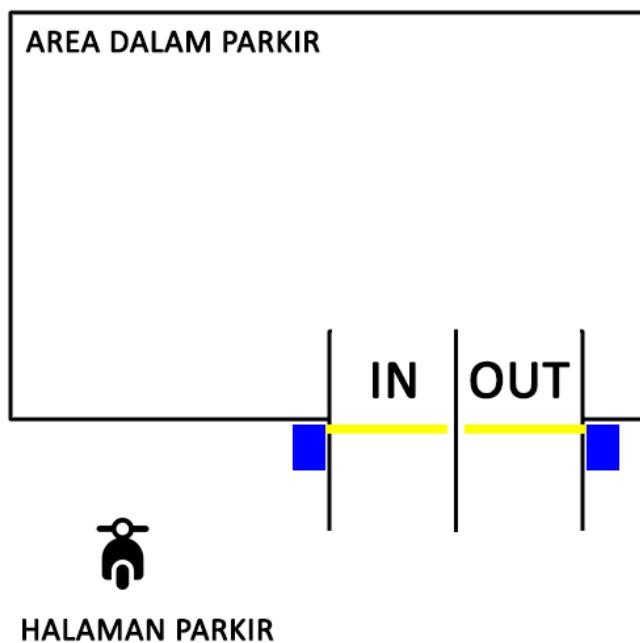
Alat	Deskripsi	Fungsi
Kartu Pengenal (RFID Tag)	Kartu pengenal pengendara.	Menyimpan identitas pengendara.
TMD (<i>Transceiver Motorcycle Device</i>)	Alat yang disematkan pada sepeda motor dengan menggunakan accu sebagai sumber listrik.	<ol style="list-style-type: none"> 1) Menghubungkan ke <i>WiFi</i>. 2) Menerima input dari <i>RFID Tag</i>. 3) Mengirimkan input ke <i>database</i> sebagai bentuk asosiasi.
SPD (<i>Smart Portal Device</i>)	Alat yang diletakan pada portal masuk dan keluar.	<ol style="list-style-type: none"> 1) Menghubungkan ke <i>WiFi</i>. 2) Mengecek asosiasi pada <i>database</i>. 3) Mengirimkan input ke <i>database</i> sebagai bentuk sepeda motor telah masuk. 4) Membuka dan menutup palang portal.
<i>Monitoring Control System</i>	Sistem berbasis website untuk melakukan pemantauan dan control.	<ol style="list-style-type: none"> 1) Melakukan pemantauan sepeda motor masuk dan keluar. 2) Mengelola data sepeda motor. 3) Mengelola data pengendara.

Pada implementasi miniatur ini *Transceiver Motorcycle Device* diasumsikan sudah terasosiasi dengan pengendara, dan *Smart Portal Device* akan mengecek asosiasi antara pengendara dengan sepeda motor yang digunakan, seperti pada Gambar 3.2 *Flowchart Sistem Parkir Cerdas*.



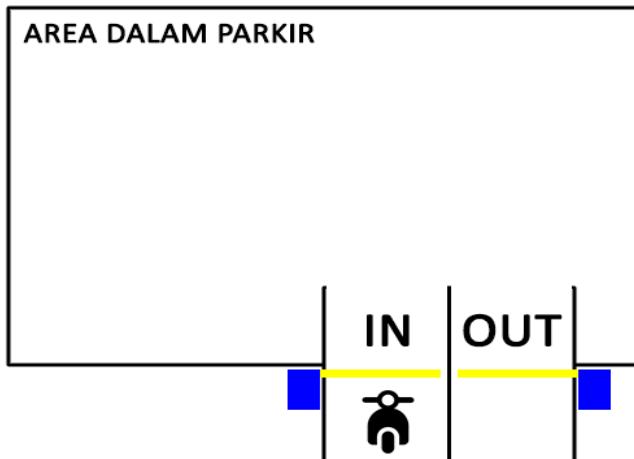
Gambar 3.2 *Flowchart Sistem Parkir Cerdas*.

Ketika pengendara masuk ke halaman parkir, *Transceiver Motorcycle Device* akan secara otomatis terhubung dengan *WiFi Access Point* dan mengeluarkan bunyi yang menandakan bahwa *Transceiver Motorcycle Device* sudah terhubung dengan baik, pengendara yang masuk ke parkiran melakukan tap terlebih dahulu pada sepeda motornya, hal ini dilakukan untuk mengirimkan data dari kartu pengenal yang nantinya akan di asosiasikan dengan sepeda motor yang digunakan, apabila tap sudah berhasil, *Transceiver Motorcycle Device* akan mengeluarkan bunyi menandakan bahwa identitas pengendara dan sepeda motor sudah terasosiasi dan tercatat dengan baik.



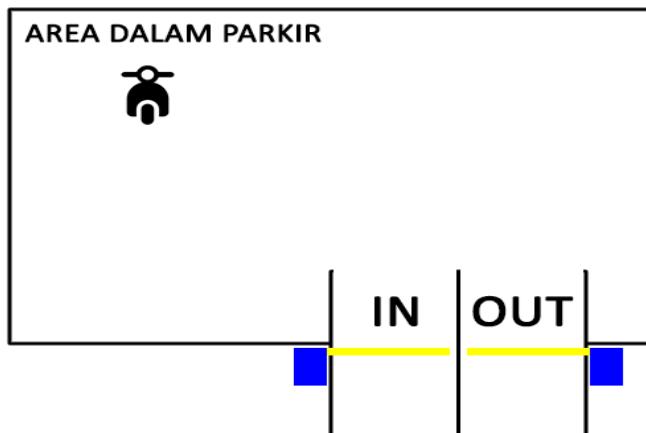
Gambar 3.3 Alur Parkir Cerdas Masuk – 1.

Pengendara yang sudah terasosiasi dengan sepeda motornya, selanjutnya melakukan tap kembali pada *Smart Portal Device*, ini dilakukan untuk mengecek apakah pengendara sudah terasosiasi atau belum, jika pengendara sudah terasosiasi maka portal akan secara otomatis terbuka dan mengirimkan data ke *database* menandakan bahwa pengendara sudah masuk dan sepeda motor berada di dalam parkiran, apabila pengendara belum terasosiasi dengan sepeda motor manapun, *Smart Portal Device* akan mengeluarkan bunyi yang menandakan bahwa pengendara belum terasosiasi dengan sepeda motor.



HALAMAN PARKIR

Gambar 3.4 Alur Parkir Cerdas Masuk – 2.



HALAMAN PARKIR

Gambar 3.5 Alur Parkir Cerdas Masuk – 3.

Pada alur keluar Sistem Parkir Sepeda Motor Cerdas memiliki kesamaan dengan alur masuk, yaitu:

1. Pengendara melakukan tap ke sepeda motor, hal ini dilakukan untuk melakukan asosiasi terhadap motor yang digunakan, apa motor tersebut motor yang sama atau tidak.
2. Pengendara melakukan tap pada mesin portal parkir, yang nantinya *Smart Portal Device* akan melakukan cek apa pengendara tersebut sudah pernah masuk dan sedang berada didalam atau tidak.
3. Portal akan terbuka dan *Smart Portal Device* mengirimkan data ke *database* menandakan bahwa pengendara sudah keluar dari parkiran.

3.2 Perancangan Database

Rancangan *database* ini digunakan sebagai media penyimpanan data oleh Arduino yang nantinya data akan berubah sesuai data yang dikirim oleh Arduino, yang akan ditampilkan melalui *website*. *Database* akan berisi 4 tabel seperti pada Tabel 3.2 :

Tabel 3.2 Deskripsi Tabel *Database*.

Nama Tabel	Deskripsi
SMP_MOTOR	Tabel yang berisi data motor.
SMP_PENGENDARA	Tabel yang berisi data pengendara.
SMP_ACCESS	Tabel yang berisi asosiasi antara pengendara dengan motor.
SMP_PORTAL	Tabel yang berisi data pengendara yang sudah masuk atau keluar dari parkiran.

Tabel 3.3 Detail Tabel SMP_MOTOR.

SMP_MOTOR		
Atribut	Data Type	Deskripsi
MTR_ID	Integer (11)	Atribut <i>primary key</i> pada tabel ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
PGN_ID	Integer (11)	Atribut <i>foreign key</i> pada tabel yang didapat dari tabel SMP_PENGENDARA.
MTR_HW_ID	Varchar (50)	Atribut yang berisi nomor <i>hardware</i> dari alat yang disematkan pada motor.
MTR_NO_POL	Varchar (15)	Atribut yang berisi nomor polisi motor yang disematkan alat.

Tabel 3.4 Detail Tabel SMP_PENGENDARA.

SMP_PENGENDARA		
Atribut	DataType	Deskripsi
PGN_ID	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
PGN_CARD_ID	Varchar (50)	Atribut yang berisi ID dari RFID Card.
PGN_NAMA	Varchar (25)	Atribut yang berisi nama pengendara.

Tabel 3.5 Detail Tabel SMP_ACCESS.

SMP_ACCESS		
Atribut	DataType	Deskripsi
ACS_ID	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
PGN_ID	Integer (11)	Atribut <i>foreign key</i> pada tabel yang didapat dari tabel SMP_PENGENDARA.
MTR_ID	Integer (11)	Atribut <i>foreign key</i> pada tabel yang didapat dari tabel SMP_MOTOR.
ACS_TIME_IN	Datetime	Atribut yang digunakan sebagai status masuk jika pengendara sudah melakukan tap pada motor.
ACS_TIME_OUT	Datetime	Atribut yang digunakan sebagai status keluar jika pengendara sudah melakukan tap pada motor.
IS_DEL	Integer (1)	Atribut yang digunakan sebagai status jika proses telah selesai.

Tabel 3.6 Detail Tabel SMP_PORTAL.

SMP_PENGENDARA		
Atribut	DataType	Deskripsi
PTL_ID	Integer (11)	Atribut <i>primary key</i> pada table ini yang berfungsi sebagai ID <i>primary</i> pada tabel.
ACS_ID	Integer (11)	Atribut <i>foreign key</i> pada tabel yang didapat dari tabel SMP_ACCESS.
PTL_IN	Datetime	Atribut yang digunakan sebagai status pengendara sudah masuk kedalam parkiran.
PTL_OUT	Datetime	Atribut yang digunakan sebagai status pengendara sudah keluar dari dalam parkiran.
IS_DEL	Integer (1)	Atribut yang digunakan sebagai status jika proses telah selesai.

3.3 Perancangan Alat

3.3.1 Transceiver Motorcycle Device

Perancangan alat *Transceiver Motorcycle Device* akan menggunakan beberapa komponen *hardware*, dengan WeMos D1 R1 sebagai *microcontroller*, diantaranya :

1. RFID Reader RC522
2. Buzzer

Microcontroller WeMos D1 R1 nantinya akan dihubungkan dengan masing masing komponen, yang akan berfungsi sebagai berikut:

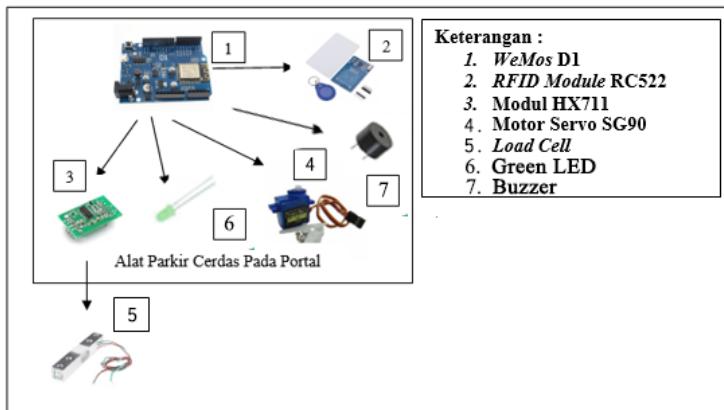
1. Mengeluarkan bunyi beep sebanyak dua kali, beep pertama adalah notifikasi bahwa *Smart Portal Device* telah terhubung dengan *internet*, beep kedua adalah notifikasi *Smart Portal Device* telah terhubung dengan *database*.
2. Melakukan cek pada tabel SMP_ACCESS jika terdapat pengendara yang sudah melakukan tap pada motornya, akan melakukan penambahan data pada tabel SMP_ACCESS, jika pengendara tidak dikenali, maka *Transceiver Motorcycle Device* akan mengeluarkan notifikasi beep sebanyak dua kali secara beruntun.

3.3.2 *Smart Portal Device*

Perancangan alat *Smart Portal Device* akan menggunakan beberapa komponen *hardware*, dengan WeMos D1 R1 sebagai *microcontroller*, diantaranya :

1. RFID Reader RC522
2. Motor Servo SG90
3. Modul HX711
4. Load Cell
5. Buzzer
6. Green LED

Microcontroller WeMos D1 R1 nantinya akan dihubungkan dengan masing masing komponen seperti pada Gambar 3.6 *Smart Portal Device*.



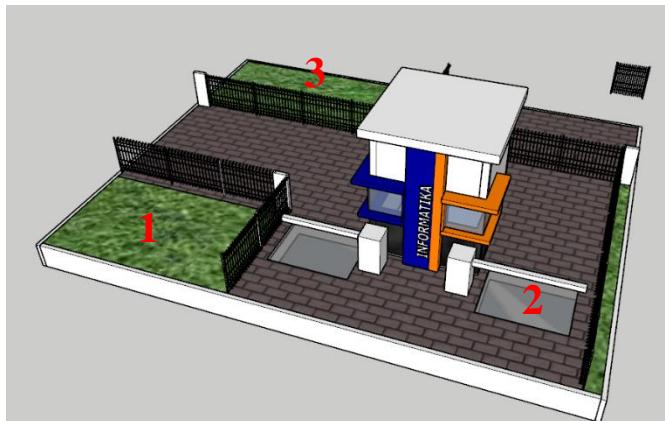
Gambar 3.6 *Smart Portal Device*.

Smart Portal Device dibagi menjadi dua, yaitu *Smart Portal Device* Masuk dan Keluar, *Smart Portal Device* akan berfungsi sebagai berikut:

1. Mengeluarkan bunyi beep sebanyak dua kali, beep pertama adalah notifikasi bahwa *Smart Portal Device* telah terhubung dengan *internet*, beep kedua adalah notifikasi *Smart Portal Device* telah terhubung dengan *database*.
2. Melakukan cek pada tabel SMP_ACCESS jika terdapat pengendara yang sudah melakukan tap pada motornya, akan melakukan penambahan data pada SMP_PORTAL, jika pengendara tidak dikenali, maka *Smart Portal Device* akan mengeluarkan notifikasi beep sebanyak dua kali secara beruntun.
3. Membuka portal dengan menggunakan Motor Servo SG90 jika proses pengecekan dan pemasukan data telah selesai.
4. Melakukan update status IS_DEL = 1 pada tabel SMP_ACCESS dan SMP_PORTAL untuk menandakan proses telah selesai.

3.3.3 Miniatur

Rancang miniatur yang akan dibuat pada Tugas Akhir adalah sebagai berikut



Gambar 3.7 Rancang Miniatur Sistem Parkir Cerdas.

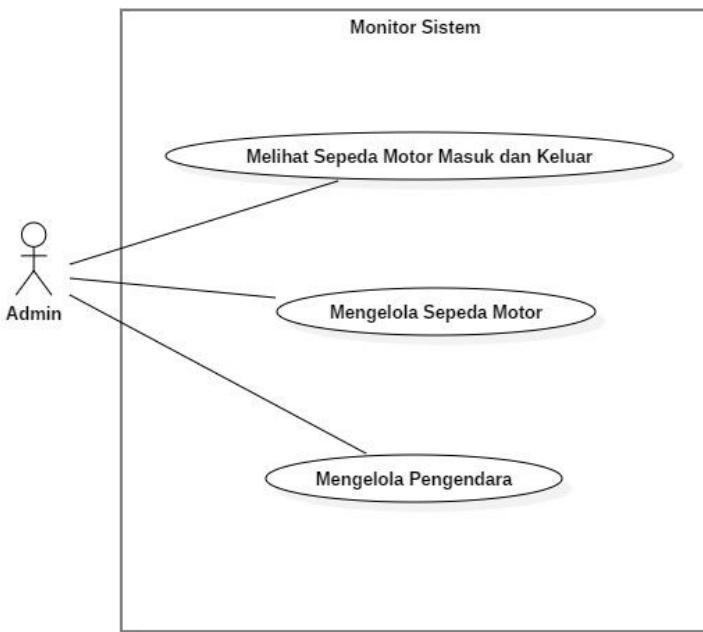
Deskripsi pada gambar adalah sebagai berikut:

1. Lokasi penempatan *RFID Reader* pada miniatur.
2. Lokasi sensor *Load Cell* pada miniatur.
3. Lokasi penyimpanan *Smart Portal Device*, komponen lainnya akan berada pada lokasi ini juga.

Miniatur akan dibuat dengan menggunakan bahan dasar PVC.

3.4 Perancangan Sistem Monitor dan Kontrol

Perancangan sistem monitor dan control digambarkan melalui Use Case Diagram pada Gambar 3.8 User Case Diagram Sistem Monitor.



Gambar 3.8 Use Case Diagram Sistem Monitor.

Sistem Monitor dan Kontrol dibangun menggunakan *framework* CodeIgniter 3 untuk dapat memantau jalannya proses Sistem Parkir Sepeda Motor Cerdas, yang dapat melakukan:

1. Pemantauan sepeda motor yang masuk dan keluar.
2. Mengelola sepeda motor berupa melakukan penambahan, penghapusan, serta update data sepeda motor.
3. Mengelola pengendara berupa melakukan penambahan, penghapusan, serta update data pengendara sepeda motor.

[*Halaman ini sengaja dikosongkan*]

BAB IV

IMPLEMENTASI

Pada bab ini akan dijelaskan tentang implementasi yang dilakukan berdasarkan perancangan pada bab sebelumnya

4.1 Implementasi Database

Pada implementasi *database* ini, penulis menggunakan *tool* Navicat Premium untuk mengakses *database*, dan XAMPP sebagai penyimpanan *database* MySQL serta *Web Server*, *database* akan dibuat 4 tabel seperti yang telah dijelaskan pada bab perancangan.

1. Tabel SMP_MOTOR

Pada pengimplementasian tabel SMP_MOTOR akan menjalankan *query* seperti Gambar 4.1 pada Navicat Premium.

```
CREATE TABLE `smp_motor` (
  `MTR_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PGN_ID` int(11) NULL DEFAULT NULL,
  `MTR_HW_ID` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
  `MTR_NO_POL` varchar(15) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
  PRIMARY KEY (`MTR_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;
```

Gambar 4.1 Kode SQL SMP_MOTOR.

2. Tabel SMP_PENGENDARA

Pada pengimplementasian tabel SMP_PENGENDARA akan menjalankan *query* seperti Gambar 4.2 pada Navicat Premium.

```
CREATE TABLE `smp_pengendara` (
  `PGN_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PGN_CARD_ID` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
  `PGN_NAMA` varchar(25) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
  PRIMARY KEY (`PGN_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;
```

Gambar 4.2 Kode SQL SMP_PENGENDARA.

3. Tabel SMP_ACCESS

Pada pengimplementasian tabel SMP_ACCESS akan menjalankan *query* seperti Gambar 4.3 pada *Navicat Premium*.

```
CREATE TABLE `smp_access` (
  `ACS_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PGN_ID` int(11) NOT NULL,
  `MTR_ID` int(11) NOT NULL,
  `ACS_TIME_IN` datetime(0) NOT NULL DEFAULT CURRENT_TIMESTAMP(),
  `ACS_TIME_OUT` datetime(0) NOT NULL,
  `IS_DEL` int(1) NOT NULL DEFAULT 0,
  PRIMARY KEY (`ACS_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;
```

Gambar 4.3 Kode SQL SMP_ACCESS.

4. Tabel SMP_PORTAL

Pada pengimplementasian tabel SMP_PORTAL akan menjalankan *query* seperti Gambar 4.4 pada Navicat Premium.

```

CREATE TABLE `smp_portal` (
  `PTL_ID` int(11) NOT NULL AUTO_INCREMENT,
  `ACS_ID` int(11) NOT NULL,
  `PTL_IN` datetime(0) NOT NULL DEFAULT CURRENT_TIMESTAMP(0),
  `PTL_OUT` datetime(0) NOT NULL,
  `IS_DEL` int(11) NOT NULL DEFAULT 0,
  PRIMARY KEY (`PTL_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;

```

Gambar 4.4 Kode SQL SMP_PORTAL.

Perlu dilakukan konfigurasi untuk memberikan *foreign key* pada tabel SMP_ACCESS, SMP_PORTAL, serta SMP_MOTOR, dengan menjalankan *query* seperti berikut pada Navicat Premium.

```

ALTER TABLE smp_access
ADD CONSTRAINT `FK_RELATIONSHIP_1`
FOREIGN KEY (`PGN_ID`) REFERENCES
`smp_pengendara`(`PGN_ID`) ON DELETE CASCADE
ON UPDATE CASCADE,
ADD CONSTRAINT `FK_RELATIONSHIP_2`
FOREIGN KEY (`MTR_ID`) REFERENCES
`smp_motor`(`MTR_ID`) ON DELETE CASCADE
ON UPDATE CASCADE;

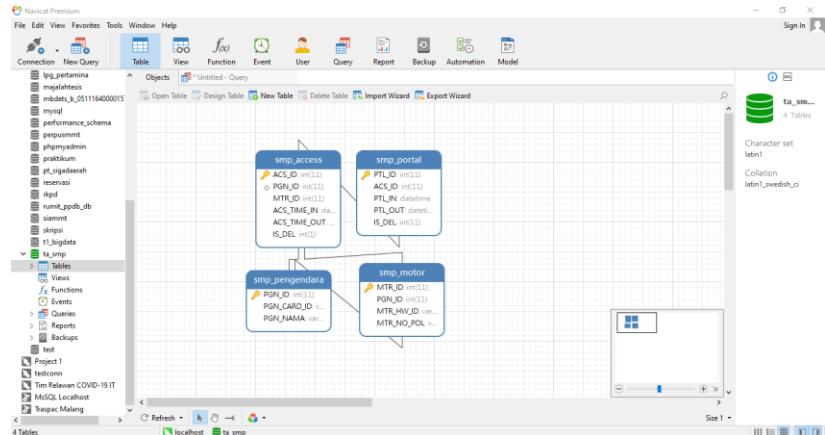
ALTER TABLE smp_portal
ADD CONSTRAINT `FK_RELATIONSHIP_3`
FOREIGN KEY (`ACS_ID`) REFERENCES
`smp_access`(`ACS_ID`) ON DELETE CASCADE
ON UPDATE CASCADE;

ALTER TABLE smp_motor
ADD CONSTRAINT `FK_RELATIONSHIP_4`
FOREIGN KEY (`PGN_ID`) REFERENCES
`smp_pengendara`(`PGN_ID`) ON DELETE CASCADE
ON UPDATE CASCADE;

```

Gambar 4.5 Kode SQL Foreign Key.

Setelah itu akan dihasilkan 4 tabel yang telah di dengan menjalnkan *query* sebelumnya, hasil *query* yang di eksekusi dapat dilihat pada gambar 4.6.

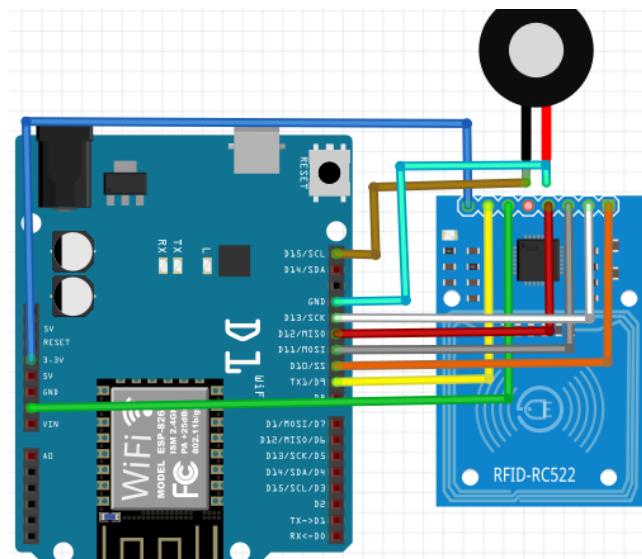


Gambar 4.6 Hasil Eksekusi SQL.

4.2 Implementasi Perangkat

4.2.1. Transceiver Motorcycle Device

Implementasi *Transceiver Motorcycle Device* diperlukan beberapa alat yang dihubungkan pada Arduino seperti penjelasan pada bab perancangan, RFC522 dan Buzzer akan dihubungkan pada Arduino, dimana RFC522 nantinya akan berfungsi sebagai pembaca kartu yang ditap, sedangkan *buzzer* akan berfungsi untuk memberikan notifikasi melalui bunyi, rangkaian alat pada Arduino adalah seperti berikut:



Gambar 4.7 Rangkaian Arduino TMD.

Tabel 4.1 Rangkaian Arduino TMD.

PIN		
RFC522	Arduino	Buzzer
SDA	D10/SS	-
SCK	D13/SCK	-
MOSI	D11/MOSI	-
MISO	D12/MISO	-
IRQ	-	-
GND	GND	-
RST	TX1/D9	-
3.3V	3.3V	-
-	D15/SCL	Input
-	GND	GND

Rangkaian Arduino dapat dijalankan dengan memasukan kode program kedalam Arduino melalui Arduino IDE, Implementasi kode *Transceiver Motorcycle Device* adalah sebagai berikut:

```
#include <SPI.h>
#include <MFRC522.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>

#define SS_PIN D10 //Define start
mrfc522 pin
#define RST_PIN D9 //Define reset
mrfc522 pin
#define BUZZER D15 //Define buzzer pin
```

Gambar 4.8 TMD Define Pin.

Baris kode program ini berisi semua *library* yang digunakan pada *Transceiver Motorcycle Device* serta deklarasi pin yang digunakan pada Arduino, SS_PIN dan RST_PIN adalah pin input yang digunakan oleh RFC522.

```

// ----- MRFC Configuration
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

// ----- WiFi Configuration
char ssid[] = "Adek Kecil Queena";
// Network Name
char pass[] = "rumahparung141291"; // Network Password

String BASE_URL =
"http://192.168.100.38/codeigniter/iot_ta/";

// ----- MySQL Configuration
char user[] = "userata"; // MySQL user
char password[] = "userata"; // MySQL password
WiFiClient client;
IPAddress server_addr(192, 168, 100, 38); // MySQL server IP
MySQL_Connection conn((Client *)&client);
MySQL_Cursor cur = MySQL_Cursor(&conn);

```

Gambar 4.9 TMD Konfigurasi.

Baris kode program ini berisi konfigurasi RFC522 di dalam Arduino, koneksi WiFi, serta konfigurasi *user* dan *server* dari MySQL.

```

// ----- Setup Begin
void setup() {
  Serial.begin(115200);
  SPI.begin();
  connectPin();
  connectWiFi();
}

```

Gambar 4.10 TMD Setup.

Fungsi void *setup* adalah fungsi utama pada Arduino, fungsi ini akan dijalankan secara otomatis oleh Arduino ketika terhubung dengan arus listrik, fungsi ini akan menjalankan fungsi *connectPin* yang berisi untuk menghubungkan pin yang telah didefinisikan kedalam Arduino, serta fungsi *connectWifi* yang berisi kode program untuk menghubungkan Arduino dengan *WiFi* dan server MySQL, ketika *WiFi* telah terhubung, Arduino akan mengeluarkan bunyi beep yang menandakan *WiFi* telah terhubung, selanjutnya beep kedua adalah menandakan bahwa Arduino telah terhubung dengan server MySQL.

Setelah menjalankan fungsi void *setup* diawal, Arduino akan melanjutkannya dengan membaca fungsi void *loop*, fungsi inilah yang nantinya akan terus dilakukan pengulangan selama Arduino terhubung dengan listrik, pada fungsi *loop* ini TMD (*Transceiver Motorcycle Device*) akan melakukan pengecekan pengendara serta motor, jika pengendara belum terdaftar pada tabel SMP_ACCESS, maka disaat melakukan tap pada motor, TMD akan menambahkan data ACS_TIME_IN yang digunakan sebagai waktu pengendara melakukan tap atau verifikasi. Sebelum melakukan cek, TMD akan melakukan cek data pengendara dan data motor yang ada, jika data motor dan data pengendara ada maka proses pengecekan dapat dilanjutkan, tetapi jika data tidak ada, TMD secara otomatis akan mengirimkan ID kartu dari RFID ke web untuk dilakukan registrasi pengendara, sama halnya dengan ID alat yang berada di motor.

```
// ----- Setup Loop
void loop() {
    ...
    MTR_ID = getMtrId(mac_address);
    PGN_ID = getPgnId(hex);
    ACS_ID = getAcsId(PGN_ID,MTR_ID);

    //Assume acs_id one only
    if ( ACS_ID == "" && PGN_ID != "" && MTR_ID != ""
    ) {
        ACSMTR_ID = getAcsMtrId(MTR_ID);
        //Check if mtr_id already used
        if (MTR_ID == ACSMTR_ID) {
            Serial.println("Data already exists...");
            ...
        } else {
            //If no data on smp_access, insert data to
            smp_access
            insertMasuk(PGN_ID,MTR_ID);
            ...
        }
        //Data not found and post data to CI
    } else if( ACS_ID != "" && PGN_ID != "" &&
    MTR_ID != "" ){
        Serial.println("Please tap your card to
        portal...");
        } if( PGN_ID == "" ) {
            parseUIDPengendara(hex);
        } if( MTR_ID == "" ) {
            parseUIDMotor(mac_address);
        }
        ...
    }
}
```

Gambar 4.11 TMD Loop.

Tabel 4.2 TMD Loop.

PGN_ID	Variabel penyimpanan hasil data PGN_ID dari tabel SMP_PENGENDARA.
MTR_ID	Variabel penyimpanan hasil data MTR_ID dari tabel SMP_MOTOR .
ACS_ID	Variabel penyimpanan hasil data ACS_ID dari tabel SMP_ACCESS.
ACSMTR_ID	Variabel penyimpanan hasil data MTR_ID dari tabel SMP_ACCESS.
insertMasuk(PGN_ID,MTR_ID)	Fungsi untuk menambahkan data pada tabel SMP_ACCESS dimana PGN_ID dan MTR_ID adalah parameternya.
parseUIDPengendara(hex)	Fungsi untuk mengirimkan parameter hex ke website untuk menambahkan data pengguna
parseUIDMotor(mac_address)	Fungsi untuk mengirimkan parameter mac_address ke website untuk menambahkan data motor.

Implementasi kode program *Transceiver Motorcycle Device* secara penuh terlampir di dalam lampiran.

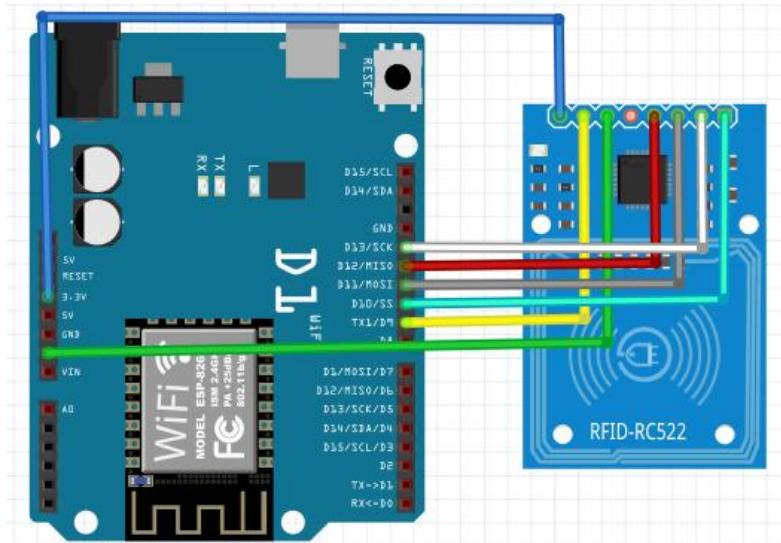
4.2.2. Smart Portal Device

Implementasi *Smart Portal Device*, akan di bagi menjadi dua implementasi, yaitu portal masuk dan keluar.

4.2.1.1 Smart Portal Device Masuk

Dalam implementasi portal device, diperlukan beberapa alat yang akan dihubungkan pada Arduino seperti penjelasan pada bab perancangan, rangkaian alat pada Arduino adalah sebagai berikut:

1. RFID Reader RC522



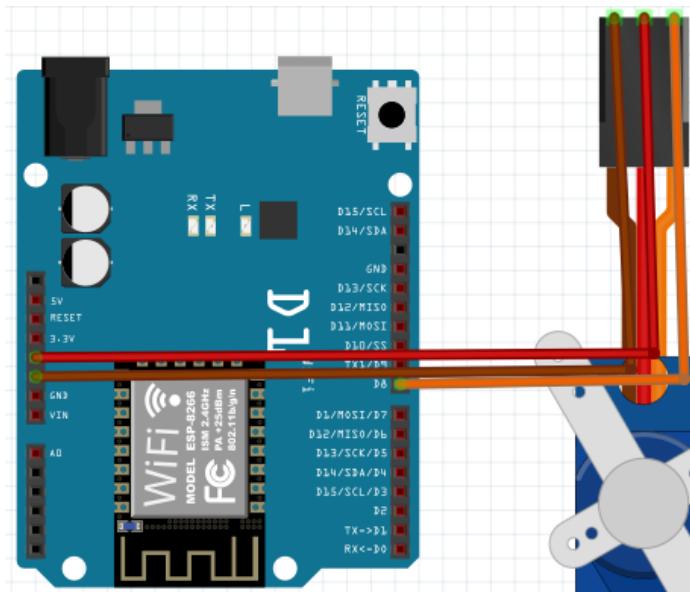
Gambar 4.12 Rangkaian Arduino RFC522.

Gambar 4.12 adalah gambar rangkaian RFC522 yang terpasang dengan Arduino, dimana pin yang digunakan seperti pada Tabel 4.3 Rangkaian Arduino RFC522.

Tabel 4.3 Rangkaian Arduino RFC522.

PIN	
RFC522	Arduino
SDA	D10/SS
SCK	D13/SCK
MOSI	D11/MOSI
MISO	D12/MISO
IRQ	-
GND	GND
RST	TX1/D9
3.3V	3.3V

2. Motor Servo SG90



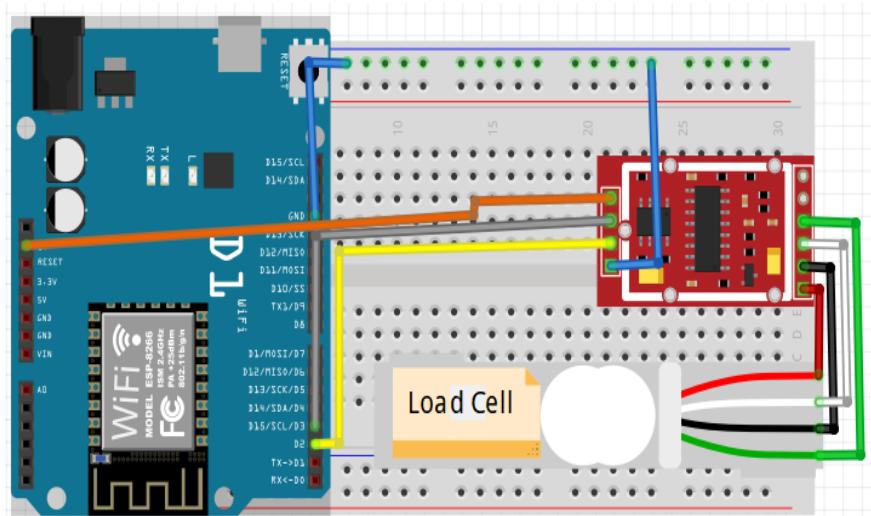
Gambar 4.13 Rangkaian Arduino SG90.

Gambar 4.13 adalah gambar rangkaian SG90 yang terpasang dengan Arduino, dimana pin yang digunakan seperti pada Tabel 4.4 Rangkaian Arduino SG90.

Tabel 4.4 Rangkaian Arduino SG90.

PIN	
SG90	Arduino
Signal	D8
5V	5V
GND	GND

3. Modul HX711 dan Load Cell



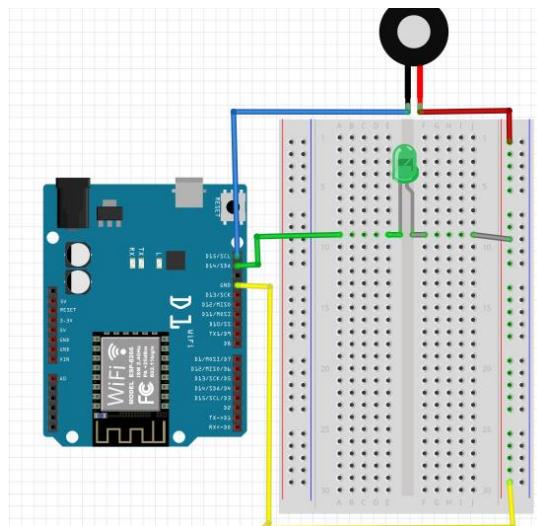
Gambar 4.14 Rangkaian Arduino Load HX711.

Gambar 4.14 adalah gambar rangkaian Load HX711 yang terpasang dengan Arduino, dimana pin yang digunakan seperti pada Tabel 4.5 Rangkaian Arduino Load HX711.

Tabel 4.5 Rangkaian Arduino Load HX711.

PIN		
Load Cell	HX711	Arduino
Red E+	E+	-
Black E-	E-	-
White A-	A-	-
Green A+	A+	-
-	GND	GND
-	DT	D15/SCL/D3
-	SCK	D2
-	VCC	5V

4. Buzzer dan Green LED



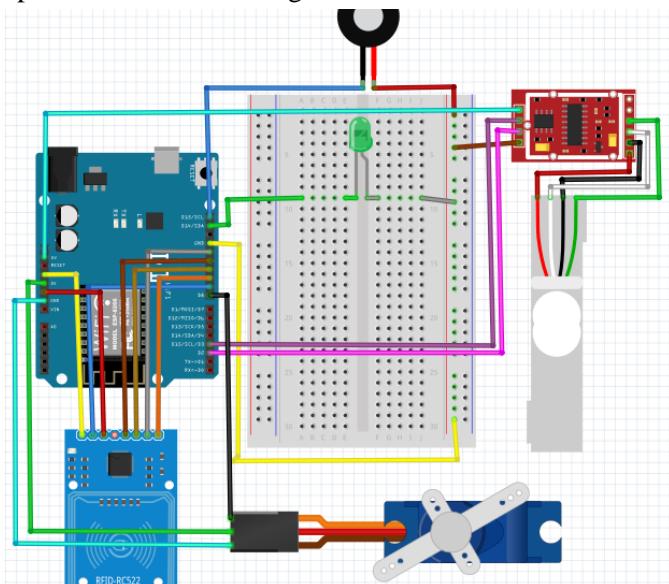
Gambar 4.15 Rangkaian Arduino Buzzer LED.

Gambar 4.15 adalah gambar rangkaian *Buzzer LED* yang terpasang dengan Arduino melalui *breadboard*, dimana pin yang digunakan seperti pada Tabel 4.6 Rangkaian Arduino *Buzzer LED*.

Tabel 4.6 Rangkaian Arduino Buzzer LED.

PIN			
LED	Buzzer	Arduino	
Input	-	D14/SDA	
GND	-	GND	
-	Input	D15/SCL	
-	GND	GND	

Rangkaian Arduino secara keseluruhan terdapat pada Gambar 4.16 Rangkaian Arduino.



Gambar 4.16 Rangkaian Arduino.

Rangkaian Arduino dapat dijalankan dengan memasukan kode program kedalam Arduino melalui Arduino IDE, Implementasi kode *Smart Portal Device* Masuk adalah sebagai berikut:

```
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <HX711.h>
```

Gambar 4.17 SPD Masuk Library.

Baris kode program ini berisi semua *library* yang digunakan pada *Smart Portal Device* Masuk.

```
#define SS_PIN D10 //Define start mrfc522 pin
#define RST_PIN D9 //Define reset mrfc522 pin
#define LED_G D14 //Define green LED pin
#define BUZZER D15 //Define buzzer pin
#define SERVO D8 //Define servo pin
#define DOUT D2 //Define hx711 dout pin
#define SCK D3 //Define hx711 sck pin
```

Gambar 4.18 SPD Masuk Define PIN.

Baris kode program ini berisi deklarasi pin yang digunakan pada Arduino, semua pin *input* yang sebelumnya telah terpasang.

```

// ----- HX711 Configuration
HX711 scale(DOUT, SCK);
float calibration_factor = -7050;
int GRAM;

// ----- MRFC Configuration
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create MFRC522 instance.

// ----- MotorServo Configuration
Servo myServo; //Define servo name
int sServo = 0; //Define status servo open

// ----- WiFi Configuration
char ssid[] = "Adek Kecil Queena"; // Network Name
char pass[] = "rumahparung141291"; // Network Password

// ----- MySQL Configuration
char user[] = "userta"; // MySQL user
char password[] = "userta"; // MySQL password
WiFiClient client;
IPAddress server_addr(192, 168, 100, 38); // MySQL server IP
MySQL Connection conn((Client *)&client);

```

Gambar 4.19 SPD Masuk Konfigurasi.

Baris kode program ini berisi konfigurasi sensor didalam Arduino, terdapat konfigurasi kalibrasi untuk HX711, konfigurasi Servo SG90, koneksiitas WiFi, serta konfigurasi user dan server dari MySQL.

```
// ----- Setup Begin
void setup() {
    Serial.begin(9600);
    SPI.begin();
    connectPin();
    connectWiFi();
}
```

Gambar 4.20 SPD Masuk Setup.

Fungsi void *setup* adalah fungsi utama pada Arduino, fungsi ini akan dijalankan secara otomatis oleh Arduino ketika terhubung dengan arus listrik, fungsi ini akan menjalankan fungsi *connectPin* yang berisi untuk menghubungkan sensor yang telah didefinisikan kedalam pin Arduino, serta fungsi *connectWifi* yang berisi kode program untuk menghubungkan Arduino dengan *WiFi* dan server MySQL, ketika *WiFi* telah terhubung, Arduino akan mengeluarkan bunyi beep yang menandakan *WiFi* telah terhubung, selanjutnya beep kedua adalah menandakan bahwa Arduino telah terhubung dengan *server MySQL*.

```
// ----- Function Define Pin Output
void connectPin() {
    scale.set_scale(); //set scale pembacaan berat
    scale.tare(); // mengenolkan pembacaan berat
    mfrc522.PCD_Init(); //Initiate MFRC522
    myServo.attach(SERVO); //servo pin
    myServo.write(90); //servo start position
    pinMode(LED_G, OUTPUT); //pinMode(LED_R,
    OUTPUT);
```

Gambar 4.21 SPD Masuk Fungsi ConnectPin.

```

// ----- Function Connect to WiFi and
MySQL
void connectWiFi() {
    // WiFi Connect
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED) {
        delay(200);
        Serial.print(".");
    }
    tone(BUZZER, 500);
    delay(300);
    noTone(BUZZER);
    Serial.println("");
    Serial.println("WiFi Connected");
    Serial.print("Assigned IP: ");
    Serial.println(WiFi.localIP());
    //MySQL Connect
    Serial.println("Connecting to database");
    while (conn.connect(server_addr, 3306, user,
password) != true) {
        delay(200);
        Serial.print (".");
    }
    tone(BUZZER, 500);
    delay(300);
    noTone(BUZZER);
    Serial.println("Connected to SQL Server!");
    Serial.println("");
}

```

Gambar 4.22 SPD Masuk Fungsi ConnectWifi.

Setelah menjalankan fungsi void *setup* diawal, Arduino akan melanjutkannya dengan membaca fungsi void *loop*, fungsi inilah yang nantinya akan terus di lakukan pengulangan selama Arduino terhubung dengan listrik.

```
// ----- Setup Loop
void loop() {
    ...
    if ( GRAM != 0 && sServo == 0 ) {
        ...
        messageReadInput(GRAM,hex,dec);
        PGN_ID = getPgnId(hex); //Mengambil PGN_ID dari
        table smp_pengendara
        ACS_ID = getAcsId(PGN_ID); //Mengambil ACS_ID
        dari table smp_access
        ACS_IN = getAcsIn(ACS_ID); //Mengambil
        ACS_TIME_IN dari table smp_access

        Serial.print("Message : ");
        if (PGN_ID != "" && ACS_IN != "0000-00-00
        00:00:00") { //Cek jika ada pengendara yang sudah
        melakukan tap pada Motor
            if (getPtlAcs(ACS_ID) == "") { //Cek apa
            pengendara sudah ada
                // Servo Open and Close
                insertMasuk(ACS_ID); //Insert data ke
                smp_portal
                ...
                sServo = 1;
            } else Serial.println("Id already exist.");
            Serial.println();
        } else {
            ...
        }
    } else if (GRAM == 0 && sServo == 1){
        sServo = 0;
        ...
    }
}
```

Gambar 4.23 SPD Masuk Fungsi Loop.

```
// ----- Setup Loop
void loop() {
    ...
    if ( GRAM != 0 && sServo == 0 ){
        // Look for new cards or Select one of the
        cards
        if ( ! mfrc522.PICC_IsNewCardPresent() || !
            mfrc522.PICC_ReadCardSerial() )
            return;

    ...

    hex.toUpperCase(); hex = hex.substring(1);
    hex.replace(" ", ":");

    dec.toUpperCase(); dec = dec.substring(1);

    messageReadInput(GRAM,hex,dec);
    ...
} else if (GRAM == 0 && sServo == 1){
    sServo = 0;
    ...
}
}
```

Gambar 4.24 SPD Masuk Fungsi Loop-1.

Fungsi *loop* akan mengecek jika terdeteksi adanya berat pada sensor *load cell* dan status dari Servo SG90 nol atau sedang tidak terbuka maka program akan membiarkan pengguna untuk melakukan tapping kemudian menampilkan data hasil dari tapping pengguna serta dilanjutkan kedalam proses pengecekan, tapi jika tidak terdeteksi berat dan status dari Servo SG90 adalah satu atau sedang terbuka, maka akan secara otomatis merubah status menjadi nol.

```
// ----- Setup Loop
void loop() {
    ...
    messageReadInput(GRAM,hex,dec);
    PGN_ID = getPgnId(hex); //Mengambil PGN_ID dari
    table smp_pengendara
    ACS_ID = getAcsId(PGN_ID); //Mengambil ACS_ID
    dari table smp_access
    ACS_IN = getAcsIn(PGN_ID); //Mengambil
    ACS_TIME_IN dari table smp_access

    Serial.print("Message : ");
    if (PGN_ID != "" && ACS_IN != "0000-00-00
    00:00:00") { //Cek jika ada pengendara yang sudah
    melakukan tap pada Motor
        if (getPtlAcs(ACS_ID) == "") { //Cek apa
        pengendara sudah ada
            // Servo Open and Close
            insertMasuk(ACS_ID); //Insert data ke
        smp_portal
            ...
            sServo = 1;
        } else Serial.println("Id already exist.");
        Serial.println();
    } else {
        ...
    }
    ...
}
```

Gambar 4.25 SPD Masuk Fungsi Loop-2.

Proses pengecekan saat melakukan tap adalah hasil yang dibaca oleh RFID akan diparsing kedalam fungsi untuk mendapatkan data pada MySQL, hasil yang didapatkan dari MySQL akan disimpan pada variabel untuk dilakukan cek

terhadap tap dengan data yang ada di MySQL penjelasan variabel seperti pada tabel 4.7:

Tabel 4.7 SPD Masuk Fungsi Loop-1.

PGN_ID	Variabel penyimpanan hasil data PGN_ID dari tabel SMP_PENGENDARA.
ACS_ID	Variabel penyimpanan hasil data ACS_ID dari tabel SMP_ACCESS.
ACS_IN	Variabel penyimpanan hasil data ACS_IN dari tabel SMP_ACCESS.

Proses pengecekan berlangsung pada baris berikut, dimana akan dilakukan pengecekan dari data yang telah diambil di MySQL sebelumnya.

```

if (PGN_ID != "" && ACS_IN != "0000-00-
00 00:00:00") {
    if (getPtlAcs(ACS_ID) == "") {
        // Servo Open and Close
        insertMasuk(ACS_ID);
        ...
        sServo = 1;
    } else Serial.println("Id
already exist.");
    Serial.println();
} else {
    ...
}

```

Gambar 4.26 SPD Masuk Fungsi Loop-3.

Jika hasil variabel PGN_ID tidak NULL dan hasil pada variabel ACS_IN tidak NULL, diasumsikan bahwa pengendara sudah terverifikasi dengan motor yang di pakai, maka akan dilakukan pengecekan portal access pada tabel SMP_PORTAL, jika hasil dari pengecekan portal adalah NULL artinya pengendara tersebut masih berada diluar parkiran dan belum masuk, maka akan dijalankan fungsi insertMasuk untuk menambahkan data pengendara kedalam SMP_PORTAL serta membuka palang portal, jika hasil dari portal access tidak NULL maka akan muncul pesan bahwa pengendara sudah berada di dalam.

Tabel 4.8 SPD Masuk Fungsi Loop-2.

getPtlAcs(ACS_ID)	Fungsi untuk melakukan cek pada tabel SMP_PORTAL dimana ACS_ID adalah parameternya.
insertMasuk(ACS_ID)	Fungsi untuk melakukan penambahan data pada tabel SMP_PORTAL dimana ACS_ID adalah parameternya.

Implementasi kode program *Smart Portal Device* Masuk secara penuh terlampir di dalam lampiran.

4.2.1.2 Smart Portal Device Keluar

Dalam implementasi portal *device* keluar, rangkaian alat pada Arduino sama seperti bab 4.2.1.1, portal keluar hanya memiliki perbedaan kode program dalam hal pengecekan saja, portal keluar akan melakukan cek pada pengendara yang sedang berada didalam parkiran dengan mengecek tabel SMP_ACCESS pada atribut ACS_TIME_IN dan ACS_TIME_OUT serta pada tabel SMP_PORTAL pada atribut PTL_IN, kedua tabel tersebut dilakukan pengecekan untuk mengetahui apa isi kedua atribut tersebut kosong atau tidak, jika kosong menandakan pengendara tidak berada didalam parkiran dan portal tidak akan terbuka, tetapi jika tidak kosong, maka pengendara diperlukan tap pada motor terlebih dahulu untuk mengupdate status ACS_TIME_OUT pada tabel SMP_ACCESS, portal akan terbuka jika atribut ACS_TIME_IN, ACS_TIME_OUT, PTL_IN memiliki value didalamnya, pengecekan dilakukan seperti pada Gambar 4.27 SPD Keluar Fungsi Loop-1.

```
// ----- Setup Loop
void loop() {
    ...
    PGN_ID = getPgnId(hex); //Mengambil PGN_ID
    dari table smp_pengendara
    ACS_ID = getAcsId(PGN_ID); //Mengambil ACS_ID
    dari table smp_access
    ACS_OUT = getAcsOut(ACS_ID); //Mengambil
    ACS_TIME_OUT dari table smp_access
    PTL_ID = getPtlAcs(ACS_ID); //Mengambil PTL_ID
    dari table smp_portal
    PTL_IN = getPtlIn(PTL_ID); //Mengambil PTL_IN
    dari table smp_portal
    PTL_OUT = getPtlOut(PTL_ID); //Mengambil
    PTL_IN dari table smp_portal

    Serial.print("Message : ");
    if (PTL_ID != "" && PTL_IN != "0000-00-00
    00:00:00") { //Cek motor sudah berada didalam
        if (ACS_OUT != "0000-00-00 00:00:00" &&
    PTL_OUT == "0000-00-00 00:00:00") { //Cek apa
        pengendara pernah ada didalam
        // Servo Open and Close
        updateKeluar(PTL_ID, ACS_ID); //Update
        data ke smp_portal
        ...
    } else Serial.println("Please tap to your
    motorcycle first.");
        Serial.println();
    } else {
        ...
    }
    ...
}
```

Gambar 4.27 SPD Keluar Fungsi Loop-1.

Tabel 4.9 SPD Keluar Fungsi Loop-1.

PGN_ID	Variabel penyimpanan hasil data PGN_ID dari tabel SMP_PENGENDARA.
ACS_ID	Variabel penyimpanan hasil data ACS_ID dari tabel SMP_ACCESS.
ACS_OUT	Variabel penyimpanan hasil data ACS_OUT dari tabel SMP_ACCESS.
PTL_ID	Variabel penyimpanan hasil data PTL_ID dari tabel SMP_PORTAL.
PTL_IN	Variabel penyimpanan hasil data PTL_IN dari tabel SMP_PORTAL.
PTL_OUT	Variabel penyimpanan hasil data PTL_OUT dari tabel SMP_PORTAL.
updateKeluar(ACS_ID)	Fungsi untuk melakukan cek pada tabel SMP_PORTAL dimana ACS_ID adalah parameternya.

Implementasi kode program *Smart Portal Device* Keluar secara penuh terlampir di dalam lampiran.

4.2.3. Miniatur

Implementasi miniatur adalah pembuatan minatur dengan menggunakan bahan dasar PVC, maket dibuat dengan menggunakan rancangan yang ada pada bab sebelumnya, maket terdiri dari tiga bagian, antara lain:



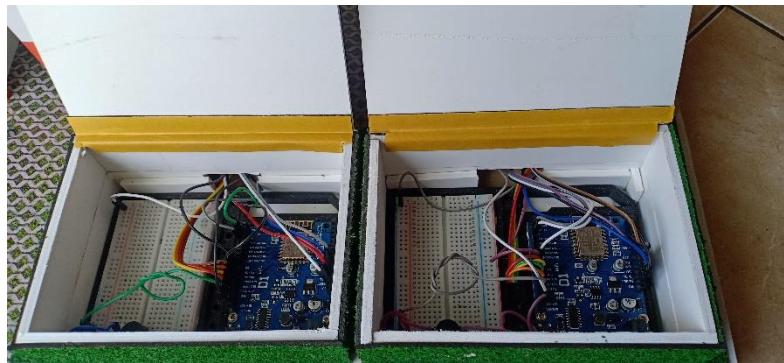
Gambar 4.28 Implementasi Miniatur-1.

Gambar 4.28 Implementasi Miniatur-1 merupakan hasil implementasi penempatan tap RFID untuk membuka portal.



Gambar 4.29 Implementasi Miniatur-2.

Gambar 4.29 Implementasi Miniatur-2 merupakan hasil implementasi penempatan sensor berat untuk membaca berat dari benda yang ada diatasnya.



Gambar 4.30 Implementasi Miniatur-3.

Gambar 4.30 Implementasi Miniatur-3 merupakan hasil implementasi penempatan Arduino, semua rangkaian kabel berada di bawah maket.



Gambar 4.31 Implementasi Miniatur-4.

Gambar 4.31 Implementasi Miniatur-4 merupakan hasil akhir implementasi miniatur secara detil.

4.3.1. CodeIgniter

Pada implementasi sistem monitor, penulis menggunakan *framework* CodeIgniter 3 dalam pembuatannya. Perlu dilakukan konfigurasi *database* agar *framework* dapat terhubung dengan MySQL seperti pada Gambar 4.32 Implementasi CodeIgniter SQL.

```
$db['default'] = array(
    'dsn'      => '',
    'hostname' => '192.168.100.38',
    'username' => 'userta',
    'password' => 'userta',
    'database' => 'ta_smp',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

Gambar 4.32 Implementasi CodeIgniter SQL

Konfigurasi *database* berada pada folder application/config/database.php, *database* pada CodeIgniter 3 dihubungkan dengan tabel yang sama dengan yang diakses oleh Arduino agar nantinya *framework* dapat menampilkan data yang telah diolah oleh Arduino.

Pada implementasi sistem monitor ini penulis hanya menggunakan tiga controller dan satu model, seperti berikut:

1. *Controller Motor.php*

Controller motor ini digunakan untuk melakukan pengelolaan data yang berhubungan dengan data motor, baik menambahkan, menghapus, mengedit data melalui *website* sistem monitor.

2. *Controller Pengendara.php*

Controller pengendara ini digunakan untuk melakukan pengelolaan data yang berhubungan dengan data pengendara, baik menambahkan, menghapus, mengedit data melalui *website* sistem monitor.

3. *Controller Smp.php*

Controller yang digunakan untuk melakukan monitor terhadap data yang diolah oleh Arduino.

4. *Model M_CallSQL.php*

Model berisi semua query yang dijalankan untuk melakukan interaksi antara *framework* dengan *database* agar bisa menjalankan fungsi yang ada di masing-masing controller.

4.3.2. Melihat Sepeda Motor Masuk dan Keluar

Untuk dapat memantau sepeda motor yang masuk dan keluar digunakan *controller* Smp.php yang berisi kode

program untuk melakukan pemanggilan *query* didalam *model*, hasil dari *query* tersebut akan ditampilkan pada *view*.

```
public function index() {
    ...
    'aksesmotor' => $this->M_CallsQL-
>getAksesMotor()->result(),
    'aksesportal' => $this->M_CallsQL-
>getAksesPortal()->result(),
    ...
}
```

Gambar 4.33 SMP Controller.

Pada kode program diatas, hasil dari *query* yang dieksekusi oleh *getAksesPortal* yang berada didalam *model M_CallsQL* akan disimpan didalam variabel *aksesportal*, dan *getAksesMotor* akan disimpan pada *aksesmotor*, kedua variabel ini berisi data yang didapatkan dari tabel *SMP_ACCESS* dan *SMP_PORTAL*, data pada *view* ditampilkan dalam bentuk tabel, *view*, *controller* dan *model* terlampir pada lampiran.

4.3.3. Mengelola Sepeda Motor

Untuk dapat mengelola sepeda motor digunakan *controller* *Motor.php* yang berisi kode program untuk menjalankan fungsi yang berisi pemanggilan *query*, jika *query* tersebut memiliki hasil maka hasil tersebut akan ditampilkan pada *view*.

```
public function index() {
    ...
    'motor' => $this->M_CallsQL-
>get_dataMotor("smp_motor")->result(),
    ...
}
```

Gambar 4.34 Motor Controller.

Baris kode program diatas akan menyimpan hasil *query* yang di eksekusi get_dataMotor pada tabel SMP_MOTOR kedalam variabel motor, data dari tabel SMP_MOTOR ditampilkan dalam bentuk tabel di dalam *view*.

```
public function input() {
    ...
    $this->M_CallSQL-
>input_data($insert,"smp_motor");
    ...
}
```

Gambar 4.35 Motor Controller-2.

Fungsi *input* digunakan untuk melakukan penambahan data pada tabel SMP_MOTOR, data motor harus di masukan melalui sistem monitor.

```
public function edit($id=null) {
    ...
    $this->M_CallSQL->update_data( [ 'MTR_ID'
=> $id],$insert,"smp_motor");
    ...
}
```

Gambar 4.36 Motor Controller-3.

Fungsi *edit* digunakan untuk melakukan perubahan data pada tabel SMP_MOTOR.

```
public function delete($id=null) {
    ...
    if ($this->M_CallSQL->delete("smp_motor",$data))
        ...
}
```

Gambar 4.37 Motor Controller-4.

Fungsi *delete* digunakan untuk melakukan penghapusan data pada tabel SMP_MOTOR, kode program *controller*, *view* dan *model* terlampir di dalam lampiran.

4.3.4. Mengelola Pengendara

Untuk dapat mengelola pengendara digunakan *controller* Pengendara.php yang berisi kode program untuk menjalankan fungsi yang berisi pemanggilan *query*, jika *query* tersebut memiliki hasil maka hasil tersebut akan ditampilkan pada *view*.

```
public function index() {
    ...
    'pengendara' => $this->M_CallsQL-
>get_data("smp_pengendara")->result(),
    ...
}
```

Gambar 4.38 Pengendara Controller.

Baris kode program diatas akan menyimpan hasil *query* yang di eksekusi *get_data* pada tabel SMP_PENGENDARA kedalam variabel pengendara, data dari tabel SMP_PENGENDARA ditampilkan dalam bentuk tabel di dalam *view*.

```
public function input() {
    ...
    $this->M_CallsQL-
>input_data($insert, "smp_pengendara");
    ...
}
```

Gambar 4.39 Pengendara Controller-2.

Fungsi *input* digunakan untuk melakukan penambahan data pada tabel SMP_PENGENDARA, data pengendara harus di masukan melalui sistem monitor.

```
public function edit($id=null) {
    ...
    $this->M_CallsSQL->update_data(['PGN_ID'
=> $id],$insert,"smp_pengendara");
    ...
}
```

Gambar 4.40 Pengendara Controller-3.

Fungsi *edit* digunakan untuk melakukan perubahan data pada tabel SMP_PENGENDARA.

```
public function delete($id=null) {
    ...
    if ($this->M_CallsSQL-
>delete("smp_pengendara", $data))
        ...
}
```

Gambar 4.41 Pengendara Controller-4.

Fungsi *delete* digunakan untuk melakukan penghapusan data pada tabel SMP_MOTOR, kode program controller, view dan model terlampir di dalam lampiran.

[Halaman ini sengaja dikosongkan]

BAB V

UJICOBA DAN EVALUASI

Pada bab ini dijelaskan tentang uji coba dan evaluasi dari implementasi yang telah dilakukan pada Tugas Akhir ini.

5.1 Lingkup Uji Coba

Pada lingkup uji coba, penulis menggunakan koneksiitas jaringan rumah “*Home Network*” untuk melakukan uji coba Sistem Parkir Sepeda Motor Cerdas, sistem monitor dan *control* dijalankan pada laptop dengan spesifikasi sebagai berikut:

Tabel 5.1 Lingkup Spesifikasi.

Komponen	Spesifikasi
CPU	Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz (8 CPUs), ~2.6GHz
Sistem Operasi	Windows 10 Pro N 64-bit (10.0, Build 18363)
Memori	12288MB RAM
Bios	BIOS Date: 04/22/16 14:05:07 Ver: 05.0000B
Penyimpanan	1 TB

Semua alat yang telah di implementasi sebelumnya terhubung pada jaringan rumah dengan menggunakan *WiFi*. Uji coba dilakukan menggunakan contoh kasus penggunaan Sistem Parkir Sepeda Motor Cerdas pada maket atau miniatur, hasil dari uji coba akan ditampilkan dalam bentuk monitoring melalui *website* yang disajikan dalam bentuk gambar.

5.2 Sistem Parkir Sepeda Motor Cerdas

Pengujian Sistem Parkir Sepeda Motor Cerdas dilakukan untuk mengetahui hasil dari implementasi, dimana pengujian dilakukan dalam bentuk kasus penggunaan pada miniatur yang telah dibuat.

1. Melakukan tap pada *Transceiver Motorcycle Device*.

Langkah awal pada Sistem Parkir Sepeda Motor Cerdas adalah melakukan tap pada *Transceiver Motorcycle Device*, pada kasus ini terdapat tiga *case* yang berbeda, diantara lain:

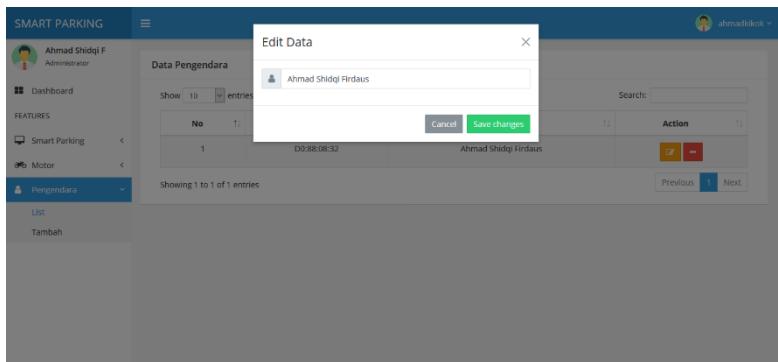
- Pengendara belum terdaftar.

Ketika pengendara belum terdaftar, *Transceiver Motorcycle Device* akan mengirimkan ID pengendara pada *website*, untuk dilakukan registrasi, pengendara yang sudah terdaftar dapat diubah atau dihapus, hasil uji coba adalah sebagai berikut

Gambar 5.1 Registrasi Pengendara.

No	T1	UID Card	T1	Nama	T1	Action
1		D0:88:08:32		Ahmad Shiddiq Firdaus		

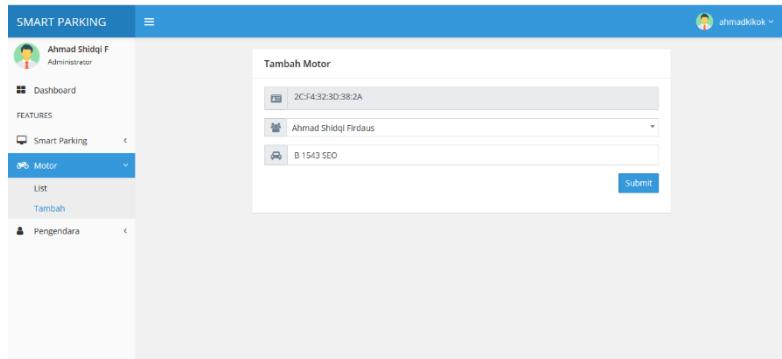
Gambar 5.2 Daftar Pengendara.



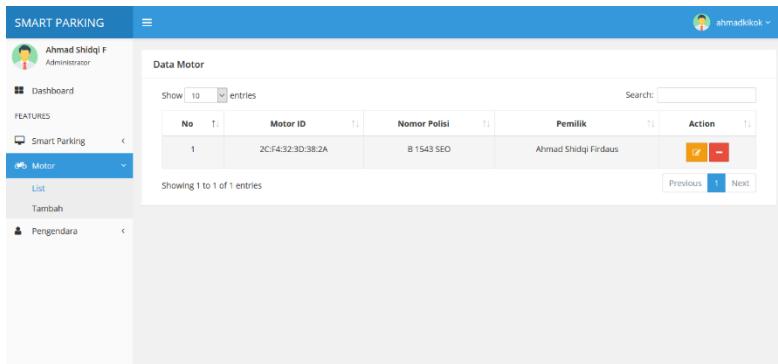
Gambar 5.3 Edit Pengendara.

b. Motor belum terdaftar.

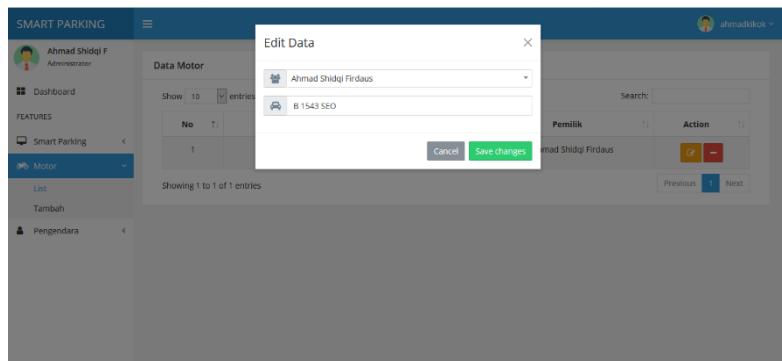
Ketika motor belum terdaftar, *Transceiver Motorcycle Device* akan mengirimkan *mac address Transceiver Motorcycle Device* ke pada *website*, untuk dilakukan registrasi, motor yang sudah terdaftar dapat diubah atau dihapus, hasil uji coba adalah sebagai berikut



Gambar 5.4 Registrasi Motor.



Gambar 5.5 Daftar Motor.



Gambar 5.6 Edit Motor.

c. Melakukan asosiasi.

Ketika motor dan pengendara sudah terdaftar, *Transceiver Motorcycle Device* akan mengirimkan data pengendara dan motor kedalam database untuk dilakukan asosiasi, asosiasi ditampilkan pada halaman monitor, hasil uji coba yang adalah sebagai berikut

ACS_ID	PGN_ID	MTR_ID	ACS_TIME_IN	ACS_TIME_OUT	IS_DEL
1	1	1	2020-05-19 19:43:55	0000-00-00 00:00:00	0

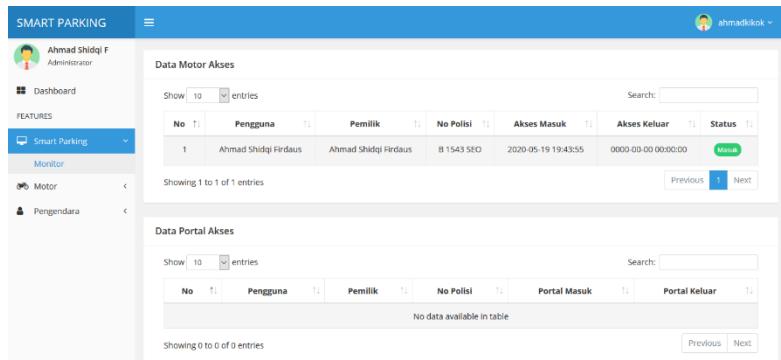
Gambar 5.7 Asosiasi Masuk TMD.

Ketika tap berhasil dilakukan, jika dilakukan monitoring melalui COM port Arduino IDE, maka akan tampil pesan sebagai berikut.



```
COM3
|Send
:rn 401
:c0 1, 481
:iwr 70 0
:wrcc 70 70 0
:ack 70
:rn 401
:c0 1, 481
:iwr 70 0
:wrcc 70 70 0
:ack 70
:rn 345
:c0 1, 345
Please tap your card to portal in...
< >
Autoscroll Show timestamp No line ending 115200 baud Clear output
```

Gambar 5.8 Asosiasi Masuk TMD IDE.



The screenshot shows the SMART PARKING application interface. On the left, there is a sidebar with user information (Ahmad Shidqi F, Administrator) and navigation links for Dashboard, FEATURES (Smart Parking Monitor, Motor, Pengendara), and a search bar. The main content area displays two tables:

- Data Motor Akses:**

No	Pengguna	Pemilik	No Polisi	Akses Masuk	Akses Keluar	Status
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B1543 SEO	2020-05-19 19:43:55	0000-00-00 00:00:00	Logout

Showing 1 to 1 of 1 entries
- Data Portal Akses:**

No	Pengguna	Pemilik	No Polisi	Portal Masuk	Portal Keluar
No data available in table					

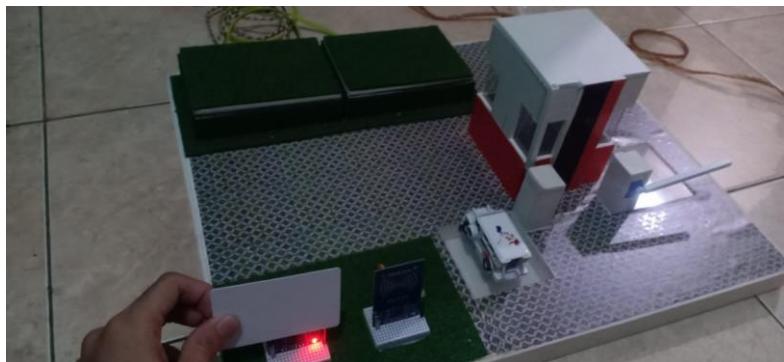
Showing 0 to 0 of 0 entries

Gambar 5.9 Asosiasi Masuk TMD Monitor.

Data Motor Akses berisi data pengendara yang terasosiasi dan sudah melakukan tap untuk, dengan Akses Masuk adalah waktu asosiasi kendaraan.

2. Melakukan tap pada *Smart Portal Device* Masuk.

Langkah selanjutnya adalah melakukan tap kartu pengguna pada *Smart Portal Device* Masuk, SMD akan melakukan pengecekan asosiasi pada database, jika ada data yang sudah terasosiasi, maka portal akan terbuka, hasil uji coba yang dilakukan adalah sebagai berikut



Gambar 5.10 SMD Masuk Tap.

Gambar 5.10 menunjukkan kartu yang sudah terasosiasi yang dilakukan pada tahap sebelumnya, portal akan terbuka dan data akan tercatat dalam *database*, hasil pencatatan adalah sebagai berikut

PTL_ID	ACS_ID	PTL_IN	PTL_OUT	IS_DEL
▶ 1	1	2020-05-19 19:48:03	0000-00-00 00:	0

Gambar 5.11 SMD Masuk *Database*.

Semua kendaraan yang sedang berada di dalam parkiran akan tercatat, dan akan ditampilkan sebagai berikut

The screenshot shows the SMART PARKING application interface. On the left, there is a sidebar with a user profile (Ahmad Shidqi F, Administrator), a dashboard icon, and a navigation menu under FEATURES: Smart Parking (selected), Monitor, Motor, and Pengendara. The main content area has two tables:

- Data Motor Akses:**

No	Pengguna	Pemilik	No Polisi	Akses Masuk	Akses Keluar	Status
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B 1543 SEO	2020-05-19 19:43:55	0000-00-00 00:00:00	Masuk

Showing 1 to 1 of 1 entries
- Data Portal Akses:**

No	Pengguna	Pemilik	No Polisi	Portal Masuk	Portal Keluar
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B 1543 SEO	2020-05-19 19:48:03	0000-00-00 00:00:00

Showing 1 to 1 of 1 entries

Gambar 5.12 SMD Masuk Monitor.

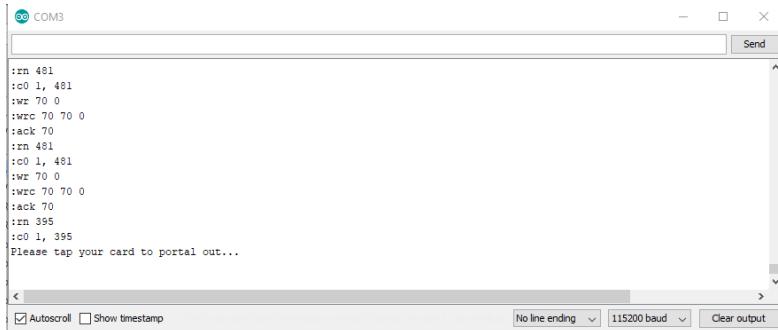
Data Portal Akses berisi data kendaraan yang sedang berada di dalam parkira, dengan Portal Masuk adalah waktu masuknya kendaraan.

3. Melakukan tap pada *Transceiver Motorcycle Device*.

Tap selanjutnya dilakukan untuk melakukan asosiasi jika pengendara ingin keluar dari parkiran, *Transceiver Motorcycle Device* akan melakukan cek pada data yang sebelumnya telah terisi di tahap sebelumnya, dan akan melakukan update sebagai bentuk asosiasi untuk keluar, hasil dari uji coba adalah sebagai berikut

ACS_ID	PGN_ID	MTR_ID	ACS_TIME_IN	ACS_TIME_OUT	IS_DEL
1	1	1	2020-05-19 19:43:55	2020-05-19 19:52:44	0

Gambar 5.13 Asosiasi Keluar TMD.



Gambar 5.14 Asosiasi Keluar TMD IDE.

Data Motor Akses						
No	Pengguna	Pemilik	No Polisi	Akses Masuk	Akses Keluar	Status
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B 1543 SEO	2020-05-19 19:43:55	2020-05-19 19:52:44	Keluar

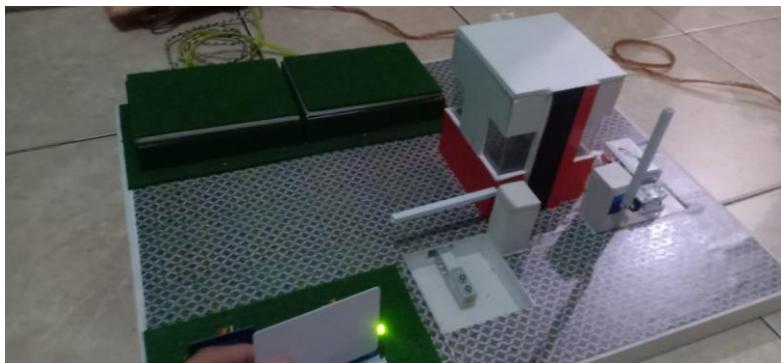
Data Portal Akses						
No	Pengguna	Pemilik	No Polisi	Portal Masuk	Portal Keluar	
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B 1543 SEO	2020-05-19 19:48:03	0000-00-00 00:00:00	

Gambar 5.15 Asosiasi Keluar TMD Monitor.

Akses Keluar adalah waktu asosiasi yang dilakukan pengendara untuk keluar dari parkiran, TMD melakukan *update* Akses Keluar ketika Akses Masuk dan Portal Masuk sudah terisi, menandakan pengendara sedang berada didalam parkiran.

4. Melakukan tap pada *Smart Portal Device* Keluar.

Langkah selanjutnya adalah melakukan tap kartu pengguna pada *Smart Portal Device* Keluar, SMD akan melakukan pengecekan asosiasi pada database, jika ada data yang sudah terasosiasi, masa portal akan terbuka, hasil uji coba yang dilakukan adalah sebagai berikut



Gambar 5.16 SMD Keluar Tap.

Gambar 5.16 menunjukkan kartu yang sudah terasosiasi yang dilakukan pada tahap sebelumnya, portal akan terbuka dan data akan tercatat dalam *database*, hasil pencatatan adalah sebagai berikut

	PTL_ID	ACS_ID	PTL_IN	PTL_OUT	IS_DEL
▶	1		1 2020-05-19 19:48:03	2020-05-19 19:	0

Gambar 5.17 SMD Keluar Database.

Semua kendaraan yang sedang berada di dalam parkiran akan tercatat, dan akan ditampilkan sebagai berikut

The screenshot shows the SMART PARKING application interface. On the left, there's a sidebar with a user profile (Ahmad Shidqi F, Administrator), a dashboard icon, and features like Smart Parking Monitor, Motor, and Pengendara. The main area has two tables:

- Data Motor Akses:**

No	Pengguna	Pemilik	No Polisi	Akses Masuk	Akses Keluar	Status
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B 1543 SEO	2020-05-19 19:43:55	2020-05-19 19:52:44	View
- Data Portal Akses:**

No	Pengguna	Pemilik	No Polisi	Portal Masuk	Portal Keluar
1	Ahmad Shidqi Firdaus	Ahmad Shidqi Firdaus	B 1543 SEO	2020-05-19 19:48:03	2020-05-19 19:53:47

Gambar 5.18 SMD Keluar Monitor.

Data Portal Akses berisi data kendaraan yang sedang berada di dalam parkira, dengan Portal Masuk adalah waktu masuknya kendaraan, Portal Keluar adalah waktu keluarnya kendaraan setelah melakukan asosiasi.

5. Menghapus data selesai.

Setelah semua tahap telah dilakukan, SMD akan secara otomatis melakukan update IS_DEL data pada *database* untuk menandakan bahwa proses telah berhasil

ACS_ID	PGN_ID	MTR_ID	ACS_TIME_IN	ACS_TIME_OUT	IS_DEL
1	1	1	2020-05-19 19:43:55	2020-05-19 19:52:44	1

Gambar 5.19 SMP_ACCESS Finished.

PTL_ID	ACS_ID	PTL_IN	PTL_OUT	IS_DEL
1	1	2020-05-19 19:48:03	2020-05-19 19:	1

Gambar 5.20 SMP_PORTAL Finished.

5.3 Performa

Uji coba performa pada *Transceiver Motorcycle Device* dilakukan pada jarak 4 meter dan 15 meter dari *acess point*, dengan tanpa adanya hambatan pada area selang jarak, didapatkan hasil pada jarak 4 meter sebagai berikut:

Tabel 5.2 Uji Coba TMD – 4 Meter.

dBm	Time WiFi (s)	Time Database (s)	Percobaan - Ke
-63 dBm	6	12	1
-63 dBm	6	20	2
-63 dBm	6	12	3
-63 dBm	6	12	4
-63 dBm	9	22	5
Rata-Rata	6.6	15.6	-

Dari hasil uji coba *Transceiver Motorcycle Device* untuk terhubung pada *WiFi* dan *database* didapatkan hasil seperti diatas, dimana dBm adalah kekuatan sinyal *WiFi* yang ditangkap oleh Arduino, dan didapatkan rata-rata waktu untuk terhubung dengan *WiFi* adalah 6.6 detik dan waktu untuk terhubung dengan *database* adalah 15.6 detik, waktu respon kartu saat melakukan tap pada jarak 4 meter didapatkan hasil seperti pada Tabel 5.3 Uji Coba TMD – 4 Meter Tap Respon.

Tabel 5.3 Uji Coba TMD – 4 Meter Tap Respon.

Jarak (m)	Insert Data (s)	Percobaan - Ke
4	0.217	1
4	0.229	2
4	0.178	3
4	0.217	4
4	0.219	5
4	0.298	6
4	0.198	7
4	0.208	8
4	0.338	9
4	0.208	10
Rata-Rata		0.231

Dari hasil uji coba tap kartu pengenal pada *Transceiver Motorcycle Device* didapatkan rata-rata 0.231 detik untuk mengirimkan data ke *database* melalui Arduino pada jarak 4 meter, waktu terhubung Arduino ke *access point* dan *database* didapatkan hasil seperti pada Tabel 5.4 Uji Coba TMD – 15 Meter.

Tabel 5.4 Uji Coba TMD – 15 Meter.

dBm	Time WiFi (s)	Time Database (s)	Percobaan - Ke
-87 dBm	9	16	1
-81 dBm	6	24	2
-83 dBm	9	34	3
-79 dBm	6	18	4
-82 dBm	9	44	5
Rata-Rata	7.8	27.2	-

Hasil dari pengujian pada jarak 15 meter didapatkan bahwa sinyal WiFi yang ditangkap oleh Arduino berbeda beda pada setiap percobannya, dengan rata-rata waktu terhubung ke WiFi adalah 7.8 detik dan waktu terhubung ke database adalah 27.2 detik, hasil dari uji coba tap pada *Transceiver Motorcycle Device* pada jarak 15 meter adalah sebagai berikut:

Tabel 5.5 Uji Coba TMD – 15 Meter Tap Respon.

Jarak (m)	Insert Data (s)	Percobaan - Ke
15	0.188	1
15	0.319	2
15	0.188	3
15	0.227	4
15	0.197	5
15	0.248	6
15	0.399	7
15	0.187	8
15	0.499	9
15	0.399	10
Rata-Rata		0.285

Dari hasil uji coba tap kartu pengenal pada *Transceiver Motorcycle Device* didapatkan rata-rata 0.285 detik untuk mengirimkan data ke database melalui Arduino pada jarak 15 meter, dari kedua uji coba tersebut didapat kesimpulan bahwa jarak antara *Transceiver Motorcycle Device* dengan *access point* mempengaruhi waktu respon pada Arduino untuk terhubung dengan WiFi, database, serta untuk pengiriman data dari Arduino ke database.

[*Halaman ini sengaja dikosongkan*]

BAB VI

KESIMPULAN

Pada bab ini akan dijelaskan kesimpulan dari hasil ujicoba yang telah dilakukan.

6.1 Kesimpulan

Kesimpulan yang diperoleh dari Tugas Akhir ini didasarkan pada hasil uji coba dan evaluasi. Kesimpulan Tugas Akhir ini adalah sebagai berikut:

1. Sistem Parkir Sepeda Motor Cerdas sudah di implementasikan dan semua fungsinya sudah berjalan dengan normal.
2. Jarak *Transceiver Motorcycle Device* dengan *access point* mempengaruhi waktu respon Arduino, dibuktikan dengan hasil uji coba performa dengan jarak 4 meter didapatkan rata-rata untuk Arduino terhubung dengan *WiFi* adalah 6.6 detik dan 15.6 detik untuk terhubung dengan *database*, sedangkan pada jarak 15 meter didapatkan rata-rata untuk Arduino terhubung dengan *WiFi* adalah adalah 7.8 detik dan 27.2 untuk terhubung dengan *database*.
3. Pengendara yang masuk atau keluar parkiran harus terasosiasi dengan kendaraannya.
4. Penambahan pengendara atau motor hanya bisa dilakukan melalui *website monitoring*.
5. Semua data yang diolah oleh arduino dapat dilakukan pemantauan pada *website monitoring*.

6.2 Saran

Saran yang dapat diberikan dari hasil uji coba dan evaluasi adalah sebagai berikut:

1. Melakukan uji coba dengan jumlah alat *Transceiver Motorcycle Device* yang lebih banyak dan kasus yang lebih banyak.
2. Melakukan uji coba dengan menggunakan router tambahan sebagai *WiFi Access Point*.
3. Penambahan alur peminjaman motor dengan menggunakan berkas secara *online* yang di setujui oleh pemilik kendaraan.

[*Halaman ini sengaja dikosongkan*]

DAFTAR PUSTAKA

- [1] William Stallings, “Operating Systems Internals and Design Principles,” in *Operating Systems Internals and Design Principles*, 7th ed., Prentice Hall, Ed. 2012, hal. 574–604.
- [2] A. I. Ramadhan, D. Triyanto, dan I. Ruslianto, “Pengembangan Sistem Parkir Otomatis Menggunakan Arduino Mega 2560 Berbasis Website,” *J. Coding, Sist. Komput. Untan*, vol. 04, no. 2, hal. 184–194, 2016, doi: 10.4234/jjoffamiliysociology.28.250.
- [3] A. Mappa, “SISTEM PARKIR CERDAS SEDERHANA BERBASIS ARDUINO MEGA 2560 Rev3,” *Electro Luceat*, 2018, doi: 10.32531/jelekn.v4i1.79.
- [4] N. M. Carl Hamacher, Zvonko G. Vranesic, Safwat G. Zaky, “Computer Organization and Embedded Systems,” in *Computer Organization and Embedded Systems*, 6th ed., McGraw-Hill, Ed. 2012, hal. 385–420.
- [5] A. Khanna dan R. Anand, “IoT based smart parking system,” in *2016 International Conference on Internet of Things and Applications, IOTA 2016*, 2016, doi: 10.1109/IOTA.2016.7562735.
- [6] Sublime Text, “Sublime Text 3 Tutorial.” <https://www.sublimetext.com/support> (diakses Okt 18, 2019).
- [7] Navicat, “Navicat Manual.” https://www.navicat.com/manual/online_manual_new/en/navicat/win_manual/ (diakses Jan 05, 2020).
- [8] Muhammad Aqib, “Use an Arduino and an ESP8266 module to Control a Servo Motor through a Webpage.”

- <https://maker.pro/arduino/tutorial/how-to-make-a-web-controlled-servo-with-arduino-and-esp8266> (diakses Des 17, 2019).
- [9] K. V. Kai “Oswald” Seidler, “XAMPP.” <https://www.apachefriends.org/index.html> (diakses Mei 22, 2020).
- [10] Hmeftunitirta, “Load Cell Sensor.” <https://www.hmeftuntirta.com/2018/06/memahami-sensor-berat-load-cell/> (diakses Mei 22, 2020).
- [11] Y. N. Ewnetu Kefale, Meklit Motbainor, “IoT Based Parking System.” <https://create.arduino.cc/projecthub/102513/iot-based-parking-system-b6a947> (diakses Des 17, 2019).
- [12] Cory Guynn, “RFID Scanner – WeMos, RC522, Homie MQTT & Node-RED.” <http://www.internetoflego.com/rfid-scanner-wemos-rc522-mqtt/> (diakses Des 17, 2019).
- [13] Components 101, “Servo Motor SG-90.” <https://components101.com/servo-motor-basics-pinout-datasheet> (diakses Mei 22, 2020).
- [14] Components 101, “RC522 RFID Module.” <https://components101.com/wireless/rc522-rfid-module> (diakses Mei 22, 2020).
- [15] Codeigniter, “CodeIgniter 3.” <https://codeigniter.com/> (diakses Mei 22, 2020).
- [16] Arduino, “Arduino Tutorial.” <https://www.arduino.cc/en/Tutorial/HomePage?from=Main.Tutorials> (diakses Des 30, 2019).
- [17] Alselectro, “WEMOS D1 R1.” <https://alselectro.wordpress.com/2018/04/14/wifi->

esp8266-development-board-wemos-d1/ (diakses Mei 22, 2020).

[*Halaman ini sengaja dikosongkan*]

LAMPIRAN

1. Kode Program SQL

```
-- -----
-- Table structure for smp_access
-- ----

DROP TABLE IF EXISTS `smp_access`;
CREATE TABLE `smp_access` (
  `ACS_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PGN_ID` int(11) NOT NULL,
  `MTR_ID` int(11) NOT NULL,
  `ACS_TIME_IN` datetime(0) NOT NULL DEFAULT CURRENT_TIMESTAMP(),
  `ACS_TIME_OUT` datetime(0) NOT NULL,
  `IS_DEL` int(1) NOT NULL DEFAULT 0,
  PRIMARY KEY (`ACS_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;

-- -----
-- Table structure for smp_motor
-- ----

DROP TABLE IF EXISTS `smp_motor`;
CREATE TABLE `smp_motor` (
  `MTR_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PGN_ID` int(11) NULL DEFAULT NULL,
  `MTR_HW_ID` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
  `MTR_NO_POL` varchar(15) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
  PRIMARY KEY (`MTR_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;

-- -----
-- Table structure for smp_pengendara
-- ----

DROP TABLE IF EXISTS `smp_pengendara`;
CREATE TABLE `smp_pengendara` (
```

```

`PGN_ID` int(11) NOT NULL AUTO_INCREMENT,
`PGN_CARD_ID` varchar(50) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
`PGN_NAMA` varchar(25) CHARACTER SET latin1 COLLATE latin1_swedish_ci NULL DEFAULT NULL,
PRIMARY KEY (`PGN_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;

-- -----
-- Table structure for smp_portal
-- -----
DROP TABLE IF EXISTS `smp_portal`;
CREATE TABLE `smp_portal` (
`PTL_ID` int(11) NOT NULL AUTO_INCREMENT,
`ACS_ID` int(11) NOT NULL,
`PTL_IN` datetime(0) NOT NULL DEFAULT CURRENT_TIMESTAMP(0),
`PTL_OUT` datetime(0) NOT NULL,
`IS_DEL` int(11) NOT NULL DEFAULT 0,
PRIMARY KEY (`PTL_ID`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 1;

ALTER TABLE smp_access
ADD CONSTRAINT `FK_RELATIONSHIP_1` FOREIGN KEY (`PGN_ID`) REFERENCES `smp_pengendara`(`PGN_ID`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `FK_RELATIONSHIP_2` FOREIGN KEY (`MTR_ID`) REFERENCES `smp_motor`(`MTR_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE smp_portal
ADD CONSTRAINT `FK_RELATIONSHIP_3` FOREIGN KEY (`ACS_ID`) REFERENCES `smp_access`(`ACS_ID`) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE smp_motor

```

```
ADD CONSTRAINT `FK_RELATIONSHIP_4` FOREIGN
KEY (`PGN_ID`) REFERENCES `smp_pengendara`
(`PGN_ID`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

2. Kode Program Transceiver Motorcycle Device

```
#include <SPI.h>
#include <MFRC522.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>

#define SS_PIN D10 //Define start mrfc522
pin
#define RST_PIN D9 //Define reset mrfc522
pin
//#define BUZZER D15 //Define buzzer pin

// ----- MRFC Configuration
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create
MFRC522 instance.

// ----- WiFi Configuration
char ssid[] = "Adek Kecil Queena";
// Network Name
char pass[] = "rumahparung141291"; // Network Password

String BASE_URL =
"http://192.168.100.38/codeigniter/iot_ta/";

// ----- MySQL Configuration
char user[] = "userta";
// MySQL user
char password[] = "userta";
// MySQL password
```

```
WiFiClient client;
IPAddress server_addr(192, 168, 100, 38);
// MySQL server IP
MySQL_Connection conn((Client *)&client);
MySQL_Cursor cur = MySQL_Cursor(&conn);

// ----- Function Define Pin
Output
void connectPin() {
    mfrc522.PCD_Init(); //Initiate MFRC522
// pinMode(BUZZER, OUTPUT);
// noTone(BUZZER);
}

// ----- Function Connect to WiFi
and MySQL
void connectWiFi(){
// WiFi Connect
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) {
    delay(200);
    Serial.print(".");
}
// tone(BUZZER, 500);
// delay(300);
// noTone(BUZZER);
Serial.println("");
Serial.println("WiFi Connected");
Serial.print("Assigned IP: ");
Serial.println(WiFi.localIP());
//MySQL Connect
Serial.println("Connecting to database");
while (conn.connect(server_addr, 3306,
user, password) != true) {
    delay(200);
    Serial.print (" . ");
}
// tone(BUZZER, 500);
```

```
// delay(300);
// noTone(BUZZER);
Serial.println("Connected to SQL Server!");
Serial.println("");
}

// ----- Setup Begin
void setup() {
    Serial.begin(115200);
    SPI.begin();
    connectPin();
    connectWiFi();
}

// ----- Setup Loop
void loop() {

    // Look for new cards or Select one of
    the cards
    if ( ! mfrc522.PICC_IsNewCardPresent() ||

        ! mfrc522.PICC_ReadCardSerial())
        return;

    Serial.print(F("PICC type: "));
    MFRC522::PICC_Type piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);

    Serial.println(mfrc522.PICC_GetTypeName(piccT
ype));
    //
    // Check is the PICC of Classic MIFARE
    type
    if (piccType !=

MFRC522::PICC_TYPE_MIFARE_MINI && piccType !=

MFRC522::PICC_TYPE_MIFARE_1K && piccType !=

MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of
    type MIFARE Classic."));
    return;
}
```

```

    // Store Hex and Dec ID
    String hex, dec, PGN_ID, ACS_ID, MTR_ID,
ACSMTR_ID, ACS_IN, ACS_OUT, PTL_IN = "";
    String mac_address = WiFi.macAddress();
    for (byte i = 0; i < mfrc522.uid.size;
i++) {
    hex.concat(String(mfrc522.uid.uidByte[i] <
0x10 ? " 0" : " "));
    hex.concat(String(mfrc522.uid.uidByte[i],
HEX));
    dec.concat(String(mfrc522.uid.uidByte[i] <
0x10 ? " 0" : " "));
    dec.concat(String(mfrc522.uid.uidByte[i],
DEC));
}
hex.toUpperCase(); hex =
hex.substring(1); hex.replace(" ", ":" );
dec.toUpperCase(); dec =
dec.substring(1);

messageReadInput(hex,dec);

MTR_ID = getMtrId(mac_address);
PGN_ID = getPgnId(hex);
ACS_ID = getAcsId(PGN_ID,MTR_ID);
ACS_IN = getAcsIn(PGN_ID);
ACS_OUT = getAcsOut(PGN_ID);
PTL_IN = getPtlIn(ACS_ID);

//Assume acs_id one only
if ( ACS_ID == NULL && PGN_ID != NULL &&
MTR_ID != NULL) {
    ACSMTR_ID = getAcsMtrId(MTR_ID);
    //Check if mtr_id already used
    if (MTR_ID == ACSMTR_ID) {

```

```
        Serial.println("Data already
exists...");
//          tone(BUZZER, 1000);
//          delay(300);
//          noTone(BUZZER);
    } else {
        //If no data on smp_access, insert
data to smp_access
        insertMasuk(PGN_ID,MTR_ID);
//          tone(BUZZER, 500);
//          delay(300);
//          noTone(BUZZER);
    }
} else if( ACS_ID != NULL && PTL_IN != NULL && PTL_IN != "0000-00-00 00:00:00" && ACS_IN != "0000-00-00 00:00:00" && ACS_OUT == "0000-00-00 00:00:00") {
    //Update data acs_time_out
    updateKeluar(ACS_ID);
//          tone(BUZZER, 500);
//          delay(300);
//          noTone(BUZZER);
} else if( ACS_ID != NULL && PGN_ID != NULL && MTR_ID != NULL && ACS_IN != "0000-00-00 00:00:00" && ACS_OUT == "0000-00-00 00:00:00" ) {
    Serial.println("Please tap your card
to portal in...");
```

} else if(ACS_ID != NULL && PGN_ID != NULL && MTR_ID != NULL && ACS_IN != "0000-00-00 00:00:00" && ACS_OUT != "0000-00-00 00:00:00" && PTL_IN != NULL && PTL_IN != "0000-00-00 00:00:00") {

```
    Serial.println("Please tap your card
to portal out...");
```

} if(PGN_ID == NULL) {
 parseUIDPengendara(hex);
} if(MTR_ID == NULL) {
 parseUIDMotor(mac_address);
}

```
    Serial.println();
    delay(500);
}

// ----- Membaca Hasil Input Kartu
void messageReadInput(String hex, String dec) {
    Serial.println("The NUID tag is:");
    Serial.print("In hex: ");
    Serial.println(hex);
    Serial.print("In dec: ");
    Serial.println(dec);
}

// ----- Mengambil ID pengendara yang melakukan tap - TABLE smp_pengendara
String getPgnId(String hex) {
    //Cek PGN_ID sudah melakukan tap atau belum pada TABLE smp_access
    String PGN_ID = ""; //Store Variable
    MySQL_Cursor *cur_pengguna = new MySQL_Cursor(&conn);
    cur_pengguna->execute(("SELECT * FROM ta_smp.smp_pengendara WHERE PGN_CARD_ID = ''" + hex + "'").c_str());
    column_names *col_pengguna =
    cur_pengguna->get_columns();
    row_values *row_pengguna = NULL;
    do {
        row_pengguna = cur_pengguna-
>get_next_row();
        if (row_pengguna != NULL) {
            for (int f = 0; f < col_pengguna-
>num_fields; f++) {
                PGN_ID = row_pengguna->values[0];
            }
        }
    } while (row_pengguna != NULL);
    delete cur_pengguna;
}
```

```
        return PGN_ID;
    }

// ----- Mengambil ID motor -
TABLE smp_motor
String getMtrId(String mac_address){
    //Cek PGN_ID sudah melakukan tap atau
belum pada TABLE smp_access
    String MTR_ID = ""; //Store Variable
    MySQL_Cursor *cur_motor = new
MySQL_Cursor(&conn);
    cur_motor->execute(("SELECT * FROM
ta_smp.smp_motor WHERE MTR_HW_ID = '" +
mac_address + "'").c_str());
    column_names *col_motor = cur_motor-
>get_columns();
    row_values *row_motor = NULL;
    do {
        row_motor = cur_motor->get_next_row();
        if (row_motor != NULL) {
            for (int f = 0; f < col_motor-
>num_fields; f++) {
                MTR_ID = row_motor->values[0];
            }
        }
    } while (row_motor != NULL);
    delete cur_motor;
    return MTR_ID;
}

// ----- Mengambil ACS_ID - TABLE
smp_access
String getAcsId(String result_id, String
result_mac){
    String ACS_ID = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE PGN_ID = '" +
```

```

result_id + "' AND MTR_ID = '" + result_mac +
"' AND IS_DEL = 0").c_str());
    column_names *col_access = cur_access-
>get_columns();
    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {
                //Store values id to id_card
                ACS_ID = row_access->values[0];
            }
        }
    } while (row_access != NULL);
    delete cur_access;
    return ACS_ID;
}

// ----- Mengambil MTR_ID - TABLE
smp_access
String getAcsMtrId(String result_mac){
    //Start select MTR_ID at SMP_ACCESS from
MySQL to prevent duplicate MAC_ADDRESS tap
    String ACSMTR_ID = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE MTR_ID = '" +
result_mac + "' AND IS_DEL = 0 ").c_str());
    column_names *col_access = cur_access-
>get_columns();
    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {

```

```

        //Store values id to id_card
        ACSMTR_ID = row_access->values[2];
    }
}
} while (row_access != NULL);
delete cur_access;
return ACSMTR_ID;
}

// ----- Mengambil ACS_IN - TABLE
smp_access
String getAcsIn(String id){
    String ACS_IN = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE PGN_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_access = cur_access-
>get_columns();
    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {
                //Store values id to id_card
                ACS_IN = row_access->values[3];
            }
        }
    } while (row_access != NULL);
    delete cur_access;
    return ACS_IN;
}

// ----- Mengambil ACS_OUT -
TABLE smp_access
String getAcsOut(String id){
    String ACS_OUT = ""; //Store Variable

```

```

    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE PGN_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_access = cur_access-
>get_columns();
    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {
                //Store values id to id_card
                ACS_OUT = row_access->values[4];
            }
        }
    } while (row_access != NULL);
    delete cur_access;
    return ACS_OUT;
}

// ----- Mengambil PTL_IN - TABLE
smp_portal
String getPtlIn(String id){
    String PTL_IN = ""; //Store Variable
    MySQL_Cursor *cur_portal = new
MySQL_Cursor(&conn);
    cur_portal->execute(("SELECT * FROM
ta_smp.smp_portal WHERE ACS_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_portal = cur_portal-
>get_columns();
    row_values *row_portal = NULL;
    do {
        row_portal = cur_portal-
>get_next_row();
        if (row_portal != NULL) {

```

```
        for (int f = 0; f < col_portal->num_fields; f++) {
            //Store values id to id_card
            PTL_IN = row_portal->values[2];
        }
    }
} while (row_portal != NULL);
delete cur_portal;
return PTL_IN;
}

// ----- Insert Access ke Access
Masuk - TABLE smp_access
void insertMasuk(String result_id, String
result_mac){
    String temp_update = String("INSERT INTO
ta_smp.smp_access (PGN_ID, MTR_ID) VALUES
('"+ result_id +"', '"+ result_mac +"')");
    MySQL_Cursor *update_query = new
MySQL_Cursor(&conn);
    update_query-
>execute(temp_update.c_str());
    delete update_query;
}

// ----- Update Access ke Portal
Masuk - TABLE smp_portal
void updateKeluar(String id){
    MySQL_Cursor *update_query = new
MySQL_Cursor(&conn);
    update_query->execute(("UPDATE
ta_smp.smp_access SET
ACS_TIME_OUT=CURRENT_TIMESTAMP() WHERE ACS_ID
= "+ id ).c_str());
    delete update_query;
}

// ----- Parse data pengendara
ke website
void parseUIDPengendara(String hex) {
```

```
//Data pengendara not found, post data to
pengendara page and add pengendara
Serial.println("Parsing card_id to
pengendara page...");  

HTTPClient http; //Declare object of
class HTTPClient
String UIDresultSend, postData;
UIDresultSend = hex;
//Post Data
postData = "UIDresult=" + UIDresultSend;
http.begin(BASE_URL +
"pengendara/parse_uid"); //Specify request
destination
http.addHeader("Content-Type",
"application/x-www-form-urlencoded");
//Specify content-type header
int httpCode = http.POST(postData);
//Send the request
String payload = http.getString();
//Get the response payload
http.end(); //Close connection
}  
  

// ----- Parse data motor ke
website
void parseUIDMotor(String mac_address) {
//Data motor not found, post data to
motor page and add motor
Serial.println("Parsing mtr_id to motor
page...");  

HTTPClient http; //Declare object of
class HTTPClient
String UIDresultSend, postData;
UIDresultSend = mac_address;
postData = "UIDresult=" + UIDresultSend;
http.begin(BASE_URL + "motor/parse_uid");
//Specify request destination
http.addHeader("Content-Type",
"application/x-www-form-urlencoded");
//Specify content-type header
```

```

        int httpCode = http.POST(postData);
    //Send the request
    String payload = http.getString();
    //Get the response payload
    http.end(); //Close connection
}

```

3. Kode Program *Smart Portal Device* Masuk

```

#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <HX711.h>

#define SS_PIN D10 //Define start mrfc522
pin
#define RST_PIN D9 //Define reset mrfc522
pin
#define LED_G D14 //Define green LED pin
// #define LED_R D4 //Define red LED pin
#define BUZZER D15 //Define buzzer pin
#define SERVO D8 //Define servo pin
#define DOUT D2 //Define hx711 dout pin
#define SCK D3 //Define hx711 sck pin

// ----- HX711 Configuration
HX711 scale(DOUT, SCK);
float calibration_factor = -7050;
int GRAM;

// ----- MRFC Configuration
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create
MFRC522 instance.

```

```
// ----- MotorServo Configuration
Servo myServo; //Define servo name
int sServo = 0; //Define status servo open

// ----- WiFi Configuration
char ssid[] = "Adek Kecil Queena";
// Network Name
char pass[] = "rumahparung141291"; // Network Password

// ----- MySQL Configuration
char user[] = "userta";
// MySQL user
char password[] = "userta";
// MySQL password
WiFiClient client;
IPAddress server_addr(192, 168, 100, 38);
// MySQL server IP
MySQL_Connection conn((Client *)&client);

// ----- Function Define Pin Output
void connectPin(){
    scale.set_scale(); //set scale pembacaan berat
    scale.tare(); //auto zero / mengenolkan pembacaan berat
    mfrc522.PCD_Init(); //Initiate MFRC522
    myServo.attach(SERVO); //servo pin
    myServo.write(90); //servo start position
    pinMode(LED_G, OUTPUT);
    // pinMode(LED_R, OUTPUT);
    pinMode(BUZZER, OUTPUT);
    noTone(BUZZER);
}

// ----- Function Connect to WiFi and MySQL
void connectWiFi(){
    // WiFi Connect
```

```
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
while (WiFi.status() != WL_CONNECTED) {
    delay(200);
    Serial.print(".");
}
tone(BUZZER, 500);
delay(300);
noTone(BUZZER);
Serial.println("");
Serial.println("WiFi Connected");
Serial.print("Assigned IP: ");
Serial.println(WiFi.localIP());
//MySQL Connect
Serial.println("Connecting to database");
while (conn.connect(server_addr, 3306,
user, password) != true) {
    delay(200);
    Serial.print (" .");
}
tone(BUZZER, 500);
delay(300);
noTone(BUZZER);
Serial.println("Connected to SQL Server!");
Serial.println("");
}

/*
Program Dimulai Dari Fungsi di Bawah
*/
// ----- Setup Begin
void setup() {
    Serial.begin(115200);
    SPI.begin();
    connectPin();
    connectWIFI();
}
```

```

// ----- Setup Loop
void loop() {

    // Checking weight sensor
    scale.set_scale(calibration_factor);
    GRAM = scale.get_units(), 1;
    Serial.print("Reading: ");
    Serial.println(GRAM);

    if ( GRAM != 0 && sServo == 0 ){
        // Look for new cards or Select one of
        the cards
        if ( ! mfrc522.PICC_IsNewCardPresent() ||
        ! mfrc522.PICC_ReadCardSerial())
            return;

        Serial.print(F("PICC type: "));
        MFRC522::PICC_Type piccType =
        mfrc522.PICC_GetType(mfrc522.uid.sak);

        Serial.println(mfrc522.PICC_GetTypeName(piccType));
        //
        // Check is the PICC of Classic MIFARE
        type
        // if (piccType !=
        MFRC522::PICC_TYPE_MIFARE_MINI && piccType !=
        MFRC522::PICC_TYPE_MIFARE_1K && piccType !=
        MFRC522::PICC_TYPE_MIFARE_4K) {
        //     Serial.println(F("Your tag is not of
        type MIFARE Classic."));
        //     return;
        // }

        // Store Hex and Dec ID
        String hex, dec, PGN_ID, ACS_ID, ACS_IN =
        "";
        for (byte i = 0; i < mfrc522.uid.size;
        i++) {

```

```
hex.concat(String(mfrc522.uid.uidByte[i] <
0x10 ? " 0" : " "));

hex.concat(String(mfrc522.uid.uidByte[i],
HEX));

dec.concat(String(mfrc522.uid.uidByte[i] <
0x10 ? " 0" : " "));

dec.concat(String(mfrc522.uid.uidByte[i],
DEC));
}

hex.toUpperCase(); hex =
hex.substring(1); hex.replace(" ", ":");

dec.toUpperCase(); dec =
dec.substring(1);

messageReadInput(GRAM,hex,dec);
PGN_ID = getPgnId(hex); //Mengambil
PGN_ID dari table smp_pengendara
ACS_ID = getAcsId(PGN_ID); //Mengambil
ACS_ID dari table smp_access
ACS_IN = getAcsIn(ACS_ID); //Mengambil
ACS_TIME_IN dari table smp_access

Serial.print("Message : ");
if (PGN_ID != NULL && ACS_ID != NULL &&
ACS_IN != "0000-00-00 00:00:00") { //Cek jika
ada pengendara yang sudah melakukan tap pada
Motor
    if (getPtlAcs(ACS_ID) == NULL) { //Cek
apa pengendara sudah ada
        // Servo Open and Close
        insertMasuk(ACS_ID); //Insert data
ke smp_portal
        Serial.println("Authorized
access");
        delay(500);
        digitalWrite(LED_G, HIGH);
```

```
        tone(BUZZER, 500);
        delay(300);
        noTone(BUZZER);
        myServo.write(180);
        sServo = 1;
    } else {
        Serial.println("Id already
exist.");
        delay(500);
        tone(BUZZER, 300);
        delay(1000);
        noTone(BUZZER);
    }
    Serial.println();
} else {
    Serial.println("Access denied");
    Serial.println();
    delay(500);
    tone(BUZZER, 300);
    delay(1000);
    noTone(BUZZER);
}
} else if (GRAM == 0 && sServo == 1){
    sServo = 0;
    delay(1000);
    digitalWrite(LED_G, LOW);
    myServo.write(90);
}
}

// ----- Membaca Hasil Input
Kartu
void messageReadInput(int GRAM, String hex,
String dec){
    Serial.print("Reading: ");
    Serial.print(GRAM);
    Serial.println(" Gram");
    Serial.println("The NUID tag is:");
    Serial.print("In hex: ");
    Serial.println(hex);
```

```

        Serial.print("In dec: ");
        Serial.println(dec);
    }

// ----- Mengambil ID pengendara
yang melakukan tap - TABLE smp_pengendara
String getPgnId(String hex){
    //Cek PGN_ID sudah melakukan tap atau
belum pada TABLE smp_access
    String PGN_ID = ""; //Store Variable
    MySQL_Cursor *cur_pengguna = new
MySQL_Cursor(&conn);
    cur_pengguna->execute(("SELECT * FROM
ta_smp.smp_pengendara WHERE PGN_CARD_ID = '"
+ hex + "'").c_str());
    column_names *col_pengguna =
cur_pengguna->get_columns();
    row_values *row_pengguna = NULL;
    do {
        row_pengguna = cur_pengguna-
>get_next_row();
        if (row_pengguna != NULL) {
            for (int f = 0; f < col_pengguna-
>num_fields; f++) {
                PGN_ID = row_pengguna->values[0];
            }
        }
    } while (row_pengguna != NULL);
    delete cur_pengguna;
    return PGN_ID;
}

// ----- Mengambil ACS_ID - TABLE
smp_access
String getAcsId(String id){
    String ACS_ID = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);

```

```

        cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE PGN_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
        column_names *col_access = cur_access-
>get_columns();
        row_values *row_access = NULL;
        do {
            row_access = cur_access-
>get_next_row();
            if (row_access != NULL) {
                for (int f = 0; f < col_access-
>num_fields; f++) {
                    //Store values id to id_card
                    ACS_ID = row_access->values[0];
                }
            }
        } while (row_access != NULL);
        delete cur_access;
        return ACS_ID;
    }

// ----- Mengambil ACS_IN - TABLE
smp_access
String getAcsIn(String id){
    String ACS_IN = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE ACS_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_access = cur_access-
>get_columns();
    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {
                //Store values id to id_card

```

```

        ACS_IN = row_access->values[3];
    }
}
} while (row_access != NULL);
delete cur_access;
return ACS_IN;
}

// ----- Mengambil PTI_ID - TABLE
smp_portal
String getPtlAcs(String id){
    String PTL_ID = ""; //Store Variable
    MySQL_Cursor *cur_portal = new
MySQL_Cursor(&conn);
    cur_portal->execute(("SELECT * FROM
ta_smp.smp_portal WHERE ACS_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_portal = cur_portal-
>get_columns();
    row_values *row_portal = NULL;
    do {
        row_portal = cur_portal-
>get_next_row();
        if (row_portal != NULL) {
            for (int f = 0; f < col_portal-
>num_fields; f++) {
                //Store values id to id_card
                PTL_ID = row_portal->values[0];
            }
        }
    } while (row_portal != NULL);
    delete cur_portal;
    return PTL_ID;
}

// ----- Insert Access ke Portal
Masuk - TABLE smp_portal
void insertMasuk(String id){

```

```

        String temp_update = String("INSERT
INTO ta_smp.smp_portal (ACS_ID) VALUES ('"+
id +')");
        MySQL_Cursor *update_query = new
MySQL_Cursor(&conn);
        update_query-
>execute(temp_update.c_str());
        delete update_query;
}

```

4. Kode Program *Smart Portal Device Keluar*

```

#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
#include <MySQL_Connection.h>
#include <MySQL_Cursor.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266HTTPClient.h>
#include <HX711.h>

#define SS_PIN D10 //Define start mrfc522
pin
#define RST_PIN D9 //Define reset mrfc522
pin
#define LED_G D14 //Define green LED pin
//#define LED_R D4 //Define red LED pin
#define BUZZER D15 //Define buzzer pin
#define SERVO D8 //Define servo pin
#define DOUT D2 //Define hx711 dout pin
#define SCK D3 //Define hx711 sck pin

// ----- HX711 Configuration
HX711 scale(DOUT, SCK);
float calibration_factor = -7050;
int GRAM;

// ----- MRFC Configuration

```

```
MFRC522 mfrc522(SS_PIN, RST_PIN); // Create
MFRC522 instance.

// ----- MotorServo Configuration
Servo myServo; //Define servo name
int sServo = 0; //Define status servo open

// ----- WiFi Configuration
char ssid[] = "Adek Kecil Queena";
// Network Name
char pass[] = "rumahparung141291"; // Network Password

// ----- MySQL Configuration
char user[] = "userta";
// MySQL user
char password[] = "userta";
// MySQL password
WiFiClient client;
IPAddress server_addr(192, 168, 100, 38);
// MySQL server IP
MySQL_Connection conn((Client *)&client);

// ----- Function Define Pin
Output
void connectPin(){
    scale.set_scale(); //set scale pembacaan berat
    scale.tare(); //auto zero / mengenolkan pembacaan berat
    mfrc522.PCD_Init(); //Initiate MFRC522
    myServo.attach(SERVO); //servo pin
    myServo.write(90); //servo start position
    pinMode(LED_G, OUTPUT);
    // pinMode(LED_R, OUTPUT);
    pinMode(BUZZER, OUTPUT);
    noTone(BUZZER);
}
```

```
// ----- Function Connect to WiFi  
and MySQL  
void connectWiFi() {  
    // WiFi Connect  
    Serial.print("Connecting to ");  
    Serial.println(ssid);  
    WiFi.begin(ssid, pass);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(200);  
        Serial.print(".");  
    }  
    tone(BUZZER, 500);  
    delay(300);  
    noTone(BUZZER);  
    Serial.println("");  
    Serial.println("WiFi Connected");  
    Serial.print("Assigned IP: ");  
    Serial.println(WiFi.localIP());  
    //MySQL Connect  
    Serial.println("Connecting to database");  
    while (conn.connect(server_addr, 3306,  
user, password) != true) {  
        delay(200);  
        Serial.print (" .");  
    }  
    tone(BUZZER, 500);  
    delay(300);  
    noTone(BUZZER);  
    Serial.println("Connected to SQL Server!");  
    Serial.println("");  
}  
  
/*  
Program Dimulai Dari Fungsi di Bawah  
*/  
  
// ----- Setup Begin  
void setup() {  
    Serial.begin(115200);  
    SPI.begin();
```

```
connectPin();
connectWiFi();
}

// ----- Setup Loop
void loop() {

    // Checking weight sensor
    scale.set_scale(calibration_factor);
    GRAM = scale.get_units(), 1;
    Serial.print("Reading: ");
    Serial.println(GRAM);

    if ( GRAM != 0 && sServo == 0 ) {
        // Look for new cards or Select one of
        the cards
        if ( ! mfrc522.PICC_IsNewCardPresent() ||

        ! mfrc522.PICC_ReadCardSerial() )
            return;

        Serial.print(F("PICC type: "));
        MFRC522::PICC_Type piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);

        Serial.println(mfrc522.PICC_GetTypeName(piccT
ype));
        //
        // Check is the PICC of Classic MIFARE
        type
        // if (piccType !=
MFRC522::PICC_TYPE_MIFARE_MINI && piccType !=
MFRC522::PICC_TYPE_MIFARE_1K && piccType !=
MFRC522::PICC_TYPE_MIFARE_4K) {
        //     Serial.println(F("Your tag is not of
        type MIFARE Classic."));
        //     return;
        // }

        // Store Hex and Dec ID
```

```

        String hex, dec, PGN_ID, ACS_ID, ACS_IN,
ACS_OUT, PTL_IN, PTL_ID, PTL_OUT = "";
    for (byte i = 0; i < mfrc522.uid.size;
i++) {

hex.concat(String(mfrc522.uid.uidByte[i] <
0x10 ? " 0" : " "));

hex.concat(String(mfrc522.uid.uidByte[i],
HEX));

dec.concat(String(mfrc522.uid.uidByte[i] <
0x10 ? " 0" : " "));

dec.concat(String(mfrc522.uid.uidByte[i],
DEC));
}

hex.toUpperCase(); hex =
hex.substring(1); hex.replace(" ", ":");

dec.toUpperCase(); dec =
dec.substring(1);

messageReadInput(GRAM,hex,dec);
PGN_ID = getPgnId(hex); //Mengambil
PGN_ID dari table smp_pengendara
ACS_ID = getAcsId(PGN_ID); //Mengambil
ACS_ID dari table smp_access
ACS_OUT = getAcsOut(ACS_ID); //Mengambil
ACS_TIME_OUT dari table smp_access
PTL_ID = getPtlAcs(ACS_ID); //Mengambil
PTL_ID dari table smp_portal
PTL_IN = getPtlIn(PTL_ID); //Mengambil
PTL_IN dari table smp_portal
PTL_OUT = getPtlOut(PTL_ID); //Mengambil
PTL_IN dari table smp_portal

Serial.print("Message : ");
if (PTL_ID != NULL && PTL_IN != "0000-
00-00 00:00:00") { //Cek motor sudah berada
didalam

```

```
        if (ACS_OUT != "0000-00-00 00:00:00"
&& PTL_OUT == "0000-00-00 00:00:00") { //cek
apa pengendara pernah ada didalam
        // Servo Open and Close
        updateKeluar(PTL_ID, ACS_ID);
//Update data ke smp_portal
        Serial.println("Authorized
access");
        delay(500);
        digitalWrite(LED_G, HIGH);
        tone(BUZZER, 500);
        delay(300);
        noTone(BUZZER);
        myServo.write(180);
        sServo = 1;
    } else {
        Serial.println("Please tap to your
motorcyle first.");
        delay(500);
        tone(BUZZER, 300);
        delay(1000);
        noTone(BUZZER);
    }
    Serial.println();
} else {
    Serial.println("Access denied.");
    Serial.println();
    delay(500);
    tone(BUZZER, 300);
    delay(1000);
    noTone(BUZZER);
}
} else if (GRAM == 0 && sServo == 1) {
    sServo = 0;
    delay(1000);
    digitalWrite(LED_G, LOW);
    myServo.write(90);
}
}
```

```

// ----- Membaca Hasil Input
Kartu
void messageReadInput(int GRAM, String hex,
String dec) {
    Serial.print("Reading: ");
    Serial.print(GRAM);
    Serial.println(" Gram");
    Serial.println("The NUID tag is:");
    Serial.print("In hex: ");
    Serial.println(hex);
    Serial.print("In dec: ");
    Serial.println(dec);
}

// ----- Mengambil ID pengendara
yang melakukan tap - TABLE smp_pengendara
String getPgnId(String hex) {
    //Cek PGN_ID sudah melakukan tap atau
    belum pada TABLE smp_access
    String PGN_ID = ""; //Store Variable
    MySQL_Cursor *cur_pengguna = new
    MySQL_Cursor(&conn);
    cur_pengguna->execute(("SELECT * FROM
    ta_smp.smp_pengendara WHERE PGN_CARD_ID = '"
    + hex + "'").c_str());
    column_names *col_pengguna =
    cur_pengguna->get_columns();
    row_values *row_pengguna = NULL;
    do {
        row_pengguna = cur_pengguna-
        >get_next_row();
        if (row_pengguna != NULL) {
            for (int f = 0; f < col_pengguna-
            >num_fields; f++) {
                PGN_ID = row_pengguna->values[0];
            }
        }
    } while (row_pengguna != NULL);
    delete cur_pengguna;
    return PGN_ID;
}

```

```

}

// ----- Mengambil ACS_ID - TABLE
smp_access
String getAcsId(String id){
    String ACS_ID = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE PGN_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_access = cur_access-
>get_columns();
    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {
                //Store values id to id_card
                ACS_ID = row_access->values[0];
            }
        }
    } while (row_access != NULL);
    delete cur_access;
    return ACS_ID;
}

// ----- Mengambil ACS_IN - TABLE
smp_access
String getAcsOut(String id){
    String ACS_OUT = ""; //Store Variable
    MySQL_Cursor *cur_access = new
MySQL_Cursor(&conn);
    cur_access->execute(("SELECT * FROM
ta_smp.smp_access WHERE ACS_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_access = cur_access-
>get_columns();
}

```

```

    row_values *row_access = NULL;
    do {
        row_access = cur_access-
>get_next_row();
        if (row_access != NULL) {
            for (int f = 0; f < col_access-
>num_fields; f++) {
                //Store values id to id_card
                ACS_OUT = row_access->values[4];
            }
        }
    } while (row_access != NULL);
    delete cur_access;
    return ACS_OUT;
}

// ----- Mengambil PTL_ID - TABLE
smp_portal
String getPtlAcs(String id){
    String PTL_ID = ""; //Store Variable
    MySQL_Cursor *cur_portal = new
MySQL_Cursor(&conn);
    cur_portal->execute(("SELECT * FROM
ta_smp.smp_portal WHERE ACS_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_portal = cur_portal-
>get_columns();
    row_values *row_portal = NULL;
    do {
        row_portal = cur_portal-
>get_next_row();
        if (row_portal != NULL) {
            for (int f = 0; f < col_portal-
>num_fields; f++) {
                //Store values id to id_card
                PTL_ID = row_portal->values[0];
            }
        }
    } while (row_portal != NULL);
    delete cur_portal;
}

```

```

        return PTL_ID;
    }

// ----- Mengambil PTL_IN - TABLE
smp_portal
String getPt1In(String id){
    String PTL_IN = ""; //Store Variable
    MySQL_Cursor *cur_portal = new
MySQL_Cursor(&conn);
    cur_portal->execute(("SELECT * FROM
ta_smp.smp_portal WHERE PTL_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
    column_names *col_portal = cur_portal-
>get_columns();
    row_values *row_portal = NULL;
    do {
        row_portal = cur_portal-
>get_next_row();
        if (row_portal != NULL) {
            for (int f = 0; f < col_portal-
>num_fields; f++) {
                //Store values id to id_card
                PTL_IN = row_portal->values[2];
            }
        }
    } while (row_portal != NULL);
    delete cur_portal;
    return PTL_IN;
}

// ----- Mengambil PTL_OUT -
TABLE smp_portal
String getPt1Out(String id){
    String PTL_OUT = ""; //Store Variable
    MySQL_Cursor *cur_portal = new
MySQL_Cursor(&conn);
    cur_portal->execute(("SELECT * FROM
ta_smp.smp_portal WHERE PTL_ID = '" + id + "'"
AND IS_DEL = 0").c_str());
}

```

```

        column_names *col_portal = cur_portal-
>get_columns();
        row_values *row_portal = NULL;
        do {
            row_portal = cur_portal-
>get_next_row();
            if (row_portal != NULL) {
                for (int f = 0; f < col_portal-
>num_fields; f++) {
                    //Store values id to id_card
                    PTL_OUT = row_portal->values[3];
                }
            }
            } while (row_portal != NULL);
            delete cur_portal;
            return PTL_OUT;
        }

// ----- Insert Access ke Portal
Masuk - TABLE smp_portal
void updateKeluar(String id, String acs){
    MySQL_Cursor *update_query = new
MySQL_Cursor(&conn);
    update_query->execute(("UPDATE
ta_smp.smp_portal SET
PTL_OUT=CURRENT_TIMESTAMP(), IS_DEL=1 WHERE
PTL_ID = "+ id ).c_str());
    delete update_query;

    MySQL_Cursor *update_query2 = new
MySQL_Cursor(&conn);
    update_query2->execute(("UPDATE
ta_smp.smp_access SET IS_DEL=1 WHERE ACS_ID =
"+ acs ).c_str());
    delete update_query2;
}

```

5. Kode Program *Controller* Motor

```

<?php
defined('BASEPATH') OR exit('No direct script
access allowed');

class Motor extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('M_CallsQL');
        $this->load->model('M_Login');
    }

    public function index() {
        $header = array(
            'title' => 'Motor - List | SMP
System',
            'content' => 'motor/v_index',
        );

        $data = array(
            'header' => $header,
            'sess' => $this->M_Login-
>sessdata(),
            'motor' => $this->M_CallsQL-
>get_dataMotor("smp_pengendara")->result(),
            'pemilik' => $this->M_CallsQL-
>get_data("smp_pengendara")->result(),
        );
        $this->load->view('layout/v_app',
$data);
    }

    public function input() {
        $this->form_validation-
>set_rules('nama', 'Nama', 'required');
        $this->form_validation-
>set_rules('getUID', 'UID Card', 'required');
    }
}

```

```

        $this->form_validation-
>set_rules('nopol', 'Nomor
Polisi', 'required');

        if ($_POST) {
            if ($this->form_validation->run ()
!= FALSE) {
                $nama = $this->input-
>post('nama');
                $nopol = $this->input-
>post('nopol');
                $uid = $this->input-
>post('getUID');
                $insert = array(
                    'MTR_HW_ID' => $uid,
                    'MTR_NO_POL' => $nopol,
                    'PGN_ID' => $nama
                );
                $check = $this->M_CallSQL-
>where("smp_motor", array('MTR_HW_ID' =>
$uid))->row();
                if ($check) {
                    $this->session-
>set_flashdata('error', 'Kartu sudah
terdaftar.');
                } else {
                    $this->M_CallSQL-
>input_data($insert, "smp_motor");
                    $this->session-
>set_flashdata('success', 'Data telah
berhasil ditambahkan.');
                }
            } else {
                $this->session-
>set_flashdata('error', 'Silahkan cek kembali
pengisian anda.');
            }
            redirect('motor/input');
        } else {
            $header = array(

```

```

        'title' => 'Motor - Tambah |  

SMP System',  

        'content' => 'motor/v_input',  

    );
}

$data = array(  

    'header' => $header,  

    'sess' => $this->M_Login->sessdata(),  

    'pemilik' => $this->M_CallsSQL->get_data("smp_pengendara")->result(),  

);
$this->reset_uid();
$this->load->view('layout/v_app',  

$data);
}
}

public function edit($id=null)
{
    if (!isset($id)) redirect('motor');

    $this->form_validation->set_rules('nama','Nama','required');
    $this->form_validation->set_rules('nopol','Nomor Polisi','required');

    if($this->form_validation->run() != FALSE) {
        $nama = $this->input->post('nama');
        $nopol = $this->input->post('nopol');
        $insert = array(
            'MTR_NO_POL' => $nopol,
            'PGN_ID' => $nama
        );
    }
}

```

```

        $this->M_CallsQL-
>update_data(['MTR_ID' =>
$id],$insert,"smp_motor");
        $this->session-
>set_flashdata('success', 'Data telah
berhasil diubah.');
    } else {
        $this->session-
>set_flashdata('error', 'Silahkan cek kembali
pengisian anda.');
    }
    redirect('motor');
}

public function getuidmtr()
{
    $this->load->view('UIDContainermtr');
}

public function reset_uid()
{
    $Write=<?php $" . "UIDresult=''; " .
"echo $" . "UIDresult;" . " ?>";
    file_put_contents('application/views/UIDConta
inermtr.php', $Write);
}

public function parse_uid()
{
    $UIDresult = $this->input-
>post('UIDresult');
    $Write=<?php $" . "UIDresult='"
$UIDresult . "'; " . "echo $" . "UIDresult;" .
" ?>";
    echo
    file_put_contents('application/views/UIDConta
inermtr.php', $Write);
}

```

```

public function delete($id=null)
{
    $data = [
        'MTR_ID' => $id,
    ];
    if (!isset($id)) show_404();
    if ($this->M_CallSQL-
>delete("smp_motor", $data)) {
        $this->session-
>set_flashdata('success', 'Berhasil menghapus
data.');
        redirect('motor');
    }
}

```

6. Kode Program *View Index Motor*

```

<div class="ibox">
    <div class="ibox-head">
        <div class="ibox-title">Data
Motor</div>
    </div>
    <div class="ibox-body">
        <div style="overflow-x:auto;">
            <table class="table table-
striped table-bordered table-hover"
id="table-Datatable" cellspacing="0"
width="100%">
                <thead>
                    <tr>
                        <th style="text-
align:center;">No</th>
                        <th style="text-
align:center;">Motor ID</th>
                        <th style="text-
align:center;">Nomor Polisi</th>

```

```
 Pemilik | Action |
<?php
$no = 1;
foreach($motor as
$pgn) {
echo '
|  |  |  |  |
| --- | --- | --- | --- |
|' . $no . '
 ' . $pgn->MTR_HW_ID . ' | ' . $pgn->MTR_NO_POL . ' | ' . $pgn->PGN_NAMA . ' | <button class="btn btn-warning" type="button" data- id="' . $pgn->MTR_ID . '" data-nopol="' . $pgn->MTR_NO_POL . '" data-pgn="' . $pgn->PGN_ID . '" data- toggle="modal" data-target="#edit"><i class="fa fa-edit"></i></button> <button class="btn btn-danger" type="button" data- id="' . $pgn->MTR_ID . '" data-toggle="modal" data-target="#delete"><i class="fa fa- minus"></i></button> |

';
$no++;
}
?>
```

```
        </tbody>
    </table>
</div>
</div>
</div>

<!-- Delete modal -->
<div class="modal fade" id="delete"
tabindex="-1" role="dialog" aria-
hidden="true">
    <div class="modal-dialog modal-md">
        <div class="modal-content">

            <div class="modal-header">
                <h4 class="modal-title" id="myModalLabel">Delete Data</h4>
                <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">x</span>
            </button>
        </div>
        <div class="modal-body">
            <p>Apa anda yakin ingin
menghapus data ini?</p>
        </div>
        <div class="modal-footer">
            <form id="delForm"
class="form-horizontal form-label-left"
action="" method="POST"
enctype="multipart/form-data">
                <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
                <button type="submit" class="btn btn-danger">Delete</button>
            </form>
        </div>
    </div>
</div>
</div>
```

```
<!-- Edit modal -->
<div class="modal fade" id="edit"
tabindex="-1" role="dialog" aria-
hidden="true">
    <div class="modal-dialog modal-md">
        <div class="modal-content">

            <div class="modal-header">
                <h4 class="modal-title" id="myModalLabel">Edit Data</h4>
                <button type="button" class="close" data-dismiss="modal"><span aria-hidden="true">x</span>
                </button>
            </div>
            <div class="modal-body">
                <div class="ibox-body">
                    <form id="editForm" class="form-horizontal" action="" method="POST" >
                        <div class="form-group">
                            <div class="input-group">
                                <div class="input-group-addon"><i class="fa fa-users font-16"></i></div>
                                <select id="nama" name="nama" style="width: 100%" class="form-control select2-apply" placeholder="" required>
                                    <?php
foreach ($pemilik as $pgn) {
echo
'<option value="'. $pgn->PGN_ID.'">'>' . $pgn->PGN_NAMA . '</option>';
}
?>
                            </select>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

                </div>
            </div>
            <div class="form-group">
                <div class="input-
group">
                    <div
class="input-group-addon"><i class="fa fa-car
font-16"></i></div>
                    <input
type="text" id="nopol" name="nopol"
class="form-control" placeholder="Nomor
Polisi" required>
                </div>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button"
class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
            <button type="submit"
class="btn btn-success">Save changes</button>
        </form>
    </div>
</div>
</div>

<!-- CORE PLUGINS-->
<script src="

```

```

        if ( event.relatedTarget != null)
{
    var div =
$(event.relatedTarget)
}

var modal = $(this)

modal.find('#nopol').attr("value",div.data('n
opol'));
        modal.find('#nama
option[value="'+div.data('pgn')+']']").attr('s
elected', true);

modal.find('#editForm').attr("action",
function(i, value) {
    return "<?php echo
base_url('motor/edit') ?>/"
+ div.data('id');
});

$('#edit').on('hidden.bs.modal',
function (event) {
    var div = $(event.relatedTarget)
    var modal = $(this);
    modal.find('#nama
option').attr('selected', false);
})

$('#delete').on('show.bs.modal',
function (event) {
    var div =
$(event.relatedTarget)
    var modal = $(this)

modal.find('#delForm').attr("action",
function(i, value) {

```

```

        return "<?php echo
base_url('motor/delete') ?>/" +
div.data('id')) +
});
});
});
</script>

```

7. Kode Program *View Input Motor*

```

<div class="row">
<div class="col-md-
2">
</div>
<div class="col-md-
8">
<div
class="ibox">
<div
class="ibox-head">
<div
class="ibox-title">Tambah Motor</div>
</div>
<div
class="ibox-body">
<form
class="form-horizontal" action="<?php echo
base_url('motor/input') ?>" method="POST" >
<div
class="form-group">
<div class="input-group">
<div class="input-group-addon"><i class="fa
fa-id-card font-16"></i></div>
<textarea id="getUID" name="getUID"
class="form-control" placeholder="Please tap
your card" style="resize: none; height:

```

```
40px;" rows="1" cols="1" required  
readonly></textarea>  
  
</div>  
  
</div> <div  
class="form-group">  
  
<div class="input-group">  
  
<div class="input-group-addon"><i class="fa  
fa-users font-16"></i></div>  
  
<select id="nama" name="nama" class="form-  
control select2-apply" placeholder=""  
required>  
  
<option selected disabled>Pemilik</option>  
  
<?php  
  
foreach($pemilik as $pgn) {  
  
echo '<option value="' . $pgn-  
>PGN_ID. '">' . $pgn->PGN_NAMA. '</option>' ;  
  
}  
  
?  
  
</select>  
  
</div>  
  
</div> <div  
class="form-group">
```

```
<div class="input-group">

<div class="input-group-addon"><i class="fa fa-car font-16"></i></div>

<input type="text" id="nopol" name="nopol"
class="form-control" placeholder="Nomor
Polisi" required>

</div>

</div>
<div
class="form-group row">

<div class="col-md-12 ml-md-auto">

<button type="submit" class="btn btn-primary
pull-right">Submit</button>

</div>

</div>
</div>
</form>
</div>
</div>
</div>
<div class="col-md-
2">
</div>
</div>

<script src="php echo
config_item('assets'); ?&gt;vendors/jquery/dist/j
query.min.js"
type="text/javascript"&gt;&lt;/script&gt;

&lt;script&gt;
$(document).ready(function() {
    $("#getUID").load("getuidmtr");</pre
```

```

        setInterval(function() {
            $("#getUID").load("getuidmtr");
        }, 500);
    });
</script>

```

8. Kode Program *Controller Pengendara*

```

<?php
defined('BASEPATH') OR exit('No direct script
access allowed');

class Pengendara extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model ('M_CallSQL');
        $this->load->model ('M_Login');
    }

    public function index() {
        $header = array(
            'title' => 'Pengendara - List |
SMP System',
            'content' =>
'pengendara/v_index',
        );

        $data = array(
            'header' => $header,
            'sess' => $this->M_Login-
>sessdata(),
            'pengendara' => $this->M_CallSQL-
>get_data ("smp_pengendara")->result(),
        );
        $this->load->view('layout/v_app',
$data);
    }

    public function input() {

```

```

        $this->form_validation-
>set_rules('nama', 'Nama', 'required');
        $this->form_validation-
>set_rules('getUID', 'UID Card', 'required');

    if($_POST) {
        if($this->form_validation->run()
!= FALSE) {
            $uid = $this->input-
>post('getUID');
            $nama = $this->input-
>post('nama');
            $insert = array(
                'PGN_CARD_ID' => $uid,
                'PGN_NAMA' => $nama,
            );
            $check = $this->M_CallSQL-
>where("smp_pengendara", ['PGN_CARD_ID' =>
$uid])->row();
            if($check) {
                $this->session-
>set_flashdata('error', 'Kartu sudah
terdaftar.');
            } else {
                $this->M_CallSQL-
>input_data($insert, "smp_pengendara");
                $this->session-
>set_flashdata('success', 'Data telah
berhasil ditambahkan.');
            }
        } else {
            $this->session-
>set_flashdata('error', 'Silahkan cek kembali
pengisian anda.');
        }
        redirect('pengendara/input');
    } else {
        $header = array(
            'title' => 'Pengendara -
Tambah | SMP System',

```

```

        'content' =>
'pengendara/v_input',
);

$data = array(
    'header' => $header,
    'sess' => $this->M_Login-
>sessdata(),
);
$this->reset_uid();
$this->load->view('layout/v_app',
$data);
}
}

public function edit($id=null)
{
    if (!isset($id)) redirect('motor');

    $this->form_validation-
>set_rules('nama', 'Nama', 'required');

    if ($this->form_validation->run() != FALSE) {
        $nama = $this->input-
>post('nama');
        $insert = array(
            'PGN_NAMA' => $nama
        );
        $this->M_CallSQL-
>update_data(['PGN_ID' =>
$id], $insert, "smp_pengendara");
        $this->session-
>set_flashdata('success', 'Data telah
berhasil diubah.');
    } else {
        $this->session-
>set_flashdata('error', 'Silahkan cek kembali
pengisian anda.');
    }
}

```

```

        redirect('pengendara');
    }

public function getuidpgn()
{
    $this->load->view('UIDContainerpgn');
}

public function reset_uid()
{
    $Write = "<?php \$" . "UIDresult=''; " .
"echo \$" . "UIDresult;" . " ?>";

file_put_contents('application/views/UIDConta
inerpgn.php', $Write);
}

public function parse_uid()
{
    $UIDresult = $this->input-
>post('UIDresult');
    $Write = "<?php \$" . "UIDresult='".
$UIDresult . "'; " . "echo \$" . "UIDresult;" .
. " ?>";
    echo
file_put_contents('application/views/UIDConta
inerpgn.php', $Write);
}

public function delete($id=null)
{
    $data = [
        'PGN_ID' => $id,
    ];
    if (!isset($id)) show_404();
    if ($this->M_CallSQL-
>delete("smp_pengendara", $data)) {
        $this->session-
>set_flashdata('success', 'Berhasil menghapus
data.');
}

```

```
        redirect('pengendara');
```

9. Kode Program *View Index Pengendara*

```

<td>' . $no . '</td>
<td>' . $pgn-
>PGN_CARD_ID.'</td>
<td>' . $pgn-
>PGN_NAMA.'</td>
<td>
<button
class="btn btn-warning" type="button" data-
id="' . $pgn->PGN_ID.'" data-nama="' . $pgn-
>PGN_NAMA.'" data-toggle="modal" data-
target="#edit"><i class="fa fa-
edit"></i></button>
<button
class="btn btn-danger" type="button" data-
id="' . $pgn->PGN_ID.'" data-toggle="modal"
data-target="#delete"><i class="fa fa-
minus"></i></button>
</td>
</tr>';
$no++;
}
?>
</tbody>
</table>
</div>
</div>
</div>

<!-- Delete modal -->
<div class="modal fade" id="delete"
tabindex="-1" role="dialog" aria-
hidden="true">
<div class="modal-dialog modal-md">
<div class="modal-content">

<div class="modal-header">
<h4 class="modal-title"
id="myModalLabel">Delete Data</h4>

```

```
        <button type="button"
class="close" data-dismiss="modal"><span
aria-hidden="true">x</span>
        </button>
    </div>
    <div class="modal-body">
        <p>Apa anda yakin ingin
menghapus data ini?</p>
    </div>
    <div class="modal-footer">
        <form id="delForm"
class="form-horizontal form-label-left"
action="" method="POST"
enctype="multipart/form-data">
            <button type="button"
class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
            <button type="submit"
class="btn btn-danger">Delete</button>
        </form>
    </div>
</div>
</div>

<!-- Edit modal -->
<div class="modal fade" id="edit"
tabindex="-1" role="dialog" aria-
hidden="true">
    <div class="modal-dialog modal-md">
        <div class="modal-content">

            <div class="modal-header">
                <h4 class="modal-title"
id="myModalLabel">Edit Data</h4>
                <button type="button"
class="close" data-dismiss="modal"><span
aria-hidden="true">x</span>
                </button>
            </div>
```

```

<div class="modal-body">
    <div class="ibox-body">
        <form id="editForm"
class="form-horizontal" action=""
method="POST" >
            <div class="form-group">
                <div class="input-
group">
                    <div
class="input-group-addon"><i class="fa fa-
user font-16"></i></div>
                    <input
type="text" id="nama" name="nama"
class="form-control" placeholder="Name"
required>
                </div>
            </div>
        </div>
        <div class="modal-footer">
            <button type="button"
class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
            <button type="submit"
class="btn btn-success">Save changes</button>
        </form>
    </div>
</div>
</div>
</div>

<!-- CORE PLUGINS-->
<script src=<?php echo
config_item('assets') ; ?>vendors/jquery/dist/j
query.min.js"
type="text/javascript"></script>

<!-- Modal -->
<script>
    $(document).ready(function() {

```

```
$('#edit').on('show.bs.modal',
function (event) {

    if ( event.relatedTarget != null)
{
        var div =
$(event.relatedTarget)
    }

    var modal = $('#this')

modal.find('#nama').attr("value",div.data('na
ma'));

modal.find('#editForm').attr("action",
function(i, value) {
    return "<?php echo
base_url('pengendara/edit') ?>/"
+ div.data('id');
    });
}

$('#delete').on('show.bs.modal',
function (event) {
    var div =
$(event.relatedTarget)
    var modal = $('#this')

modal.find('#delForm').attr("action",
function(i, value) {
    return "<?php echo
base_url('pengendara/delete') ?>/"
+ div.data('id');
    });
}
);

</script>
```

10. Kode Program *View Input Pengendara*

```
<div class="row">
    <div class="col-md-2">
        </div>
    <div class="col-md-8">
        <div
            class="ibox">
                <div
                    class="ibox-head">
                        <div
                            class="ibox-title">Tambah Pengendara</div>
                        </div>
                    <div
                        class="ibox-body">
                            <form
                                class="form-horizontal" action="php echo
                                base_url('pengendara/input') ?&gt;"&gt;
                                method="POST" &gt;
                            &lt;div
                                class="form-group"&gt;
                                &lt;div class="input-group"&gt;
                                    &lt;div class="input-group-addon"&gt;&lt;i class="fa
                                        fa-id-card"&gt;&lt;/i&gt;&lt;/div&gt;
                                    &lt;textarea id="getUID" name="getUID"
                                        class="form-control" placeholder="Please tap
                                        your card" style="resize: none; height:
                                        40px;" rows="1" cols="1" required
                                        readonly&gt;&lt;/textarea&gt;
                                &lt;/div&gt;
                            &lt;/div&gt;
                        &lt;/div&gt;
                    &lt;div
                        class="form-group"&gt;</pre
```

```
<div class="input-group">

<div class="input-group-addon"><i class="fa fa-user font-16"></i></div>

<input type="text" id="nama" name="nama"
class="form-control" placeholder="Name"
required>

</div>

</div>
<div
class="form-group row">

<div class="col-md-12 ml-md-auto">

<button type="submit" class="btn btn-primary
pull-right">Submit</button>

</div>

</div>
</div>
</form>
</div>
</div>
</div>
<div class="col-md-2">
</div>
</div>

<script src="php echo
config_item('assets'); ?&gt;vendors/jquery/dist/j
query.min.js"
type="text/javascript"&gt;&lt;/script&gt;

&lt;script&gt;
$(document).ready(function() {</pre
```

```

$( "#getUID" ).load("getuidpgn");
setInterval(function() {
    $( "#getUID" ).load("getuidpgn");
}, 500);
});
</script>

```

11. Kode Program *Model SQL*

```

<?php
defined('BASEPATH') OR exit('No direct script
access allowed');

class M_CallSQL extends CI_Model{

    function where($table,$where) {
        return $this->db-
>get_where($table,$where);
    }

    function get_data($table) {
        return $this->db->get_where($table);
    }

    function get_dataMotor($table) {
        return $this->db->select("*")
            ->from("smp_motor a")
            ->join("smp_pengendara b",
"b.PGN_ID = a.PGN_ID")
            ->get();
    }

    function input_data($data,$table) {
        $this->db->insert($table,$data);
    }

    function
update_data($where,$data,$table){
    $this->db->where($where);
}

```

```

        $this->db->update ($table, $data);
    }

function delete ($table, $data)
{
    return $this->db-
>delete ($table, $data);
}

function getAksesMotor ()
{
    return $this->db->select ("a.ACS_ID,
a.PGN_ID, a.ACS_TIME_IN, a.ACS_TIME_OUT,
b.PGN_NAMA, c.MTR_NO_POL, (SELECT PGN_NAMA
FROM smp_pengendara WHERE PGN_ID = c.PGN_ID)
as PGN_MTR_NAMA,
IF(a.ACS_TIME_IN&a.ACS_TIME_OUT, 'Keluar', 'Mas
uk') PTL_STATUS")
                                ->from ("smp_access
a")
                                -
>join ("smp_pengendara b", "b.PGN_ID =
a.PGN_ID")
                                ->join ("smp_motor
c", "c.MTR_ID = a.MTR_ID")
                                -
>where ("a.IS_DEL", "0")
                                ->get ();
}

function getAksesPortal ()
{
    return $this->db->select ("d.PTL_ID,
a.ACS_ID, a.PGN_ID, d.PTL_IN, d.PTL_OUT,
b.PGN_NAMA, c.MTR_NO_POL, (SELECT PGN_NAMA
FROM smp_pengendara WHERE PGN_ID = c.PGN_ID)
as PGN_MTR_NAMA")
                                ->from ("smp_portal
d")
                                ->join ("smp_access
a", "a.ACS_ID = d.ACS_ID")
}

```

```
>join("smp_pengendara b", "b.PGN_ID =  
a.PGN_ID")  
      ->join("smp_motor  
c", "c.MTR_ID = a.MTR_ID")  
      ->where("d.IS_DEL", "0")  
      ->get();  
    }  
}
```

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Ahmad Shidqi Firdaus, lahir di Jakarta tanggal 11 Maret 1998. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis telah menempuh pendidikan formal di SDIT Nurfatahillah (2004-2010), SMPIT Insan Harapan (2010-2013), SMA Negeri 12 Kota Tangerang Selatan (2013-2016). Penulis melanjutkan studi dengan berkuliah pada program sarjana (S1) di Departemen Informatika ITS. Selama kuliah di Informatika ITS, penulis mengambil bidang minat Arsitektur dan Jaringan Komputer (AJK). Selama menempuh perkuliahan, penulis pernah menjadi asisten dosen Sistem Basis Data, dan Jaringan Komputer. Penulis juga aktif mengikuti organisasi kemahasiswaan, yaitu sebagai Otak Kiri di Teater Tiyang Alit ITS. Selain aktif mengikuti organisasi, penulis juga aktif dalam kegiatan kepanitiaan Schematics, yaitu sebagai staff Perkap pada tahun 2017-2018, kepanitiaan GERIGI ITS sebagai staff perkap pada tahun 2017, kepanitiaan GERIGI ITS sebagai konseptor acara pada tahun 2018, kepanitiaan INI LHO ITS sebagai staff ahli logistik pada tahun 2018. Penulis dapat dihubungi melalui surel di ahmadshidqifirdaus@gmail.com.