



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR-TE 141599

**RANCANG BANGUN SISTEM PEMBAYARAN UNTUK PARKIR
PRABAYAR DI TEPI JALAN MENGGUNAKAN RFID**

Khoiruli Miftachul Huda
NRP 2212 100 076

Dosen Pembimbing
Suwito , S.T., M.T.
Ir. Tasripan, M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR-TE 141599

**PREPAID PAYMENT SYSTEM FOR PARKING ON THE ROADSIDE
USING RFID DESIGN**

Khoiruli Miftachul Huda
NRP 2212 100 076

Dosen Pembimbing
Suwito , S.T., M.T.
Ir. Tasripan, M.T.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industry Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “*Rancang Bangun Sistem Pembayaran untuk Parkir Di Tepi Jalan Menggunakan RFID*” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2017

Khoiruli Miftachul Huda
NRP 2212100076

Halaman ini sengaja dikosongkan

**RANCANG BANGUN SISTEM PEMBAYARAN UNTUK
PARKIR PRABAYAR DI TEPI JALAN MENGGUNAKAN RFID**

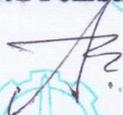
TUGAS AKHIR

**Diajukan Untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada
Bidang Studi Teknik Elektronika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Menyetujui:

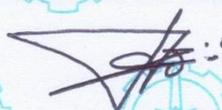
Dosen Pembimbing I,



Suwito, S.T., M.T.

NIP: 19810105200501104

Dosen Pembimbing II,



Ir. Tasripan, M.T.

NIP: 19620418199031004



Halaman ini sengaja dikosongkan

RANCANG BANGUN SISTEM PEMBAYARAN UNTUK PARKIR PRABAYAR DITEPI JALAN MENGGUNAKAN RFID

Nama : Khoiruli Miftachul Huda
Pembimbing : Suwito, S.T., M.T.
Ir. Tasripan, M.T.

ABSTRAK

Dalam tugas akhir ini dirancang sebuah sistem pendukung pelayanan parkir di tepi jalan umum berupa alat bantu pemungutan retribusi parkir berbasis *Radio Frequency Identification* (RFID). Sistem ini menggantikan sistem karcis dan setoran. Sehingga retribusi tersebut bisa langsung terkumpul sebagai pendapatan kota atau daerah.

Pada kendaraan dipasangkan sebuah *tag* RFID yang akan dibaca oleh *reader* yang dioperasikan operator saat akan parkir. *Reader* menggunakan modul MFRC522 dan mikrokontroler Atmega328 pada Arduino Nano. *Reader* terhubung dengan *Raspberry-Pi* sebagai *server* lokal dalam suatu area parkir melalui komunikasi radio dengan modul NRF24L01. *Raspberry-Pi* terhubung ke *web server* melalui koneksi internet. *Web server* ini berfungsi sebagai pusat data dan penghitung biaya parkir kendaraan terhadap durasi parkir.

Pengujian yang dilakukan meliputi pengujian pembacaan *tag*, penghitungan biaya parkir, dan proses mulai parkir dan mengakhiri layanan parkir. Pembacaan *tag* oleh *reader* dan pengiriman hasil bacanya ke *Raspberry Pi* menggunakan NRF24L01 memiliki tingkat keberhasilan yang tinggi, yakni mencapai 100%. Proses sistem untuk memulai parkir dan mengakhiri parkir memiliki rata-rata tingkat keberhasilan masing-masing 92% dan 84%. Tingkat keberhasilannya dipengaruhi oleh kecepatan koneksi internet dan *server host* yang dipakai.

Kata Kunci : parkir di tepi Jalan, RFID, *web server*

Halaman ini sengaja dikosongkan

PREPAID PAYMENT SYSTEM FOR PARKING ON THE ROADSIDE USING RFID DESIGN

Nama : Khoiruli Miftachul Huda
Pembimbing : Suwito, S.T., M.T.
Ir. Tasripan, M.T.

ABSTRACT

In this final project, designed a support system of parking services on the public roads in the form of parking fee collection tools based on Radio Frequency Identification (RFID). This system replaces the ticket and old payment system. So that the fees can go directly collected as revenue the city or county.

On the vehicles installed an RFID tag to be read by a reader which operated by an operator while the vehicles will be parking. Reader using MFRC522 module and a microcontroller ATmega328 on the Arduino Nano. Reader connected with Raspberry-Pi as a local server in a parking area through radio communication with NRF24L01 module. Raspberry-Pi is connected to the web server via an internet connection. The Web server serves as a data center and computing vehicle parking fee on parking duration.

Tests performed include testing the readability tag, counting the cost of parking, and the starting parking and ending parking service. The reading of the tag by a reader and sending the result to the Raspberry Pi using NRF24L01 has a high success rate, wich reached 100%. Park system processes to start and end the parking service have an average success rate of respectively 92% and 84%. The success rate is influenced by the speed of the Internet connection and the server host is used.

Kata Kunci : parking on the public roads, RFID, Web server

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat-Nya, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul : “**Rancang Bangun Sistem Pembayaran Untuk Parkir Prabayar Ditepi Jalan Menggunakan RFID**”. Penulis menyadari bahwa dalam penyelesaian pengerjaan Tugas Akhir ini tidak terlepas dari bantuan dari berbagai pihak. Untuk itu dengan segala hormat, penulis mengucapkan terima kasih kepada

1. Orang tua penulis yang telah memberikan dukungan baik secara moril maupun materiil,
2. Dosen pembimbing Bapak Suwito, S.T., M.T. dan Bapak Ir. Tasripan, M.T. atas bimbingannya selama dalam proses pengerjaan tugas akhir ini,
3. Teman-teman asisten Laboratorium B202 yang senantiasa membantu dan memberikan semangat dalam mengerjakan tugas akhir,
4. Dan pihak-pihak lain yang telah berperan dalam penyelesaian Tugas Akhir ini,

Penulis memohon maaf atas segala kekurangan pada Tugas Akhir ini. Semoga Tugas Akhir ini dapat menjadi referensi dan tambahan ilmu bagi pembaca.

Surabaya, Januari 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	v
LEMBAR PENGESAHAN	vii
ABSTRAK	iii
ABSTRACT	v
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Tujuan	2
1.4 Batasan Masalah	3
1.5 Metodologi	3
1.5.1 Studi Literatur	3
1.5.2 Perancangan Hardware	3
1.5.3 Perancangan Software	4
1.5.4 Pengujian Sistem	4
1.5.5 Penyusunan Buku Tugas Akhir	4
1.6 Sistematika Penulisan	4
1.7 Relevansi	5
BAB II TEORI PENUNJANG	7
2.1 Parkir di Tepi Jalan Umum	7
2.2 Radio Frequency Identification	8
2.2.1 Cara Kerja RFID	8
2.3 RFID Reader MFRC522	10
2.4 Arduino Nano	12
2.5 RF Transceiver NRF24L01	13
2.6 MySQL Database	15
2.7 PHP	15
2.8 Web Server	15

BAB III PERANCANGAN SISTEM	17
3.1 Diagram Blok Sistem	17
3.2 Perancangan <i>Reader</i>	18
3.2.1 Perancangan Perangkat Keras	19
3.2.2 Perancangan Program Arduino Nano	20
3.3 Perancangan <i>Sub-server</i>	22
3.4 Perancangan <i>WebServer</i>	24
3.4.1 Perancangan Program PHP Pada Server	24
3.4.2 Perancangan Antarmuka Web.....	26
3.5 Perancangan Sistem RFID <i>Reader</i> untuk Pengoperasian Tanpa Operator.....	28
3.5.1 Perancangan Hardware	28
3.5.2 Perancangan Software.....	29
BAB IV HASIL DAN PENGUJIAN	31
4.1 Pengujian RFID <i>Reader</i>	31
4.2 Pengujian <i>WebServer</i> sistem parkir	32
4.2.1 Pengujian penghitungan durasi dan biaya parkir	32
4.2.2 Pengujian Halaman <i>Entry Member</i>	34
4.3 Pengujian Sistem Parkir Terintegrasi	35
4.3.1 Uji Proses Memulai Parkir.....	36
4.3.2 Uji Proses Mengakhiri Parkir.....	37
4.4 Pengujian <i>Interface</i> Sistem RFID <i>reader</i> tanpa operator	40
BAB V KESIMPULAN DAN SARAN	43
5.1 Kesimpulan.....	43
5.2 Saran.....	43
DAFTAR PUSTAKA.....	45
LAMPIRAN	47
BIODATA PENULIS	59

DAFTAR GAMBAR

Gambar 2.1 Diagram blok sederhana komponen transponder RFID	9
Gambar 2.2 Sinyal <i>carrier</i> termodulasi dengan metode <i>backscatter</i>	9
Gambar 2. 3 Diagram blok IC MFRC522 yang disederhanakan	11
Gambar 2. 4 Diagram rangkaian MFRC522	11
Gambar 2.5 <i>Board</i> Arduino Nano	12
Gambar 2.6 Diagram block nRF24L01	13
Gambar 3.1 Diagram blok sistem.....	18
Gambar 3.2 <i>Board Reader</i> tampak atas	20
Gambar 3.3 <i>Board Reader</i> tampak dari samping	20
Gambar 3.4 <i>Flow chart</i> program pada Arduino	21
Gambar 3.5 <i>Flow chart</i> sistem pada <i>sub-server</i>	23
Gambar 3.6 <i>Flow chart server</i>	25
Gambar 3.7 Rancangan tampilan halaman data member	26
Gambar 3.8 Rancangan halaman <i>history</i> parkir	27
Gambar 3.9 Rancangan halaman log parkir	27
Gambar 3.10 Rancangan halaman <i>entry member</i>	28
Gambar 3.11 Diagram blok sistem RFID reader tanpa operator	29
Gambar 3.12 <i>State</i> diagram sistem RFID reader tanpa operator	30
Gambar 4.1 Tampilan IDLE <i>Python</i> menampilkan data dari pembaca RFID	31
Gambar 4.2 Hasil <i>screen shoot</i> dari halaman log parkir pada pengujian penghitung durasi dan biaya parkir.....	33
Gambar 4.3 Proses pendaftaran member.....	35
Gambar 4.4 Member baru berhasil ditambahkan	35
Gambar 4.5 Konsol <i>sub-server</i>	36
Gambar 4.6 ID kendaraan ditambahkan ke halaman Log Parkir	37
Gambar 4.7 Perubahan status pada halaman data member	37
Gambar 4.8 Tampilan konsol <i>sub-server</i> saat proses mengakhiri layanan parkir	38
Gambar 4.9 Log parkir dengan ID “f6af3dd5” telah dihapus	38
Gambar 4.10 <i>History</i> parkir telah ditambah.....	38
Gambar 4.11 Cuplikan gambar dari halaman History Parkir setelah pengujian	39

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Konfigurasi pin untunk SPI Arduino Nano	19
Tabel 3.2 Asumsi harga pada sistem penghitung biaya	24
Tabel 3.3 Tabel hubungan <i>state</i> dan <i>interface</i> pada sistem RFID reader tanpa operator.....	29
Tabel 4.1 Hasil pengujian pembacaan tag RFID dan pengiriman ke Raspberry Pi dengan NRF24L01	32
Tabel 4.2 Hasil uji biaya dan durasi untuk mobil.....	33
Tabel 4.3 Hasil uji durasi dan biaya untuk motor	34
Tabel 4.4 Hasil pengujian proses memulai layanan parkir.....	39
Tabel 4.5 Hasil pengujian proses mengakhiri parkir	40
Tabel 4.6 Cuplikan gambar pengujian <i>interface</i> Sistem RFID reader tanpa operator.....	40

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Tugas Akhir merupakan penelitian yang dilakukan oleh mahasiswa S1 Institut Teknologi Sepuluh Nopember Surabaya. Tugas Akhir ini merupakan salah satu syarat wajib untuk menyelesaikan studi dalam program sarjana teknik.

Pada bab ini, akan dibahas mengenai hal-hal yang mendahului pelaksanaan Tugas Akhir. Hal-hal tersebut meliputi latar belakang, perumusan masalah, tujuan, batasan masalah, sistematika penulisan, dan relevansi.

1.1 Latar Belakang

Area parkir merupakan komponen yang sangat penting dalam sistem lalu lintas darat. Area parkir memungkinkan pengguna kendaraan untuk berhenti dari aktivitas lalu lintas dan melakukan aktivitas lain dengan meninggalkan kendaraannya. Sehingga area parkir harus menawarkan keamanan dan mudah diakses. Salah satu area parkir yang penting adalah area parkir di tepi jalan umum. Area parkir ini biasanya penting untuk kegiatan bisnis yang ada di pinggir jalan seperti pada area pertokoan atau perkantoran di pusat kota.

Dalam pelayanannya, parkir di tepi jalan umum biasanya dijaga oleh beberapa tukang parkir yang akan menarik uang jasa kepada pemakai layanan parkir tersebut. Padahal seharusnya uang retribusi parkir merupakan pendapatan daerah atau kota. Selain itu, uang retribusi parkir yang dikenakan kepada pengguna parkir lebih tinggi dari ketentuan pemerintah setempat. Ini tentu merugikan pengguna jalan, ditambah lagi adanya kurangnya perasaan aman karena petugas parkir di lapangan tidak jelas badan hukumnya.

Dalam tugas akhir ini dirancang sebuah sistem pelayanan parkir yang menawarkan kemudahan dalam pembayaran uang jasa parkir yang lebih tepat sasaran dan lebih aman. Pembayaran dilakukan dengan pengurangan saldo pada akun pengguna yang terdaftar untuk kendaraan tertentu. Sistem ini bekerja dengan berbasis pada RFID sebagai media pengenalan kendaraan dan identitas pengguna saat dimulainya pelayanan parkir. Sistem ini tidak lepas dari jasa petugas parkir, yang bisa merupakan petugas parkir resmi. Tag RFID ditempelkan pada bagian tertentu kendaraan yang mudah diakses pembaca RFID sebagai kunci

akses ke data kendaraan dan akun pengguna. Proses parkir dimulai dengan petugas parkir yang melakukan identifikasi kendaraan pada tag dengan menggunakan pembaca RFID, dan menekan tombol start pada pembaca RFID untuk memulai penghitungan biaya parkir.

Pembaca RFID yang dibawa petugas parkir tersebut terhubung dengan mikrokontroler yang terhubung ke sistem basis data melalui *sub-server* yang memiliki koneksi internet dengan cara HTTP *request* untuk proses identifikasi. Saat RFID terbaca, data dari kartu akan ditangkap oleh mikrokontroler. Kemudian mikrokontroler akan mengirimkan sinyal ke *server*, bahwa kendaraan dengan alamat RFID tersebut akan parkir di lokasi tersebut. *Server* yang menerima sinyal tersebut akan melakukan pembacaan data berdasarkan alamat tersebut dan melakukan eksekusi terhadap database yang bersangkutan, menambahkan catatan tentang parkir dan pengurangan saldo sebagai retribusi parkir.

Dengan sistem tersebut maka kecurangan beberapa oknum dalam hal parkir di tepi jalan umum dapat ditekan. Selain itu juga dapat meningkatkan kesan aman kepada pengguna jasa layanan parkir di tepi jalan umum. Jika sistem tersebut dapat diaplikasikan secara luas dalam sebuah kota dapat menambah pendapatan daerah atau kota tersebut terkait dengan jasa layanan parkir di tepi jalan umum.

1.2 Permasalahan

Permasalahan pada penelitian ini adalah sebagai berikut:

1. Bagaimana konfigurasi sistem pembacaan RFID yang sesuai dengan kondisi layanan parkir di tepi jalan umum.
2. Bagaimana merancang database yang sesuai dengan keperluan pelayanan parkir di tepi jalan umum
3. Bagaimana merancang konfigurasi komunikasi *sub-server* sebagai *node-master* komunikasi frekuensi radio (RF) bagi kontak dan komunikasinya ke *server*.
4. Bagaimana merancang antarmuka *server* untuk penyajian database parkir di tepi jalan umum.

1.3 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Merancang sebuah alat pembaca RFID yang dapat terhubung ke sebuah *server* melalui komunikasi RF ke *sub-server* yang terkoneksi dengan *server*.

2. Merancang sebuah node merancang sebuah data base yang dapat mengorganisir data kedatangan dan catatan terkait parkir di tepi jalan umum.
3. Merancang sebuah antarmuka *server* yang sesuai dengan kebutuhan sistem layanan parkir di tepi jalan.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Sistem dirancang hanya untuk kebutuhan pelayanan parkir di tepi jalan umum.
2. Mikrokontroler yang digunakan adalah Atmega328 dengan board Arduino Nano.
3. *Sub-server* menggunakan raspberry pi 2
4. Sistem pembayaran yang dimaksudkan disini adalah proses penghitungan biaya parkir dan pengurangan saldo yang telah dibayarkan sebelumnya

1.5 Metodologi

Dalam penyelesaian Tugas Akhir ini digunakan metodologi sebagai berikut:

1.5.1 Studi Literatur

Pada tahap ini dilakukan studi dan pengumpulan teori-teori penunjang sebagai bahan penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, *proceeding*, dan artikel-artikel di internet. Meliputi:

- Teori tentang parkir di tepi jalan
- Teori tentang RFID, RF *transceiver* NRF24L01, dan Arduino Nano.
- Mempelajari pemrograman Arduino, MySQL database, PHP, dan HTML.

1.5.2 Perancangan Hardware

Perancangan hardware meliputi perangkatian modul pada kotaktor yang tersusun atas pembaca RFID, mikrokontroler Arduino, RF transmitter-receiver dan komponen pendukung lainnya.

1.5.3 Perancangan Software

Perancangan software meliputi pembuatan program arduino untuk pembacaan tag RFID, pemilihan perintah dari tombol pilihan perintah oleh arduino, dan program untuk pengiriman data berupa ID dan kode perintah dari arduino ke *sub-server*, perancangan databasedengan MySQL, program interface *web* menggunakan HTML dan PHP. Dan juga program pada *sub-server* yang berfungsi sebagai node master dari komunikasi RF dengan kontaktor serta mengirimkan HTTP *request* ke *server*.

1.5.4 Pengujian Sistem

Pengujian sistem yang dilakukan adalah pegujian pembacaan ID tag RFID, pegiriman kode perintah dari arduino ke *sub-server* , uji HTTP ruquest dari *sub-server* ke *webservice*, dan pengujian *webservice*

1.5.5 Penyusunan Buku Tugas Akhir

Penyusunan buku Tugas Akhir dilakukan setelah mendapatkan data–data yang dibutuhkan. Buku tugas akhir terdiri dari dasar teori yang dipelajari, analisis dari data yang didapatkan, smapai kesimpulan dan saran.

1.6 Sistematika Penulisan

Laporan penelitian Tugas Akhir ini terbagi menjadi lima bab dengan susunan sistematika penulisan sebagai berikut:

BAB 1. Pendahuluan

Bab ini berisi tentang latar belakang, perumusan masalah, tujuan, metodologi penelitian, dan sistematika penulisan serta relevansi atau manfaat laporan tugas akhir.

BAB 2. Dasar Teori

Bab ini berisi tentang teori–teori penunjang yang berhubungan dengan permasalahan yang dibahas pada Tugas Akhir ini, antara lain: parkir di tepi jalan, radio frequency identification, RFID reader MFRC522, Arduino Nano, NRF24L01, MySQL Database dan PHP.

BAB 3. Perancangan Sistem

Bab ini membahas langkah-langkah perencanaan dan pembuatan sistem monitoring secara keseluruhan yang terdiri

dari perancangan *hardware* dan *software*. Perancangan *hardware* terdiri dari rangkaian sensor dan rangkaian catu daya. Sedangkan perancangan *software* meliputi pemrograman Arduino, PHP, HTML, dan MySQL database.

BAB 4. Pengujian dan Analisis

Bab ini berisi tentang penjelasan mengenai data dari hasil pengujian baik *hardware*, *software*, maupun sistem secara keseluruhan beserta analisis dari data yang didapatkan tersebut.

BAB 5. Penutup

Bab ini berisi kesimpulan dari proses perancangan dan pengujian sistem monitoring beserta analisisnya yang telah dilakukan pada bab sebelumnya, serta saran-saran untuk penelitian selanjutnya sehingga didapatkan sistem monitoring stasiun pengisian kendaraan listrik yang lebih baik lagi.

1.7 Relevansi

Hasil dari tugas akhir ini diharapkan mampu membantu permasalahan parkir di tepi jalan umum, khususnya dalam hal pungutan retribusi jasa layanan parkir di jalan umum.

Halaman ini sengaja dikosongkan

BAB II

TEORI PENUNJANG

Dalam penyusunan Tugas Akhir ini terdapat beberapa teori dasar yang menjadi acuan untuk merumuskan dan menyelesaikan masalah yang akan dibahas. Pada bagian awal diberikan tinjauan pustaka yang menggambarkan landasan teori secara umum yang akan digunakan pada tugas akhir ini. Pada bagian selanjutnya membahas tentang teori-teori pendukung.

2.1 Parkir di Tepi Jalan Umum

parkir di tepi jalan umum merupakan berhentinya kendaraan dan ditinggal pemiliknya untuk beberapa saat di area parkir tepi jalan umum yang disediakan oleh pemerintah daerah atau kota[1]. Lokasi yang dapat digunakan sebagai lahan parkir ini merupakan tempat tertentu pada jalan kabupaten, desa, atau kota yang dinyatakan dengan rambu lalu-lintas. Adapun lokasi yang dilarang untuk lokasi parkir adalah[2]:

- Jalan nasional dan jalan provinsi
- Pada jarak 6 m sebelum dan sesudah hidrant
- Parkir di pinggir jalan sebaiknya dilarang pada jalan 2 arah yang lebarnya kurang dari 6 m.
- Pada jarak 6 m sebelum dan sesudah zebra cross
- Pada jarak 25 m dari persimpangan
- Pada jarak 50 m dari jembatan
- Pada jarak 100 m dari perlintasan sebidang

Parkir di tepi jalan umum diatur dalam peraturan daerah atau kota terkait berkaitan dengan lahan parkir yang disediakan, sistem pelayanannya, termasuk retribusi. Retribusi merupakan pungutan yang dikenakan kepada pengguna jasa pelayanan umum oleh penyelenggara jasa, dalam hal ini adalah pemerintah daerah. Retribusi untuk pelayanan parkir ini tergolong retribusi jasa umum. Retribusi jasa umum disediakan untuk kepentingan umum. Hasil retribusi ini akan masuk ke kas umum daerah atau kota sebagai pendapatan daerah[1].

2.2 Radio Frequency Identification

Radio Frequency identification atau disingkat dengan RFID merupakan teknologi yang memungkinkan identifikasi suatu objek dengan menggunakan gelombang elektromagnet. RFID memiliki dua komponen utama yaitu tag/transponder dan pembaca RFID.

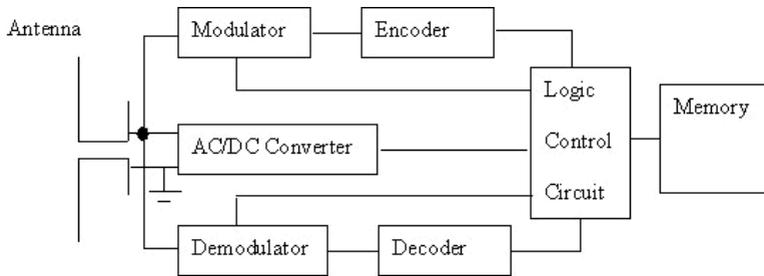
Ada dua tipe utama tag RFID, yakni tag pasif dan tag aktif. Tag pasif bekerja dengan menggunakan energi dari pembaca tag, sedangkan tag aktif bekerja menggunakan energi dari internal baterai. Tag pasif memiliki baterai atau jenis sumber energi lain, tapi memerlukan energi dari pembaca tag. Tag aktif dapat menghasilkan sinyal dengan daya sendiri sedangkan tag pasif menggunakan energi dari gelombang elektromagnet yang dipancarkan pemaca. Sehingga pembaca tag pasif harus bisa menghasilkan gelombang elektromagnet yang lebih besar dari pembaca tag aktif untuk mendapatkan sinyal balik dari tag pasif yang jauh lebih lemah dari tag aktif.

RFID juga dapat digolongkan berdasarkan frekuensi kerjanya. RFID yang bekerja dalam frekuensi 125 kHz digolongkan sebagai RFID *Low Frequency* (LF). Untuk frekuensi kerja 13,56 MHz digolongkan sebagai RFID *High Frequency* (HF). Untuk RFID yang bekerja pada frekuensi 300-1200 MHz digolongkan sebagai RFID *Ultra High Frequency*(UHF). Yang terakhir adalah RFID yang bekerja pada frekuensi 2,4 GHz dan 5,8 GHz digolongkan sebagai RFID *Microwave*.

2.2.1 Cara Kerja RFID

Sebuah tag atau transponder merupakan pembawa data dan normalnya berisikan nomor ID dan kode program EPC unik. Tag memiliki antena yang terhubung ke chip yang ada di dalamnya. Antena dapat berupa kabel, atau konduktor yang tercetak pada badan tag. Pembaca tag juga memiliki antena berupa koil yang berada di dalam *case*-nya. Melalui antena itulah pembaca tag menangkap data yang dipancarkan oleh tag dalam area deteksi pembaca tag.

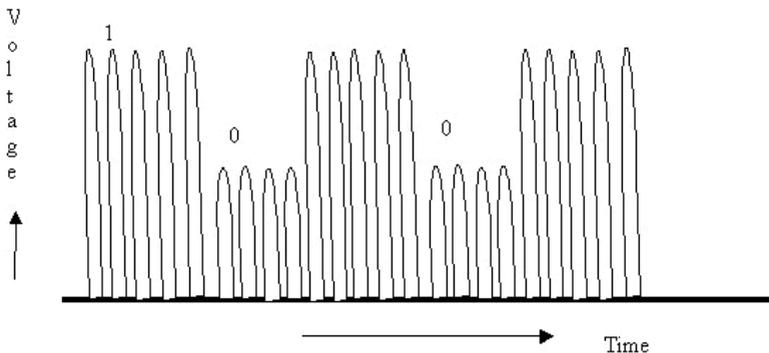
Berikut adalah diagram blok sederhana dari RFID tag, beberapa komponen dari tag dapat dilihat pada gambar 2.1. Pembacaan terjadi jika terjadi proses *handshaking* antara tag dan pembaca tag. Pembaca memancarkan sinyal RF *carrier* secara kontinu sambil terus menunggu adanya sinyal RF data yang ditangkap. Kehadiran tag pada area deteksi memodulasi medan RF, dan dalam waktu bersamaan terdeteksi oleh pembaca tag.



Gambar 2.1 Diagram blok sederhana komponen transponder RFID[4]

Tag (tag pasif) menyerap sebagian kecil dari energi yang dipancarkan oleh pembaca tag dan mulai mengirimkan data termulasi ketika energi yang diperlukan telah tercukupi dari energi medan RF. Data termulasi dengan modulasi langsung, FSK, atau modulasi fase. Pembaca tag memdemodulasi sinyal yang tertangkap dengan oleh antenna[4].

RFID bekerja dengan metode *backscatter modulation*. *Backscatter* merupakan salah satu skema modulasi yang secara luas digunakan untuk modulasi data ke RF *carrier*. Dalam metode ini, RF *carrier* akan dirubah amplitudonya berdasarkan urutan kode data sehingga berbentuk seperti diagram di bawah. Pembaca tag akan mendeteksi perubahan amplitudo pada sinyal *carrier* yang termulasi dan mereka ulang data tersebut.



Gambar 2.2 Sinyal *carrier* termulasi dengan metode *backscatter*[3]

Gambar di atas menunjukkan visualisasi sederhana dari sinyal *carrier* termodulasi dari RFID *tag*. Seperti yang terlihat di diagram, angka biner dikodekan memodulasi pembawa RF. Angka 1 direpresentasikan dengan tingkat pembawa tinggi, dan 0 diwakili oleh tingkat pembawa rendah (*tag coil* didorong). Pembaca mendemodulasi sinyal untuk memulihkan data, dan perhatikan bahwa data ini masih dikodekan. Pembaca *men-decdo* data menggunakan decoder yang cocok, dan ke depan untuk diproses lebih lanjut ke komputer (atau *server backend*) [3].

2.3 RFID Reader MFRC522

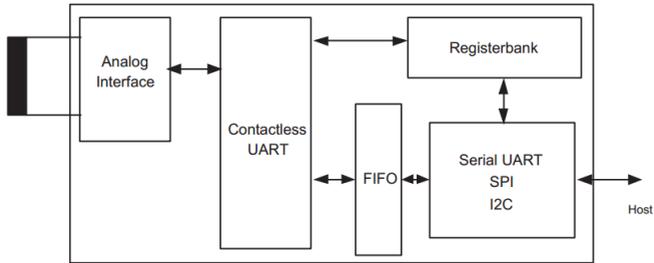
MFRC522 merupakan modul *reader* dan *writer* RFID yang terintegrasi untuk komunikasi tanpa kontak langsung pada frekuensi 13.56 MHz. Berdasarkan frekuensi kerjanya modul ini merupakan modul RFID *High Frequency*. Modul ini mendukung ISO 14443A / MIFARE[®] *mode*. Modul ini mendukung semua varia dari MIFARE Mini, MIFARE 1K, MIFARE 4K, MIFARE Ultralight, MIFARE DESFire EV1 and MIFARE Plus RF identification protocols. Modul ini memiliki jarak operasi dalam mode read/write mencapai 40 mm [5].

Transmitter internal pada IC MFRC522 dapat mengendalikan antena *writer/reader* yang didesain untuk berkomunikasi dengan kartu atau transponder ISO 14443A/MIFARE[®] tanpa tambahan rangkaian aktif lain. Bagian *receiver* pada ICI ini menyediakan implementasi yang kuat dan efisien dari rangkaian demodulasi dan *decoding* untuk sinyal dari kartu atau transponder ISO 14443A/MIFARE[®]. IC ini memiliki bagian digital yang berfungsi untuk menangani kesalahan deteksi. IC MFRC522 mendukung komunikasi menggunakan MIFARE[®] dengan kecepatan lebih tinggi mencapai 848 kbit/s untuk kedua arah[6].

Host interface yang didukung oleh IC MFRC522 antara lain adalah:

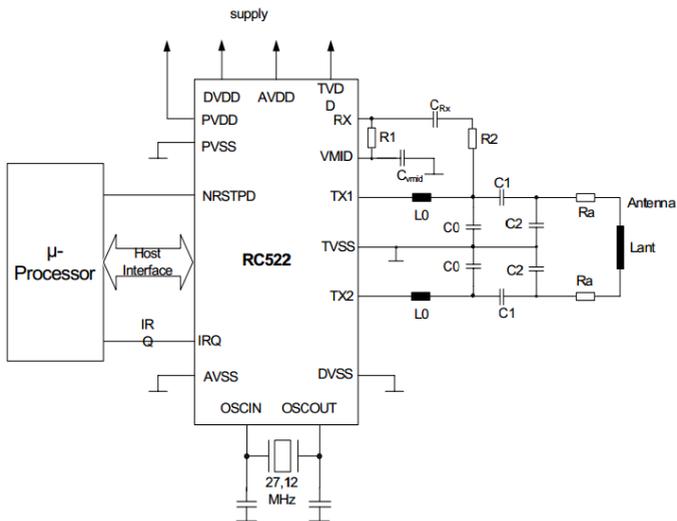
- SPI interface dengan kecepatan mencapai 10 Mbit/s
- Serial UART mencapai kecepatan 1228.8 kbit/s
- I²C mencapai 400 kbit/s pada mode Fast dan 3400 kbit/s dalam mode High-speed

IC beroperasi adalh 2.5 – 3.3 V dan memiliki osilator yang harus terhubung dengan osilator kristal 27.12 MHz[6].



Gambar 2. 3 Diagram blok IC MFRC522 yang disederhanakan[6]

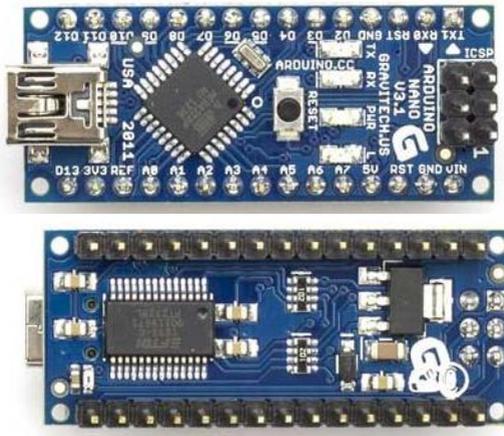
Analog Interface berfungsi untuk menangani modulasi dan demodulasi dari sinyal analog yang dikirim dan diterima oleh MFRC522. *Contactless UART* menangani protokol yang diperlukan untuk skema komunikasi dalam operasi bersama dengan *host*. *Host* disini adalah mikrokontroler yang bertindak sebagai master dalam pengoperasian modul ini. *FIFO buffer* memungkinkan untuk transfer data yang cepat dan tepat dari *host* ke *contactless UART*. Gambar 2.4 merupakan diagram rangkaian untuk mengaplikasikan IC MFRC522[6].



Gambar 2. 4 Diagram rangkaian MFRC522[6]

2.4 Arduino Nano

Arduino Nano adalah board berbasis mikrokontroler pada ATmega328. *Board* ini memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai *output PWM*), 8 input *analog*, 16 MHz *osilator* kristal, koneksi USB, jack listrik dan tombol reset. Semua pin digital dapat menyediakan atau menerima arus maksimum 40 mA dan dilegkapi dengan *internal pull-up resistor* sebesar 20-50Kohm. Beberapa pin memiliki fungsi khusus, antara lain Serial, SPI, PWM, dan eksternal *interrupt*. Analog input pada Arduino Nano beresolusi 10 bit, pin 7 tidak dapat digunakan untuk digital I/O[7].



Gambar 2.5 Board Arduino Nano[7]

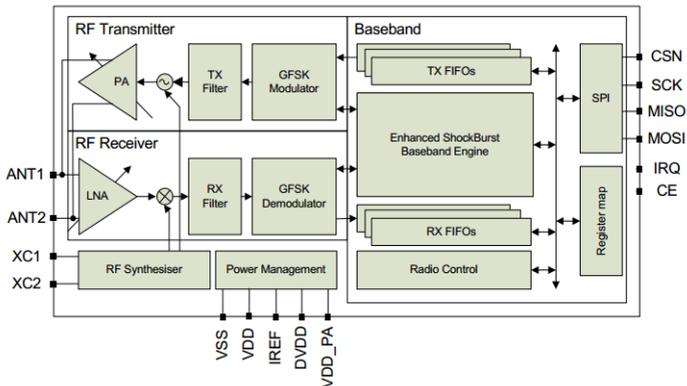
Spesifikasi arduino nano :

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8

DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"
Length	45 mm
Width	18 mm
Weigth	5 g

2.5 RF Transceiver NRF24L01

NRF24L01 merupakan chip tunggal *transceiver* Radio Frekuensi (RF) 2.4 GHz dengan mesin protokol baseband yang terpadu (Enhanced ShockBurst™), yang dirancang untuk aplikasi nirkabel daya ultra rendah. *Transceiver* ini dirancang untuk dapat beroperasi pada pita frekuensi ISM di 2.400 - 2.4835GHz. untuk merancang sistem radio dengan menggunakan nRF24L01 hanya memerlukan sebuah mikrokontroler dan sedikit komponen pasif pendukung lainnya[8].



Gambar 2.6 Diagram block nRF24L01[8]

NRF24L01 dikonfigurasi dan dioperasikan melalui Serial Peripheral Interface (SPI). Melalui antarmuka ini, peta register yang memuat semua register konfigurasi dalam nRF24L01 bisa diakses dalam semua mode operasi[6]. Chip ini mudah ditemui di pasaran sudah dalam bentuk modul. Library dan contoh yang mendukung pengaplikasiannya juga mudah dijumpai di internet.

Featur dari nRF24L01 antara lain[8] :

Radio:

- >Worldwide 2.4GHz ISM band operation
- >126 RF channels
- >Common RX and TX pins
- > GFSK modulation
- > 1 and 2Mbps air data rate
- > 1MHz non-overlapping channel spacing at 1Mbps
- > 2MHz non-overlapping channel spacing at 2Mbps
- Transmitter
 - > Programmable output power: 0, -6, -12 or -18dBm
 - > 11.3mA at 0dBm output power
- Receiver
 - > Integrated channel filters
 - > 12.3mA at 2Mbps
 - > -82dBm sensitivity at 2Mbps
 - > -85dBm sensitivity at 1Mbps
 - > Programmable LNA gain
- Power Management
 - > Integrated voltage regulator
 - > 1.9 to 3.6V supply range
 - > Idle modes with fast start-up times for advanced power management
 - > 22uA Standby-I mode, 900nA power down mode
 - > Max 1.5ms start-up from power down mode
 - > Max 130us start-up from standby-I mode
- Host Interface
 - > 4-pin hardware SPI
 - > Max 8Mbps
 - > 3 separate 32 bytes TX and RX FIFOs
 - > 5V tolerant inputs

2.6 MySQL Database

Database adalah sebuah sistem yang di buat untuk mengorganisasi, menyimpan dan menarik data dengan mudah. Database terdiri dari kumpulan data yang terorganisir untuk 1 atau lebih penggunaan, dalam bentuk digital. Database digital di manage menggunakan Database Management System (DBMS), yang menyimpan isi database, mengizinkan pembuatan dan maintenance data dan pencarian dan akses yang lain. Beberapa Database yang ada saat ini adalah : Mysql, Sql Server, Ms.Access, Oracle, dan PostgreSQL.

MySQL adalah salah satu jenis database *server* yang paling umum digunakan. MySQL menggunakan bahasa SQL untuk mengakses database nya. Lisensi Mysql adalah FOSS License Exception dan ada juga yang versi komersial nya. MySQL tersedia untuk beberapa platform, di antara nya adalah untuk versi windows dan versi linux.

2.7 PHP

PHP adalah singkatan dari PHP : Hypertext Processor. PHP merupakan bahasa pemrograman script *server-side* untuk pengembangan *web*. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>. PHP disebut bahasa program *server-side* karena PHP diproses di komputerr *server*.

PHP dapat digunakan dengan gratis (free) dan bersifat *Open Source*. PHP dirilis dalam lisensi *PHP License*, sedikit berbeda dengan lisensi *GNU General Public License (GPL)* yang biasa digunakan untuk proyek *Open Source*[9].

2.8 WebServer

Webserver merupakan software yang memberikan layanan data yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan browser *web* dan mengirimkan kembali hasilnya dalam bentuk halaman - halaman *web* yang umumnya berbentuk dokumen HTML. Beberapa macam *WebServer* diantaranya:

1. *Apache*
2. Microsoft windows *Server* 2003 Internet Information Services (IIS)
3. *Lighttpd*

Apache merupakan *webserver* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk

sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya *Apache* mengeluarkan programnya yang dapat dijalankan di Windows NT. *Apache* mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya.

BAB III

PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang perancangan sistem pelayanan pembayaran parkir prabayar untuk parkir di tepi jalan umum menggunakan RFID. Sistem ini terdiri dari tiga bagian utama, yakni *reader*, *server* lokal, dan *web-server*.

Reader berfungsi untuk mengidentifikasi ID dari tag yang dipasang pada kendaraan, baik motor atau mobil. Reader ini berupa alat *portable* yang dioperasikan oleh operator atau seorang tukang parkir. pembacaan dilakukan pada saat kendaraan akan parkir dan saat akan keluar dari area parkir. hasil pembacaan berupa ID dan kode perintah parkir “*park*” dan “*unpark*”. Kemudian kode tersebut dikirimkan ke Raspberry-Pi dengan komunikasi radio.

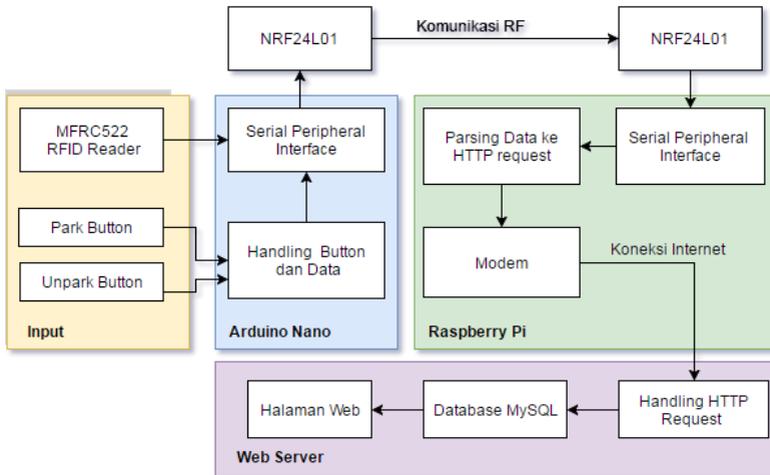
Kemudian, ID dan kode tersebut diterima oleh *server* lokal Raspberry-Pi untuk kemudian dikirimkan ke *server* menggunakan HTTP *requests*. *Server* lokal ini berperan sebagai tempat penerimaan ID dan kode dari *reader* yang kemudian di dikirimkan ke *web server*. Dalam sistem ini memungkinkan dioperasikannya beberapa *reader* untuk satu *server* lokal dalam suatu area.

Server merupakan *server web* dengan *database*. *Server* berfungsi sebagai penyimpanan data base kendaraan, seperti nomor kendaraan, pemilik, saldo, dan sebagainya. *Server* juga berfungsi sebagai peghitung biaya parkir yang dikenakan dan juga mencatat kegiatan semua aktivitas parkir dalam waktu tertentu. *Server* juga berfungsi sebagai antarmuka bagi user untuk melihat catatan kegiatan parkir kendaraannya sebagai bukti transaksi saldo.

Karena sistem ini dirancang untuk pelayanan parkir di tepi jalan umum maka sistem ini tidak lepas dari peran juru parkir. Dimana setiap juru parkir dapat memegang satu *reader* sebagai operatornya. Juru parkir tersebut yang akan mengoperasikan *reader* untuk layanan parkir di tepi jalan. Selain itu juru parkir memang tetap diperlukan untuk keperluan ketertiban.

3.1 Diagram Blok Sistem

Diagram blok sistem pelayanan pembayaran parkir prabayar untuk parkir di tepi jalan umum adalah sebagaia berikut:



Gambar 3.1 Diagram blok sistem

Seperti yang ditunjukkan gambar 3.1, sistem ini menerima *input* dari modul pembaca RFID MFRC522 dan 2 tombol, yakni tombol “park” dan “unpark”. *Input* tersebut diproses menggunakan mikrokontroler Atmega328 dengan *board* Arduino Nano. ID hasil baca modul MFRC522 dan *input* tombol dikirimkan ke Raspberry Pi melalui komunikasi frekuensi radio (RF) menggunakan modul NRF24L01. Pada Raspberry data tersebut disisipkan ke *HTTP request* dan dikirim ke server melalui koneksi internet.

Server akan menerima ID kendaraan dan kode perintah “park” atau unpark dari *HTTP request*. *Server* mengakses *database* MySQL untuk memanggil data kendaraan dengan ID tersebut untuk mengeksekusi kode perintah. Kemudian hasil proses tersebut akan ditampilkan ke halaman *website*.

3.2 Perancangan *Reader*

Perancangan *reader* ini meliputi perancangan perangkat keras dan perancangan perangkat lunak.

3.2.1 Perancangan Perangkat Keras

Perangkat keras *Reader* ini terdiri dari mikrokontroler Arduino Nano, pembaca RFID MFRC522, RF *transceiver* NRF24L01, dan input tombol.

Modul MFRC522 dan NRF24L01 menggunakan catu daya 3.3 Volt yang terhubung ke pin 3.3 V pada Arduino Nano. MFRC522 dan NRF24L01 terhubung ke arduino menggunakan komunikasi *Serial Pheriperal Interface* (SPI). Arduino hanya memiliki satu SPI sehingga keduanya harus terhubung secara bus. Pin MISO, MOSI, dan SCK keduanya terhubung ke satu pin MISO, MOSI, dan SCK pada arduino. Keduanya tidak bisa beroperasi secara bersamaan. Hanya satu perangkat saja yang dapat terhubung ke SPI arduino dalam satu waktu. Disinilah guna *Chip Select* (CS) pada masing masing device untuk memilih device mana yang akan dioperasikan. Adapun konfigurasi pin untuk arduino adalah :

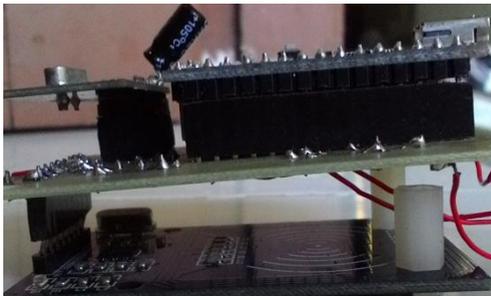
Tabel 3.1 Konfigurasi pin untunk SPI Arduino Nano

Pin Arduino	NRF24L01	MFRC522
3.3V	Vcc	Vcc
GND	GND	GND
7	CE	--
8	CSN	--
9	--	SS
10	--	RESET
11	MOSI	MOSI
12	MISO	MISO
13	SCK	SCK

Adapun bentuk fisik dari peragkat RFID *reader* yang dipakai dalam sistem inidapat dilihat pada gambar 3.2 dan gambar 3.3. Perangkat tersebut tersusun atas board arduino Nano, modul NRF24L01 dan Modul RFID *reader* MFRC522 yang disusun pada sebuah papan PCB. Pada modul NRF24L01 ditambahkan dengan kapasitor 10 uF pada pin catu dayanya untuk menyetabilkan tegangan catu dayanya saat modul beroperasi.



Gambar 3.2 Board Reader tampak atas



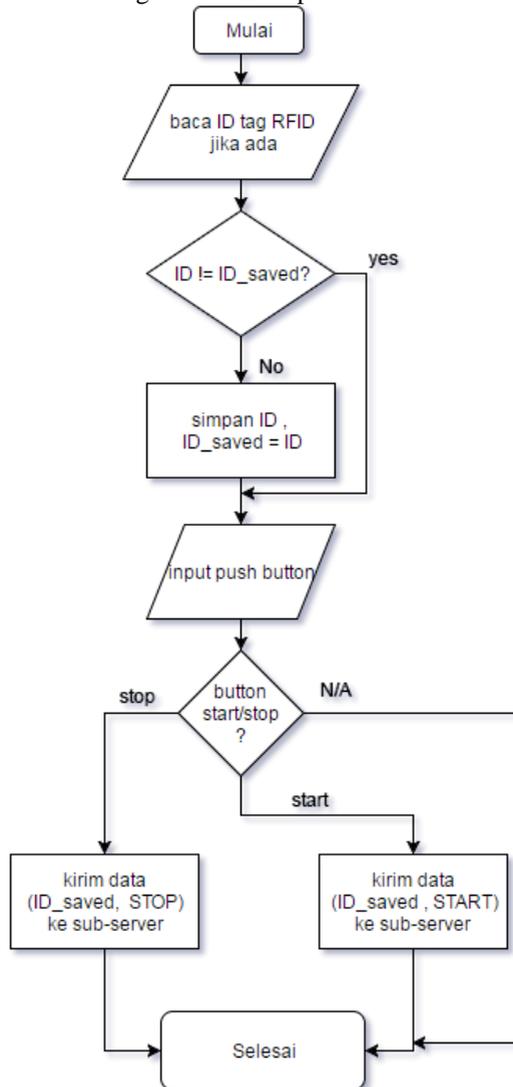
Gambar 3.3 Board Reader tampak dari samping

3.2.2 Perancangan Program Arduino Nano

Dalam proses kerjanya, pembaca RFID akan terus aktif menunggu adanya kartu yang masuk ke area pembacaannya. Area pembacaan dari pembaca RFID ini kurang lebih sekitar 5 cm dari antenanya. Setelah ada tag RFID yang masuk ke area pembacaan, pembaca RFID akan mengidentifikasi ID dari tag dan mengirimkannya ke arduino. kemudian disimpan oleh arduino dalam sebuah variable string. ID dari tag ini sebenarnya berupa 4 byte bilangan heksadesimal. ID tersebut disimpan hanya sementara, sampai ID dikirim ke *server* atau dilakukan identifikasi tag baru.

Setelah ID tersimpan, arduino menunggu perintah dari operator melalui tombol. Ada dua tombol perintah pada kontaktor yakni perintah untuk memulai parkir dan perintah untuk mengakhiri layanan parkir. Ketika salah satu tombol tersebut ditekan pembaca RFID akan terputus dari bus SPI (SS dirubah dari LOW ke HIGH) dan NRF24 akan

terhubung ke SPI (CSN diubah dari HIGH ke LOW). Lalu arduino akan mengirimkan ID dari tag beserta kode perintah dan ID dari *reader*.



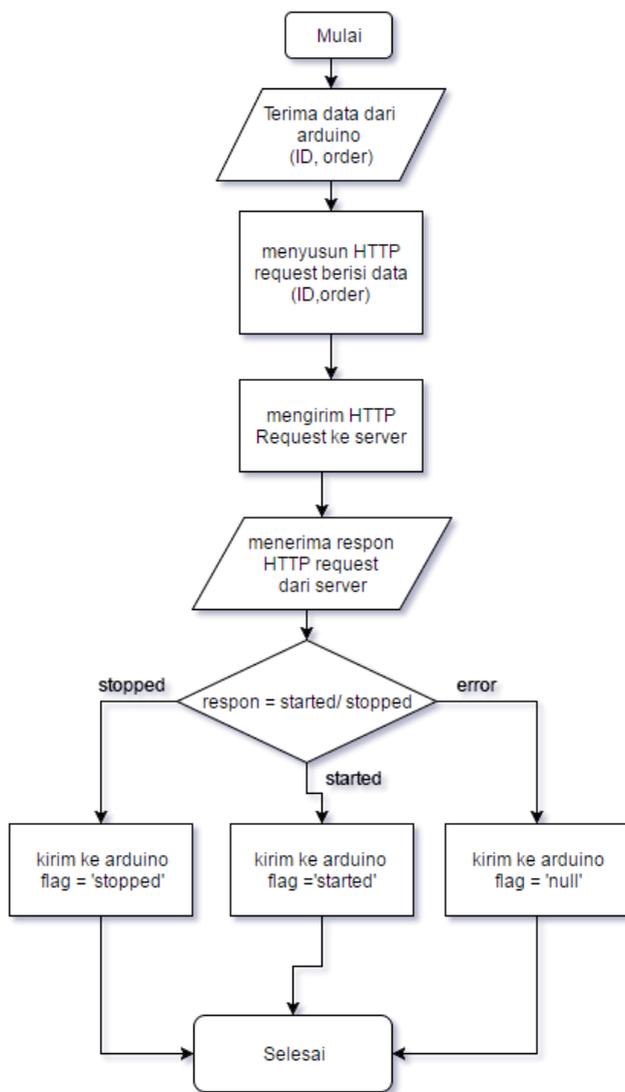
Gambar 3.4 Flow chart program pada Arduino

3.3 Perancangan *Sub-server*

Sub-server dalam sistem ini menggunakan *raspberry pi 2* yang terhubung dengan sebuah RF *transceiver* NRF24L01 dan koneksi internet. Fungsi utamanya adalah menerima ID dan kode perintah dari arduino melalui NRF24L01 dan menyusunnya menjadi sebuah HTTP *request* yang kemudian dikirim ke *server* dan menerima respon dari *server*. Dari respon tersebut kemudian di *sub-server* mengirimkan sinyal ke kontakor apakah proses suksse atau tidak. Proses yang berjalan pada *sub-server* ini menggunakan pemrograman *Python 2.7*.

Pada kondisi awalnya NRF24 pada *sub-server* akan tersiaga untuk menangkap ID dan kode perintah yang dikirim kontakor. Ketika ID dan kode perintah diterima, akan diproses dan disisipkan ke pesan HTTP *request* melalui koneksi internet. Kemudian pesan tersebut dikirimkan ke *server* untuk validasi ID dan eksekusi kode perintah, sehingga respon diterima dari *server*.

Program terbagi menjadi dua bagian, yakni program untuk menerima data dari *reader* melalui komunikasi radio dengan NRF24L01. Program ini menggunakan *library* “*lib_nrf24*” oleh Brian Lavery dari GitHub. Yang kedua adalah program untuk mengirim HTTP *request* yang berisi data dari *reader* yang telah diterima. Program ini menggunakan *library* “*urllib*” bawaan Python. Program meliputi *parsing* data ID dan *post request* ke *web-sever*. Berikut adalah *flow chart* dari program yang ada pada Raspberry-Pi:



Gambar 3.5 Flow chart sistem pada *sub-server*

3.4 Perancangan Web Server

Gambar 3.6 merupakan *flow chart* program pada *web server*. *Webserver* dalam sistem ini merupakan *webserver* yang dirancang khusus untuk berkomunikasi dengan *sub-server* melalui *HTTP request*. *Webserver* ini memiliki database yang berisikan informasi terkait pelayanan parkir di tepi jalan umum, seperti data keanggotaan, catatan kegiatan parkir untuk kurun waktu tertentu, dan catatan biaya parkir yang harus dibayar oleh pengguna layanan. *Web-server* menggunakan bahasa program PHP dengan database MySQL dan tampilan HTML.

3.4.1 Perancangan Program PHP Pada Server

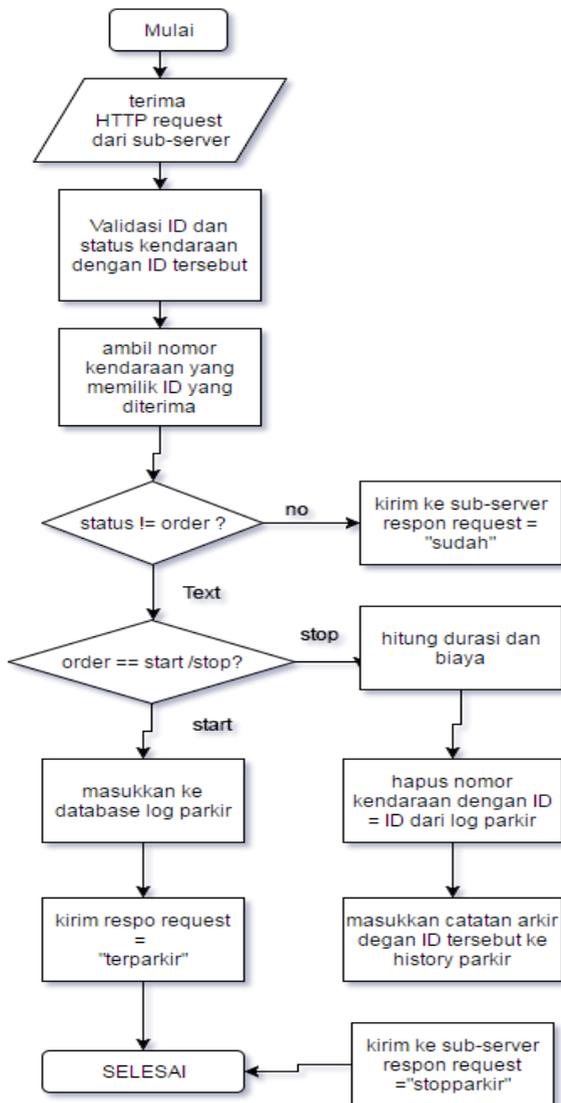
Program PHP pada serverterbagi mejadi 2 program utama , yakni program *handling HTTP request* dan program penghitungan biaya. Program Handlig *HTTP request* digunakan untuk menerima data dari Raspberry Pi dan mengeksekusinya. Data yang diterima dari Raspberry Pi merupakan data ID dan kode perintah layanan parkir.

Ketika dataditerima, program akan memanggil data dari *database* MySQL yang ID-nya sama dengan ID yang diterima. Data yang diambil berupa nomor kendaraan, nama pemilik, saldo,dan status keparkiran. Jika status keparkiran berbeda dengan kode perintah, dan jika kode perintah adalah “park”, maka data tersebut akan dimasukkan ke *database* log parkir dan status pada *database* data *member* diubah menjadi “park” sebagai tanda sedang meggunakan layanan parkir. Jika kode perintah adalah “stop”, maka data kendaraan yang bersangkutan akan disalin dari *database* log parkir ke riwayat parkir sebagai catatan kemudian dihapus dan status pada data *member* akan diubah menjadi “stop” sebagai tanda layanan parkir berhenti.

Berikutnya adalah program penghitungan biaya. Pada sistem ini, biaya parkir yang dikenakan pada pengguna layanan dihitung berdasarkandurasi lamanya penggunaan layanan parkir.Penghitungan biaya biaya ini dilakukan dan ditampilkan pada halaman log parkir yang menampilkan aktifitas parkir yang sedang berlangsung termasuk biaya. Biayajuga dihitungpada saat proses penghentian layanan parkir untuk melakukan pengurangan saldo sejumlah biaya yang terhitung.

Tabel 3.2 Asumsi harga pada sistem penghitung biaya

Biaya	Motor (Rp)	Mobil(Rp)
Per Jam	200	500
Charge	500	1000



Gambar 3.6 Flow chart server

3.4.2 Perancangan Antarmuka Web

Web server ini menggunakan domain dan server host gratis dari Hostinger (url: <https://hostinger.co.id>). Adapun alamat web server dari sistem ini adalah “<http://parkirjalanan.esy.es/> “. Web server ini terdiri dari beberapa halaman antara lain halaman data member, log parkir, history parkir, dan entry member.

Data Member

No Kendaraan	Jenis	Nama Pemilik	Telepon	Saldo	Status
1234	mobil	pajo		20100	unpark
2331	Mobil	kurniawan	032145678910	64100	parking
3456	Motor	timin		9600	unpark
45557	Mobil	kevin	089898989090	18500	unpark
5171	Motor	ruli huda	089981677000	28200	unpark

Gambar 3.7 Rancangan tampilan halaman data member

Halaman data member menampilkan informasi yang ada pada tabel data member pada database. Tampilannya dapat dilihat pada gambar 3.7. Kolom status akan berubah sesuai dengan status keparkiran dari kendaraan tersebut. *History* parkir seperti pada gambar 3.8 merupakan halaman *web* yang menampilkan catatan kegiatan parkir yang telah berlangsung dalam kurun waktu tertentu. Halaman ini menampilkan kegiatan parkir oleh semua kendaraan eserta waktu parkir, lokasi, durasi, dan biaya.

Sedangkan log parkir (gambar 3.9) merupakan halaman yang berisikan informasi kegiatan parkir yang sedang berlangsung. Halaman ini terhubung ke database log parkir yang isinya akan segera terhapus setelah pelayanan parkir berakhir. Dan informasi yang terhapus tersebut seelumnya disalin ke database *history* parkir.

History Parkir

Mulai Parkir	Lokasi	Jenis	No Kendaraan	Selesai Parkir	Durasi	Biaya
0000-00-00 00:00:00	B3	Motor	3456	2016-12-17 17:01:19	00:00:17	82331600
2016-12-17 17:06:34	B3	motor	7474	2016-12-17 17:32:53	00:00:07	100
2016-12-17 17:42:38	B3	motor	7474	2016-12-17 17:42:58	07:00:20	0
2016-12-17 17:46:02	B3	motor	7474	2016-12-17 17:47:36	00:01:34	0
2016-12-17 17:48:31	B3	motor	7474	2016-12-17 17:48:37	00:00:06	500
2016-12-18 14:10:27	A2	mobil	1234	2016-12-18 15:02:26	00:51:59	1400
2016-12-18 14:10:56	A2	Motor	3456	2016-12-18 15:05:45	00:54:49	700
2016-12-18 14:11:42	A2	Motor	6789	2016-12-18 15:07:41	00:55:59	700
2016-12-18 14:12:20	A2	motor	7474	2016-12-18 15:08:29	00:56:09	700

Gambar 3.8 Rancangan halaman *history* parkir

[mber](#) | [login](#) | [contact](#)

Log Parkir

No Kendaraan	Jenis	Lokasi	Mulai Parkir	Durasi	Biaya
1234	mobil	A3	2016-12-18 20:32:10	00:02:49	1020
3456	Motor	A3	2016-12-18 20:31:10	00:03:50	510

Gambar 3.9 Rancangan halaman log parkir

Pendaftaran Member

UID :

No Kendaraan:

Nama Pemilik:

Telepone :

Saldo :

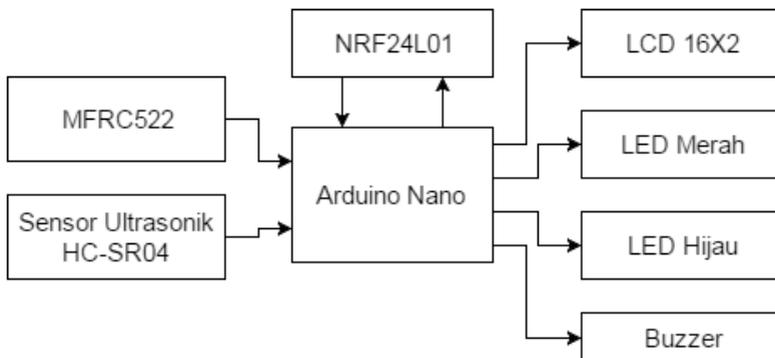
Jenis Kendaraan: Mobil Motor

Gambar 3.10 Rancangan halaman *entry member*

3.5 Perancangan Sistem RFID *Reader* untuk Pengoperasian Tanpa Operator

3.5.1 Peracagan Hardware

Sistem reader tanpa operator tidak berbeda jauh dengan sistem reader dengan operator. Komponen utama dari reader ini adalah Arduino Nano, modul RFID reader MFRC522, dan modul NRF24L01 untuk berkomunikasi dengan Raspberry Pi. Sistem RFID reader tanpa operator yang dirancang dilengkapi dengan sensor ultrasonik HC-SR04 untuk mendeteksi kehadiran kendaraan di area parkir tempat sistem reader diaplikasikan. Selain itu sistem reader ini juga memiliki antarmuka berupa LED dan LCD 16X2. Antarmuka tersebut berfungsi untuk mengindikasikan hubungan antara kehadiran kedaraan dan status keparkiran dari ID yang dipakai. Dan penggunaan push button untuk memilih perintah “park” dan “unpark” dihilangkan.



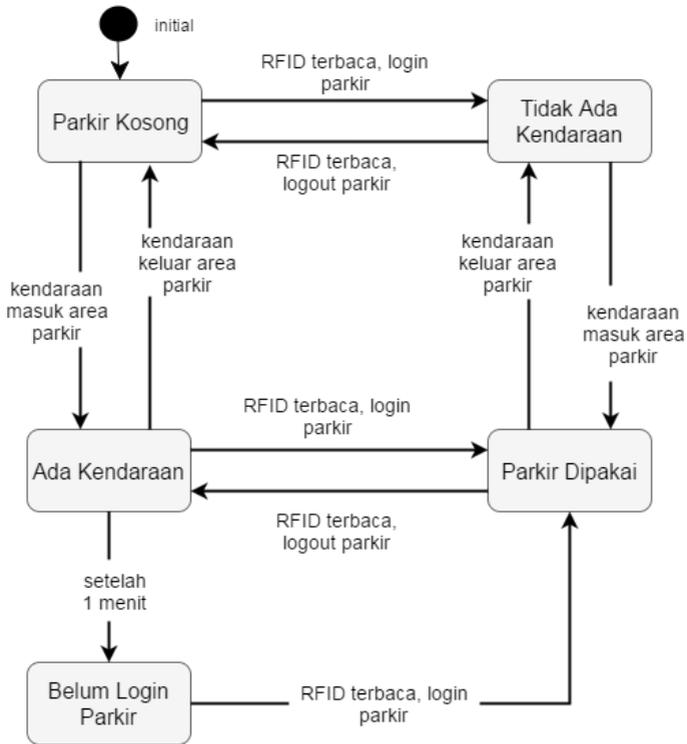
Gambar 3. 11 Diagram blok sistem RFID reader tanpa operator

3.5.2 Perancangan Software

Sistem ini bekerja berdasarkan *state* dari hubungan antara kehadiran kendaraan di area parkir dan status keparkiran ID yang digunakan. Setiap *state* tersebut diindikasikan dengan output interface berupa LED dan buzzer. Berikut merupakan tabel hubungan antara *state* dan *interface* tersebut adalah sebagai berikut :

Tabel 3. 3 Tabel hubungan *state* dan *interface* pada sistem RFID reader tanpa operator

No	State	Interface		
		LED Hijau	LED Merah	Buzzer
1	Parkir Kosong	mati	mati	mati
2	Ada Kendaraan	mati	nyala	Mati
3	Belum Login Parkir	mati	nyala	nyala
4	Parkir Dipakai	yala	mati	mati
5	Tidak Ada Kendaraan	nyala	nyala	nyala



Gambar 3. 12 State diagram sistem RFID reader tanpa operator

State awal dari sistem ini adalah state “parkir kosong”. Pada saat state ini, tidak ada kendaraan yang dideteksi oleh sensor ultrasonik dan belum ada login parkir. kemudian, state “ada kendaraan” terjadi saat kendaraan terdeteksi tapi belum ada ID yang terdaftar parkir. state ini akan berubah ke state parkir ilegal setelah 1 menit kendaraan terdeteksi dan belum ada ID masuk. State “kendaraan parkir” terjadi saat ada kendaraan terdeteksi dan ada ID yang terdaftar parkir. Dan state “Tidak Ditemukan kendaraan” terjadi ketika tidak ada kendaraan terdeteksi di area parkir namun ada ID yang terdaftar parkir.

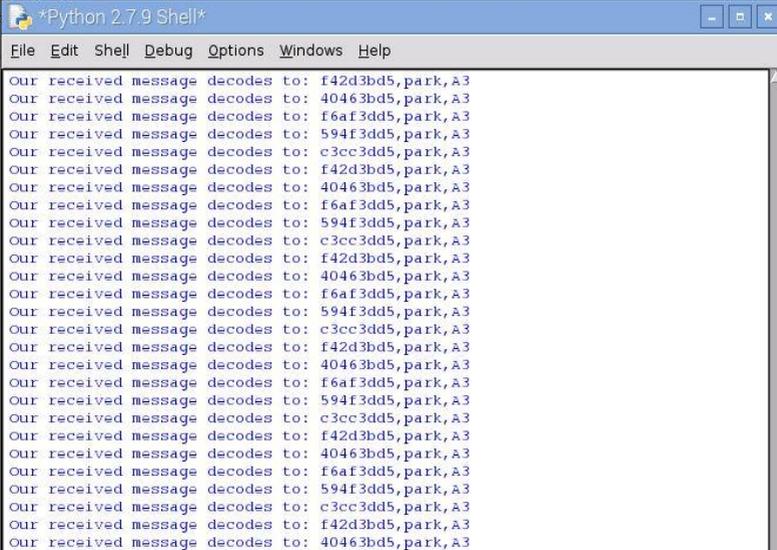
BAB IV

HASIL DAN PENGUJIAN

Pada bab ini akan dijelaskan hasil pengujian terhadap sistem pembayaran parkir prabayar yang meliputi sub bab pengujian sistem RFID *reader*, komunikasi antara kontaktor dan *sub-server* dan *WebServer* sistem parkir. Adapun penjelasan detail pengujian tersebut adalah sebagai berikut.

4.1 Pengujian RFID Reader

Tujuan dari pengujian sistem pembaca RFID ini adalah untuk mengetahui apakah sistem pembaca sudah bekerja sesuai yang diharapkan, yakni mampu membaca ID *tag* dan mengemasnya menjadi kode data untuk dikirim ke *sub-server*.



```
*Python 2.7.9 Shell*
File Edit Shell Debug Options Windows Help
Our received message decodes to: f42d3bd5, park, A3
Our received message decodes to: 40463bd5, park, A3
Our received message decodes to: f6af3dd5, park, A3
Our received message decodes to: 594f3dd5, park, A3
Our received message decodes to: c3cc3dd5, park, A3
Our received message decodes to: f42d3bd5, park, A3
Our received message decodes to: 40463bd5, park, A3
Our received message decodes to: f6af3dd5, park, A3
Our received message decodes to: 594f3dd5, park, A3
Our received message decodes to: c3cc3dd5, park, A3
Our received message decodes to: f42d3bd5, park, A3
Our received message decodes to: 40463bd5, park, A3
Our received message decodes to: f6af3dd5, park, A3
Our received message decodes to: 594f3dd5, park, A3
Our received message decodes to: c3cc3dd5, park, A3
Our received message decodes to: f42d3bd5, park, A3
Our received message decodes to: 40463bd5, park, A3
Our received message decodes to: f6af3dd5, park, A3
Our received message decodes to: 594f3dd5, park, A3
Our received message decodes to: c3cc3dd5, park, A3
Our received message decodes to: f42d3bd5, park, A3
Our received message decodes to: 40463bd5, park, A3
Our received message decodes to: f6af3dd5, park, A3
Our received message decodes to: 594f3dd5, park, A3
Our received message decodes to: c3cc3dd5, park, A3
Our received message decodes to: f42d3bd5, park, A3
Our received message decodes to: 40463bd5, park, A3
```

Gambar 4.1 Tampilan IDLE *Python* menampilkan data dari pembaca RFID

Proses pengujian ini dilakukan dengan mendekati RFID tag ke sistem pembaca RFID dan hasil pembacaan akan dikirimkan *sub-server* dan ditampilkan di kosol *sub-server*. Sistem pembaca RFID ini adalah

modul MFRC522, menggunakan pengendali utama mikrokontroler ATMEGA 328, dilengkapi dengan push button perintah dan terhubung ke *sub-server* menggunakan NRF24L01. Pengujian dilakukan pada 5 tag yang berbeda dan diulang sepuluh kali setiap tag.

Gambar 4.1 merupakan cuplikan gambar dari konsol Python pada Raspberry Pi saat berjalannya pengujian. Padagambar tersebut terlihat bahwa sistem pembacaRFID telah bisa membaca dan mengirimkan ID dari tag dalam bentuk kode data. Adapun hasil pengujiannya dapat dilihat pada tabel berikut:

Tabel 4.1 Hasil pengujian pembacaan tag RFID dan pengiriman ke Raspberry Pi dengan NRF24L01

No	ID	Jumlah pengujian	Berhasil	Gagal	Keberhasilan (%)
1	C3 CC 3D D5	10	10	-	100
2	F4 2D 3B D5	10	10	-	100
3	40 46 3B D5	10	10	-	100
4	59 4F 3D D5	10	10	-	100
5	F6 AF 3D D5	10	10	-	100

Dari data di atas dapat diketahui bahwa pembacaan RFID dan pengiriman data ID tersebut melalui komunikasi radio NRF24L01 memiliki tingkat keberhasilan yang tinggi.

4.2 Pengujian *WebServer* sistem parkir

Pengujian ini dilakukan untuk mengetahui apakah *webserver* telah berjalan dengan seharusnya. Pengujian ini meliputi pengujian sistem penghitungan biaya dan uji menampilkan data.

Pengujian dilakukan dengan mengakses halaman *web* yang bersangkutan dengan perhitungan durasi dan biaya dari *website* sistem, kemudian dilakukan penghentian hitungan untuk kurun waktu yang acak. *Server* yang digunakan adalah *serverApache* dengan bahasa program PHP dan database MySQL.

4.2.1 Pengujian penghitungan durasi dan biaya parkir

Pengujian dilakukan dengan pengambilan *screen shoot* pada halaman log parkir dengan kurun waktu yang acak seperti pada gambar dibawah ini.

Log Parkir

No Kendaraan	Jenis	Lokasi	Mulai Parkir	Durasi	Biaya
1234	mobil	A3	2016-12-18 20:32:10	05:02:34	15520
3456	Motor	A3	2016-12-18 20:31:10	05:03:34	6310

Log Parkir

No Kendaraan	ID	Jenis	Lokasi	Mulai Parkir	Durasi	Biaya
1234	e31f8b00	mobil	A3	2016-12-18 20:32:10	05:56:01	15970
3456	f6af3dd5	Motor	A3	2016-12-18 20:31:10	05:57:02	6490

Gambar 4.2 Hasil *screen shoot* dari halaman log parkir pada pegujian penghitung durasi dan biaya parkir

Hasil dari pengujian ini didapatkan data:

Tabel 4.2 Hasil uji biaya dan durasi untuk mobil

No	Durasi (mnt)	Biaya (Rp)	Biaya/Meni (Rp)
1	1740	15500	8,33
2	1742	15520	8,34
3	1765	15710	8,33
4	1772	15770	8,34
5	1781	15840	8,33
6	1784	15860	8,33
7	1796	15970	8,34

Tabel 4.3 Hasil uji durasi dan biaya untuk motor

No	Durasi (mnt)	Biaya (Rp)	Biaya/Menit (Rp)
1	1741	6300	3,33
2	1743	6310	3,33
3	1766	6390	3,34
4	1773	6410	3,33
5	1782	6440	3,33
6	1785	6450	3,33
7	1797	6490	3,33

Pengambilan data dilakukan pada tanggal 20-12-2016, sehinggadaselesih 1hari, nilai konversi darijam ke menit ditambahkan jumlah menit dalam 1 hari (1440 menit). Dari kedua tabel hasil tersebut dapat dilihat bahwa nilai biaya permenitnya sangat stabil, hampir sama.

4.2.2 Pengujian Halaman *Entry* Member

Pengujian bertujuan untuk megetahui apakah halaman *entry* member sudah bekerja semestinya. Pengujian ini dilakukan dengan melakukan pendaftaran member baru lalu dilihat apakah datamember baruberhasilmasuk kedata base. Hasil pengujiannya dapat dilihat pada gambar 4.2 dan gambar 4.3. Pada gamar 4.2 dilakukan proses pendaftaran dan gambar 4.3 menunjukkan data member baru telah masuk ke database. Ini menunjukkan bahwa halaman pendaftaran sudah bekerja dengan semestinya.

Pendaftaran Member

UID :

No Kendaraan:

Nama Pemilik:

Telepone :

Saldo :

Jenis Kendaraan: Mobil Motor

:

Gambar 4.3 Proses pendaftaran member

Data Member

No Kendaraan	Jenis	Nama Pemilik	Telepon	Saldo	Status
1234	mobil	paijo		20100	unpark
2331	Mobil	kurniawan	032145678910	64100	parking
3456	Motor	timin		9600	unpark
45557	Mobil	kevin	089898989090	20000	unpark
5171	Motor	ruli huda	089981677000	28200	unpark
6789	Motor	siti		-6600	unpark

Gambar 4.4 Member baru berhasil ditambahkan

4.3 Pengujian Sistem Parkir Terintegrasi

Pengujian ini merupakan pengujian sistem secara menyeluruh, mulaidari pembacaan ID tag RFID, pengiriman kode data ke *sub-server*,

HTTP *request*, dan perhitungan biaya. Pengujian ini bertujuan untuk mengetahui apakah sistem sudah bekerja sesuai yang diharapkan.

Pengujian dilakukan dengan simulasi start parkir dan stop parkir menggunakan sistem yang telah diintegrasikan setiap komponennya. Hasil dari proses pengujian ini ditampilkan menggunakan *IDLE(python GUI)* dan halaman *web* pada *server*.

4.3.1 Uji Proses Memulai Parkir

Yang dilakukan adalah pembacaan ID tag oleh RFID *reader* yang sudah terintegrasi ke *webservice* melalui *sub-server*. *Sub-server* merupakan mini komputer Raspberry Pi dengan atarmuka program *Python* yang menjembatani antara RFID *reader* dan *server*. Hasil dari pengujian ditampilkan pada konsol *IDLE Python* dan perubahan log parkir ang dapat dilihat pada gambardi bawah :

```
>>>
STATUS      = 0x0e RX_DR=0 TX_DS=0 MAX_RT=0 RX_P_NO=7 TX_FULL=0
RX_ADDR_P0-1 = 0xf0f0f0f0e1 0xc2c2c2c2
RX_ADDR_P2-5 = 0xc3 0xc4 0xc5 0xc6
TX_ADDR     = 0xe7e7e7e7e7
RX_PW_P0-6  = 0x20 0x00 0x00 0x00 0x00 0x00
EN_AA      = 0x3f
EN_RXADDR  = 0x03
RF_CH      = 0x76
RF_SETUP   = 0x01
CONFIG     = 0x0c
DYNPD/FEATURE = 0x3f 0x06
Data Rate  = 1MBPS
Model      = nRF24101+
CRC Length = 16 bits
PA Power   = PA_MIN
Received: [102, 54, 97, 102, 51, 100, 100, 53, 44, 112, 97, 114, 107,
, 44, 65, 51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Translating our received Message into unicode characters...
Our received message decodes to: f6af3dd5,park,A3
loaded payload reply of[1]
posting nuid . . .
respon nuid:connectedberhasil_masukin_ke LOGkendaraan dengan noPol =
'3456' terparkir.
```

Gambar 4.5 Konsol *sub-server*

Pada gambar 4.4 di atas berisikan laporan penerimaan data dari RFID *reader* berupa string “ f6af3dd5,park,A3” yang kemudian di masukkan ke HTTP *request* dan dikirm ke *sever*. Kemudian *sub-server* menerima respon “... nopol='3456' terparkir” yang menunjukkan bahwa layanan parkir telah dimulai. Kemudian berikut adalah perubahan yang terjadi pada *server* :

Log Parkir

No Kendaraan	ID	Jenis	Lokasi	Mulai Parkir	Durasi	Biaya
1234	e31f8b00	mobil	A3	2016-12-18 20:32:10	08:50:20	17420
3456	f6af3dd5	Motor	A3	2016-12-20 05:21:40	00:00:50	500

Gambar 4. 6 ID kendaraan ditambahkan ke halaman Log Parkir

Data Member

No Kendaraan	Jenis	Nama Pemilik	Telepon	Saldo	Status
1234	mobil	paijo		20100	unpark
2331	Mobil	kurniawan	032145678910	64100	parking
3456	Motor	timin		9600	unpark
45557	Mobil	kevin	089898989090	20000	unpark
2331	Mobil	kurniawan	032145678910	64100	parking
3456	Motor	timin		9600	parking
45557	Mobil	kevin	089898989090	20000	unpark
5171	Motor	ruli huda	089981677000	28200	unpark

Gambar 4.7 Perubahan status pada halaman data member

HTTP *request* yang diterima oleh *server* yang berisi perintah ‘park’ akan dieksekusi dengan menambahkan ID yang bersangkutan ke log parkir kemudian diikuti berubahnya status keparkiran dalam data member.

4.3.2 Uji Proses Mengakhiri Parkir

Dilakukan lagi pembacaan ID *tag* target. Kemudian dikirim ke *sub-server*.

```

PA Power          = PA_MIN
Received: [102, 54, 97, 102, 51, 100, 100, 53, 44, 112, 97, 114, 107
, 44, 65, 51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Translating our received Message into unicode charatcters...
Our received message decodes to: f6af3dd5,park,A3
loaded payload reply of[1]
posting nuid . . .
respon nuid:connectedberhasil_masukin_ke LOGkendaraan dengan noPol =
'3456' stopparkir
Received: [102, 54, 97, 102, 51, 100, 100, 53, 44, 115, 116, 111, 11
2, 44, 65, 51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Translating our received Message into unicode charatcters...
Our received message decodes to: f6af3dd5,stop,A3
loaded payload reply of[1]
posting nuid . . .
respon nuid:connectedsukses_masukin_ke HISTORYkendaraan dengan noPol
='3456' stopparkir.sukses_hapus LOG 3456

```

Gambar 4.8 Tampilan konsole *sub-server* saat proses mengakhiri layanan parkir

Kode datayang diterima dari *RFIDreader* adalah “f6af3dd5,stop,A3”. Lalu respon yang diperoleh adalah “...nopol = ‘3456 stopparkir.sukses hapus log...” ini menandakan layanan parkir kepada ID tersebut telah selesai dan log telah dihapus. Perubahan yang terjadi pada *web server* :

Log Parkir

No Kendaraan	ID	Jenis	Lokasi	Mulai Parkir	Durasi	Biaya
1234	e31f8b00	mobil	A3	2016-12-18 20:32:10	08:52:41	17430

Gambar 4.9 Log parkir dengan ID “f6af3dd5” telah dihapus

2016-12-20 02:37:46	A3	Motor	5171	2016-12-20 02:40:35	00:02:49	500
2016-12-20 02:40:46	A3	Motor	5171	2016-12-20 02:47:01	00:06:15	500
2016-12-20 05:21:40	A3	Motor	3456	2016-12-20 05:24:23	00:02:43	500

Gambar 4.10 *History* parkir telah ditambah

Melihat dari hasil pengujian di atas, maka dapat dikatakan sistem sudah bisa berjalan degan semestinya. Kemudian pengujian memulai layanan parkir dan mengakhiri layanan parkir di atas diulangi untuk 5 ID tag berbeda sebanyak 5 kali per ID. Berikut merupakan cuplikan gambar dari halaman History parkir Setelah pengujian:

History Parkir

Mulai Parkir	Lokasi	Jenis	No Kendaraan	Selesai Parkir	Durasi	Biaya
2017-01-11 16:53:34	A3	motor	7474	2017-01-11 16:54:22	00:00:48	0
2017-01-11 16:50:20	A3	Motor	3456	2017-01-11 16:52:51	00:02:31	0
2017-01-11 16:50:15	A3	Mobil	2331	2017-01-11 16:52:44	00:02:29	0
2017-01-11 16:50:12	A3	Mobil	9090	2017-01-11 16:52:23	00:02:11	0
2017-01-11 16:51:34	A3	Mobil	45557	2017-01-11 16:52:17	00:00:43	0
2017-01-11 16:50:06	A3	motor	7474	2017-01-11 16:52:10	00:02:04	0
2017-01-11 16:45:19	A3	Motor	3456	2017-01-11 16:48:14	00:02:55	0
2017-01-11 16:43:42	A3	Mobil	9090	2017-01-11 16:48:01	00:04:19	0
2017-01-11 16:43:46	A3	Mobil	2331	2017-01-11 16:47:23	00:03:37	0
2017-01-11 16:43:41	A3	Mobil	45557	2017-01-11 16:46:10	00:02:29	0
2017-01-11 16:43:39	A3	motor	7474	2017-01-11 16:45:33	00:01:54	0

Gambar 4.11 Cuplikan gambar dari halaman History Parkir setelah pengujian

Gambar 4.11 menunjukkan adanya aktivitas parkir dengan durasi singkat dan berulang. Adapun hasil dari pengujian tersebut adalah sebagai berikut

Tabel 4.4 Hasil pengujian proses memulai layanan parkir

No	ID	Jumlah pengujian	Berhasil	Gagal	Keberhasilan (%)
1	C3 CC 3D D5	5	5	-	100
2	F4 2D 3B D5	5	4	1	80
3	40 46 3B D5	5	5	-	100
4	59 4F 3D D5	5	5	1	80
5	F6 AF 3D D5	5	4	-	100
Rata-rata keberhasilan (%)					92

Tabel 4.5 Hasil pengujian proses mengakhiri parkir

No	ID	Jumlah pengujian	Berhasil	Gagal	Keberhasilan (%)
1	C3 CC 3D D5	5	5	-	100
2	F4 2D 3B D5	5	5	-	100
3	40 46 3B D5	5	4	1	80
4	59 4F 3D D5	5	4	1	80
5	F6 AF 3D D5	5	3	2	60
Rata-rata keberhasilan (%)					84

Dari tabel 4.4 dan 4.5 terlihat bahwa terjadi beberapa kegagalan uji coba. Kegagalan tersebut dikarenakan gangguan koneksi internet pada Raspberry Pi sehingga menghambat pengiriman HTTP *request* ke *server*. Tapi meskipun ada error, rata-rata presentasi keberhasilannya masih tinggi.

4.4 Pengujian *Interface* Sistem RFID *reader* tanpa operator

Pengujian ini bertujuan untuk mengetahui apakah *interface* untuk RFID *reader* tanpa operator sudah dapat berjalan dengan semestinya sesuai dengan *state* yang terjadi. pengujian dilakukan dengan melakukan simulasi setiap *state* dan mengamati *output* dari *interface*.

Berikut merupakan beberapa cuplikan gambar hasil pengujian :

Tabel 4. 6 Cuplikan gambar pengujian *interface* Sistem RFID *reader* tanpa operator

No	State	Interface	ket
1	Parkir Kosong		LED merah mati LED hijau mati LCD: "parkir kosong"

No	State	Interface	ket
2	Ada Kendar aan		LED merah nyala LED hijau mati LCD:”ada kendaraan”
3	Belum Login Parkir		LED merah nyala LED hijau mati LCD:”belum login parkir” Buzzer nyala
4	Parkir Dipakai		LED merah mati LED hijau nyala LCD:”sedang dipakai”

No	State	Interface	ket
5	Tidak ada Kendar aan		LED merah nyala LED hijau nyala LCD: "tidak ada ada kendaraan" Buzzer nyala

Dari tabel 4.6 di atas dapat dilihat bahwa *output interface* sudah sesuai dengan *state* yang sedang aktif

BAB V

KESIMPULAN DAN SARAN

Dari penelitian yang dilakukan, didapatkan kesimpulan dan saran sebagai berikut :

5.1 Kesimpulan

- Tingkat keberhasilan pembacaan RFID dengan modul MFRC522 dan pengiriman hasil bacanya ke Raspberry Pi menggunakan modul NRF24L01 sangat tinggi, mencapai 100%.
- Proses sistem untuk memulai parkir dan mengakhiri parkir memiliki rata-rata tingkat keberhasilannya masing-masing 92% dan 84%. Tingkat keberhasilannya dipengaruhi oleh kecepatan koneksi internet dan *server host* yang dipakai.

5.2 Saran

Berikut adalah saran penulis untuk pengembangan sistem ini selanjutnya:

1. Ditambahkan sistem pengaman dengan bantuan sistem identifikasi lain seperti password atau sidik jari, atau menggunakan ID yang dinamis yang dirubah setiap kali selesai pembacaan.
2. Dari segi pengamankeandaraandi area parkir sistem ini bisa ditambahkan modul pelacak jarak dekat untuk memastikan kendaraan masih berada dalam area parkir saat pelayanan masih berjalan.
3. Sistem ini akan sangat bisa bekerja secara maksimal jika memiliki komputer server sendiri yang memadai untuk penggunaan secara luas.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1]. Peraturan Daerah Kota Surabaya Nomor 8 Tahun 2012 Tentang Retribusi Pelayanan Parkir Di Tepi Jalan Umum
- [2]. _____, "Parkir Di Pinggir Jalan" <URL: https://id.wikipedia.org/wiki/Parkir_di_pinggir_jalan>, Februari 2011
- [3]. Tutorials Web. "Operation of RFID Systems" <URL: <http://www.tutorialsworld.com/rfid/operation-of-rfid-systems.htm>>, (diakses tanggal 13 September 2016)
- [4]. Elektronika Dasar. "Pengertian Dan Komponen Radio Frequency Identification (RFID)", <URL: <http://elektronika.dasar.web.id/pengertian-dan-komponen-radio-frequency-identification-rfid/>>, (diakses tanggal 13 September 2016)
- [5]. ElecFreaks, "Module MFRC522" <URL: http://www.electronicsforu.com/wiki/index.php?title=MFRC522_Module>, 9 Mei 2015
- [6]. NXP Semiconductor. "MFRC522 Contactless Reader IC Rev. 3.2", 22 Mei 2007
- [7]. Arduino. "Arduino Nano", <URL: <https://www.arduino.cc/en/Main/ArduinoBoardNano>>, (diakses tanggal 13 September 2016)
- [8]. NORDIC Semiconductor. "Single Chip 2.4GHz Transceiver Product Specification", Juli 2007
- [9]. Andre, "Tutorial Belajar PHP Part 1: Pengertian dan Fungsi PHP dalam Pemrograman Web" <URL: <http://www.duniailkom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/>> Desember 2014

Halaman ini sengaja dikosongkan

LAMPIRAN

Program Pada Arduino:

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <stdint.h>
#include <MFRC522.h>
#define ce 7
#define csn 8
#define RST_PIN 9
#define SS_PIN 10
#define ledr 6
#define ledh 5
#define buz 2

int jarak; //
unsigned long interval1,mulaiAdaKend;
boolean adaKend,detKend,alarm,park,rfidRed,recFeed,nuidSent=0;
String stat=""; //
byte nuid[3]; //
String nuidStr,laporan,nuidPark //id bentuk string, kode data bentuk str,
respon dalam string

RF24 radio(ce,csn);
MFRC522 rfid(SS_PIN, RST_PIN); // Create MFRC522 instance.
MFRC522::MIFARE_Key key;

void setup() {
  park=0;
  nuidSent=0;
  // put your setup code here, to run once:
  Serial.begin(9600); // Open serial monitor at 115200 baud to see ping
  results.
  lcd.begin(16,2);
  lcd.print("Parkir Di Tepi Jalan");
  pinMode(ledh,1);
  pinMode(ledr,1);
  pinMode(buz,1);
  SPI.begin();
  rfid.PCD_Init(); // Init MFRC522 card
  for (byte i = 0; i < 6; i++) {
```

```

        key.keyByte[i] = 0xFF;
    }

    pinMode(SS_PIN, OUTPUT);
    digitalWrite(SS_PIN,HIGH) ;
    radio.begin();
    radio.setRetries(15,15);
    radio.setPALevel(RF24_PA_MIN);
    radio.setChannel(0x76);
    radio.openWritingPipe(0xF0F0F0F0E1LL);
    const uint64_t pipe = 0xE8E8F0F0E1LL;
    radio.openReadingPipe(0,pipe);
    radio.enableDynamicPayloads();
    radio.setAutoAck(true);
    radio.enableAckPayload();
    radio.powerUp();
    Serial.print("budal");
    pinMode(csn, OUTPUT);
    digitalWrite(csn,HIGH);
}

void loop() {

    getFeed();
    readNuid();
    delay(50);
}

void readNuid(){
    delay(10);
    digitalWrite(csn, HIGH);
    digitalWrite(SS_PIN,LOW);
    if ( ! rfid.PICC_IsNewCardPresent())
        return;
    // Verify if the NUID has been readed
    if ( ! rfid.PICC_ReadCardSerial())
        return;
    MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);

    if (rfid.uid.uidByte[0] != nuid[0] ||
        rfid.uid.uidByte[1] != nuid[1] ||
        rfid.uid.uidByte[2] != nuid[2] ||
        rfid.uid.uidByte[3] != nuid[3] ) {

```

```

nuidStr="";
// Store NUID into nuid array
for (byte i = 0; i < 4; i++) {
    nuid[i] = rfid.uid.uidByte[i];
    nuidStr+= String(rfid.uid.uidByte[i],HEX);
}
//perubahan semua status pada program
rfidRed = 1;
printHex(rfid.uid.uidByte, rfid.uid.size);
//Serial.println();
delay(1);
Serial.println(nuidStr);
delay(1);
}
if(park==1 and nuidPark!=nuidStr){
    Serial.print(F("nuid salah, unpark gagal"));
    delay(1);
    return;
}

rfid.PICC_HaltA();
// Stop encryption on PCD
rfid.PCD_StopCrypto1();
digitalWrite(csn, LOW);
digitalWrite(SS_PIN,HIGH);

//digitalWrite(SS_PIN,HIGH);
 kirimNRF();
}

void printHex(byte *buffer, byte bufferSize) {
    for (byte i = 0; i < bufferSize; i++) {
        Serial.print(buffer[i] < 0x10 ? " 0" : " ");
        Serial.print(buffer[i], HEX);
    }
}

//===== kirimNrf =====
void kirimNRF(){
    String temp="";
    char nuidSend[32]="";
    nuidSent=0;
    recFeed =0;

```

```

if (park==0){
  stat=",parking";
} else { stat=",unpark";}
const char mintaFeed[]="getfeed";
char respon[32]={0};
temp= nuidStr+stat+",A3";
temp.toCharArray(nuidSend,sizeof(nuidSend));
radio.stopListening();
radio.write(nuidSend,sizeof(nuidSend));
radio.startListening();
nuidSent=1;
//nuidStr="";
delay(1);
Serial.println(F("nuid send"));
Serial.println(nuidSend);
Serial.println(temp);
delay(1);
for (byte i = 0; i <4 ; i++) {
  nuid[i] = 0;}
  nuidPark=nuidStr;
  delay(100);
}

void getFeed(){
if (nuidSent==1){
  const char mintaFeed[]="getfeed";
char respon[32]={0};
if( recFeed==0){
  radio.stopListening();
radio.write(mintaFeed,sizeof(mintaFeed));
radio.startListening();
if(radio.available()){
  radio.read( &respon, sizeof(respon) );
  delay(1);
  Serial.print(F("respon="));
  Serial.println(respon);
String laporan(respon);
digitalWrite(buz,0);
delay(10);
digitalWrite(buz,1);
delay(10);
digitalWrite(buz,0);

```

```

delay(10);
digitalWrite(buz,1);
delay(10);
if(laporan=="parked"){
  //masukkan tone sukses
  park=1;
  delay(1);
  nuidPark=nuidStr;
  recFeed=1;
  nuidSent=0;
  lcd.clear();
  lcd.print("berhasil parkir");}
else if(laporan=="unparked"){
  //maukkan tone telolet
  park=0;
  delay(1);
  park=0;
  recFeed=1;
  nuidSent=0;
  nuidPark=nuidStr;
  lcd.clear();
  lcd.print("berhasil unpark");}
delay(1);
  Serial.println(park);
  delay(1);
}}}
delay(10);
}

```

Kode Program Pada Raspberry Pi:

```

#!/usr/bin/env python
def post(a,b,c):
  data = {'nuid':,"stat':"lokasi':"}
  data['nuid']= a
  data['stat']= b
  data['lokasi']= c
  parms = urllib.urlencode(data)
  #print parms
  u = urllib.urlopen("http://parkirjalanan.esy.es/check_nuid.php", parms)
  code = u.code
  res = u.read()

```

```

return res
while True:
    try:
        import RPi.GPIO as GPIO
        from lib_nrf24 import NRF24
        import time
        import spidev
        import urllib

        GPIO.setmode(GPIO.BCM)

        pipes = [[0XF0, 0XF0, 0XF0, 0XF0, 0XE1],[0xE8, 0XE8, 0XF0,
0XF0, 0XE1]]

        radio = NRF24(GPIO, spidev.SpiDev())
        radio.begin(0, 17)

        radio.setRetries(15,15)

        radio.setPayloadSize(32)
        radio.setChannel(0x76)
        radio.setDataRate(NRF24.BR_1MBPS)
        radio.setPALevel(NRF24.PA_MIN)
        radio.setAutoAck(True)
        radio.enableDynamicPayloads()
        radio.enableAckPayload()

        radio.openReadingPipe(0, pipes[0])
        radio.openWritingPipe(pipes[1])
        radio.printDetails()
        radio.startListening()

        command = ""
        respon=""
        pesan=[]
        while True:
            ackPL =[1]
            ackSt =[2]
            while not radio.available(0):
                time.sleep(1/100)

            receivedMessage = []
            radio.read(receivedMessage, radio.getDynamicPayloadSize())

```

```

#print("Received: {}".format(receivedMessage))

#print("Translating our received Message into unicode
charatcters...")
string = ""
radio.stopListening()

for n in receivedMessage:
    if (n >= 0 and n <= 126):
        string += chr(n)
print("Our received message decodes to: {}".format(string))

##print("Our received command decodes to:
{}".format(command[1]))
if ('in string):
    command = string.split(',')
    a = command[0]
    b= command[1]
    c = command[2]
    if len(a)<6:
        print "nuid salah"
        respon ="nuid salah"
    else:
        print("posting nuid . . .")
        respon =""
        res = post(a,b,c)
        print("respon nuid:{}".format(res))
        time.sleep(1/1000)

    if(res.find('yes')>0):
        respon="parked"
    elif('2'in res):
        respon="unparked"
    elif('parking' in res):
        respon="parked"
    elif('un' in res):
        respon="unparked"
    pesan = list(respon)
    while len(pesan)<32:
        pesan.append(0)
    radio.write(pesan)
    print ("respon terkirim{}".format(pesan))
elif("get"in string):

```

```

        print'send feed'
        radio.write(pesan)
        print ("respon ke kirim{ }".format(respon))

    radio.startListening()

except:
    GPIO.cleanup()
    time.sleep(1)

```

Program Pada Server:

HTTP *Request* Handling

```

<?php
include("koneksi.php");

date_default_timezone_set("Asia/Bangkok");
$perjam = 200;
//$charge = 500;
DATABASE);
$nuid=$_POST["nuid"];
$stat=$_POST["stat"];
$lokasi = $_POST["lokasi"];

    $chek = "SELECT No_Kendaraan,Jenis_Kendaraan,Stat FROM
data_member WHERE ID ='$nuid'";
    $chekek = mysqli_query ($db,$chek); // $link->query($chek);
    if ($chekek) {
        if($row = mysqli_fetch_array($chekek)){
            $noKendaraan = $row["No_Kendaraan"];
            $jenis = $row["Jenis_Kendaraan"];
            $statNow = $row["Stat"];}
        if($stat!=$statNow){
            if($stat=="parking"){
                $t = date('Y-m-d H:i:s');
                $parking = "INSERT INTO log_parkir
(mulai_parkir,lokasi,jenis,no_kendaraan)
('$t','$lokasi','$jenis','$noKendaraan)";
                if(mysqli_query($db,$parking)){
                    echo"berhasil masukin ke LOG";
                    $sqlStat = "UPDATE data_member SET stat='$stat' WHERE ID
='$nuid'";

```

```

        if(mysql_query($db,$sqlStat)){
            echo"park yes kendaraan dengan noPol ='$noKendaraan'
            ##parked## ";
        }
        } else{echo"gagal_masukin_ke LOG";}
    }
elseif($stat=="unpark"){
    $t = date('Y-m-d H:i:s');
    $sambilmula= mysql_query($db,"select mulai_parkir from log_parkir
    where no_kendaraan ='$noKendaraan'");
    if(!$sambilmula){echo"no_kendaraan_tidak_ada_di LOG";}
    else{
        $mula= mysql_fetch_array($sambilmula);
        $mulai = strtotime($mula['mulai_parkir']);
        $now = time();

        $diff_secs = abs($now - $mulai);
        $minutes_total = floor($diff_secs / 60);
        $durasi = date("H:i:s",$diff_secs-25200);
        if ($jenis == "mobil"){
            $perjam =500;
            //$charge =1000;
        }
        $biaya =round($minutes_total * ($perjam/60), -2);
        //$history          =          "INSERT          INTO
        history_parkir(mulai_parkir,lokasi,jenis,no_kendaraan,selesai_parkir,durasi,biaya)
        values('$sambilmula','$lokasi','$jenis','$noKendaraan',NOW(),'$durasi','$biaya');
        if(mysql_query($db,"INSERT          INTO
        history_parkir(mulai_parkir,lokasi,jenis,no_kendaraan,selesai_parkir,durasi,biaya)
        values('$mula[0]','$lokasi','$jenis','$noKendaraan','$t','$durasi','$biaya')
        )){
            echo"sukses_masukin_ke HISTORY";
            $saldo=mysql_query($db,"select Saldo from data_member where
            No_Kendaraan = '$noKendaraan'");
            $rsaldo =mysql_fetch_array($saldo);
            $sisasaldo = ($rsaldo['Saldo']-$biaya);
            mysql_query($db,"update data_member set Saldo=$sisasaldo
            where No_Kendaraan = '$noKendaraan' ");
            $dellog = mysql_query($db,"delete from log_parkir where
            no_kendaraan = '$noKendaraan'");
            if($dellog){

```

```

        $sqlStat = "UPDATE data_member SET stat='$stat' WHERE ID
=$nuid";
        if(mysql_query($db,$sqlStat)){
            echo" unpark 2 kendaraan dengan noPol ='$noKendaraan'
##stop##parkir.";
            }echo"sukses_hapus LOG $noKendaraan";
            }else{echo"gagal_hapus LOG $noKendaraan";}
            }else{echo"gagal_masukin_ke HISTORY";}

        }

    }
    }else{echo"sudah $stat sebelumnya";}
    }else{die("nuid_tidak_ditemukan");}

    //$countrow = mysql_num_rows($query);

mysql_close($db);

    //header("Location:hparkir.php");
    ?>
    <?php
    //define variable
    //$UID = $NoKendaraan= $JenisKendaraan= $NamaPemilik= $Telepone
= $Saldo= "";
    define('DB_SERVER', 'localhost');
    define('DB_USERNAME', 'root');
    define('DB_PASSWORD', "");
    define('DB_DATABASE', 'parkir');
    $db =
mysql_connect(DB_SERVER,DB_USERNAME,DB_PASSWORD,DB_DAT
ABASE);
    if (!$db){
        echo('connect error'.mysql_error());
    }
    else {echo('connected');}
    //$conn = mysql_connect("localhost","root", "", "parkir");

```

```

$UID = $_POST['uid'];
$NoKendaraan = $_POST['NoKendaraan'];
$JenisKendaraan = $_POST['JenisKendaraan'];
>NamaPemilik = $_POST['NamaPemilik'];
$Telepone = $_POST['Telepone'];
$Saldo = $_POST['Saldo'];

$sq = "INSERT INTO data_member (ID, No_Kendaraan,
Jenis_Kendaraan, Nama_Pemilik, Telp, Saldo) VALUES('$UID',
'$NoKendaraan',
'$JenisKendaraan', '$NamaPemilik', '$Telepone', $Saldo)";
$hasil = mysqli_query($db, $sq);
if ($hasil) {
    echo "New record created successfully";
    echo "<br>";
    echo "<h2>Your Input:</h2>";
    echo "<br>";
    echo $uid;
    echo "<br>";
    echo $NoKendaraan;
    echo "<br>";
    echo $JenisKendaraan;
    echo "<br>";
    echo $NamaPemilik;
    echo "<br>";
    echo $Telepone;
    echo "<br>";
    echo $Saldo;
    echo "<br>";
} else {
    echo('connect error'.mysqli_error($db));
}

mysqli_close($db);
header('location:test_connect.php?pesan=tambah_member_sukses');
?>

```

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Penulis bernama Khoiruli Miftachul Huda. Penulis dilahirkan di Sidoarjo pada tanggal 28 Maret 1994. Penulis adalah anak kedua dari 2 bersaudara. Penulis menempuh pendidikan dasar di SDN Gebang 1 Sidoarjo. Dilanjutkan dengan pendidikan menengah pertama di SMPN 3 Sidoarjo. Dan penulis menyelesaikan pendidikan menengah akhir di SMA Negeri 2 Sidoarjo. Pada tahun 2012 penulis melanjutkan pendidikannya di Institut Teknologi Sepuluh Nopember di Jurusan Teknik Elektro dengan bidang Elektronika. Judul tugas akhir penulis adalah “*Rancang Bangun Sistem Pembayaran untuk Parkir Prabayar Di Tepi jalan menggunakan RFID*”.

Email: rulihuda@gmail.com

