

**SKRIPSI**

**ALAT PENGHITUNG PENUMPANG BUS OTOMATIS**

**Disusun dan diajukan oleh:**

**ALVANYA YOSTHA PARAMITA  
D041 17 1320**



**PROGRAM STUDI SARJANA TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
GOWA  
2023**

**LEMBAR PENGESAHAN SKRIPSI**

**ALAT PENGHITUNG PENUMPANG BUS OTOMATIS**

Disusun dan diajukan oleh

**Alvanya Yostha Paramita**

**D041171320**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka  
Penyelesaian Studi Program Sarjana Program Studi Teknik Elektro  
Fakultas Teknik Universitas Hasanuddin  
Pada Tanggal 16 Februari 2023  
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,

Pembimbing Pendamping,

Dr. Ir. Rhiza S. Sadjad, MSEE.  
NIP 19570906 198203 1 004

Dr. A. Ejah Umraeni Salam, S.T., M.T.  
NIP 19720908 199702 2 001

Ketua Program Studi,

Dr. Eng. Ir. Dewiani, M.T.  
NIP 19691026 199412 2 001

## **PERNYATAAN KEASLIAN**

Yang bertanda tangan dibawah ini :

Nama : Alvanya Yostha Paramita

NIM : D041171320

Program Studi : Teknik Elektro

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

### **ALAT PENGHITUNG PENUMPANG BUS OTOMATIS**

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 27 Februari 2023

Yang Menyatakan

Alvanya Yostha Paramita

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat dan kasih karunia-Nya sehingga penulis dapat menyelesaikan skripsi berjudul: “Alat Penghitung Penumpang Bus Otomatis”.

Penulis menyadari bahwa dalam menyelesaikan tugas akhir ini banyak kendala yang dihadapi. Namun, berkat penyertaan dan pertolongan Tuhan Yesus Kristus serta bantuan dan dukungan dari berbagai pihak, akhirnya tugas akhir ini dapat terselesaikan.

Pada kesempatan ini, penulis menyampaikan ucapan terima kasih bagi semua pihak yang telah memberikan bantuan, dukungan dan doa dalam penyusunan tugas akhir ini hingga selesai. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis, Bapak Agus Suherman dan Ibu Margaretha Pamasi, Saudara(i) Donny Suherman dan Adelline Putri Meredith beserta seluruh keluarga yang senantiasa memberikan doa dan dukungan dalam bentuk apapun sehingga penulis dapat menyelesaikan tugas akhir.
2. Bapak Dr. Ir. Rhiza S. Sadjad, MSEE selaku Pembimbing Utama dan Ibu Dr. A. Ejah Umraeni Salam, S.T., M.T. selaku Pembimbing Pendamping yang telah meluangkan waktu dan memberikan bimbingan terhadap penelitian yang dilakukan.
3. Bapak Prof. Dr. -Ing. Faizal Arya Samman, S.T., M.T. dan Ibu Ida Rachmaniar Sahali, S.T., M.T. selaku dosen penguji yang memberikan saran, koreksi dan arahan dalam menyelesaikan tugas akhir penulis.
4. Ibu Dr. Eng. Ir. Dewiani, M.T. dan Bapak Dr. Ikhlas Kitta, S.T., M.T. selaku Ketua dan Sekretaris Departemen Teknik Elektro Fakultas Teknik Universitas Hasanuddin.
5. Seluruh dosen pengajar serta pegawai Departemen Teknik Elektro Universitas Hasanuddin atas bimbingan, didikan, kemudahan dan bantuan yang telah diberikan kepada penulis selama menempuh perkuliahan.
6. Teman-teman seperjuangan di Laboratorium Sistem Kendali dan Instrumentasi. Suciati, Min Idznullah Said, Arson Marianus, Ilham Ramli dan Muhammad

Alif Fikri yang selalu memberikan bantuan, doa dan dorongan dalam menyelesaikan tugas akhir ini.

7. Saudari Aryni Ponto, Abigail Anggie, Ovianti Mallisa, Alya Previa, Stefani Suryaningsih, Nurul Fadillah dan Mawar Melati yang senantiasa berbagi kisah dan memberikan dukungan dan dorongan kepada penulis dalam menempuh perkuliahan, serta El dan Opi yang selalu memberikan dorongan, dukungan dan motivasi dalam menyelesaikan tugas akhir.
8. Seluruh teman-teman EQUAL17ER, REDISCOVER YOU dan TEKNIK 2017 yang telah menjadi rekan seperjuangan, memberikan banyak pengalaman bersama, selalu berbagi cerita, waktu dan motivasi untuk menempuh perkuliahan.
9. Semua pihak yang telah membantu dalam penyelesaian tugas akhir ini namun tidak dapat disebutkan satu persatu pada kesempatan ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan tugas akhir ini. Oleh karena itu, penulis mengharapkan kritik dan saran dari berbagai pihak guna dijadikan bahan evaluasi untuk meningkatkan kualitas kedepannya. Semoga tugas akhir ini bermanfaat dan dapat menjadi bahan informasi bagi kita semua.

Gowa, Februari 2023

Alvanya Yostha Paramita

## ABSTRAK

**ALVANYA YOSTHA PARAMITA.** *Alat Penghitung Penumpang Bus Otomatis*  
(dibimbing oleh Rhiza S. Sadjad dan A. Ejah Umraeni Salam)

Untuk menuju ke daerah terpencil ataupun daerah yang sulit dijangkau tentu dibutuhkan transportasi tertentu untuk mencapainya, salah satu transportasi umum yang bisa digunakan yaitu bus antar daerah. Dalam melakukan pengecekan jumlah penumpang bus, kernet atau pembantu supir masih melakukan perhitungan manual. Hal ini dapat menghambat ketepatan waktu tiba bus apabila pada saat pengecekan jumlah penumpang tidak sesuai dengan jumlah *real* di dalam bus atau terjadi kesalahan dalam perhitungan sehingga ada penumpang yang tertinggal saat melakukan pemberhentian. Penelitian ini bertujuan untuk merancang sebuah alat yang dapat menghitung jumlah penumpang pada bus secara otomatis yang akan ditampilkan di LCD dan dapat di-*monitoring* oleh kernet atau pembantu supir melalui layar monitor yang disediakan di dalam bus. Untuk membantu proses *monitoring*, pada penelitian ini juga dirancang *website* yang berfungsi untuk mengatur *database* penumpang dan menampilkan keadaan bus secara *realtime*. Alat ini menggunakan NodeMCU ESP8266 sebagai mikrokontroler yang akan mengontrol *reader* dan sensor *infrared* dalam membaca *Smart Card* yang akan digunakan penumpang. Hasil pengujian dari penelitian ini adalah alat yang dirancang berhasil membaca informasi penumpang masuk atau keluar bus dari *Smart Card* yang terbaca oleh *reader* dan jumlah penumpang di dalam bus ditampilkan pada LCD. Selain itu, keadaan bus dapat di-*monitoring* oleh kernet secara *realtime* melalui monitor.

Kata Kunci: NodeMCU ESP8266, *Smart Card*, Sensor *Infrared*, Penghitung Penumpang

## ABSTRACT

**ALVANYA YOSTHA PARAMITA.** *Automatic Bus Passenger Counter*  
(supervised by Rhiza S. Sadjad and A. Ejah Umraeni Salam)

To get to remote areas or hard-to-reach areas, certain transportation is needed. One of the public transportation that can be used is interregional buses. In checking the number of bus passengers, kernet or assistant drivers are still doing manual calculations. This can hinder the timely arrival of the bus if at the time of checking the number of passengers does not match the real number on the bus or if there is an error in the calculation so that some passengers are left behind when making stops. This research aims to design a device that can calculate the number of passengers on the bus automatically which will be displayed on the LCD and can be monitored by the driver's assistant through the monitor screen provided on the bus. To assist the monitoring process, this research also designs a website to manage the passenger database and to display the state of the bus in real time. This device uses NodeMCU ESP8266 as a microcontroller which will control the reader and infrared sensor in reading the smart card that will be used by passengers. The results from this research are that the designed device succeeds in reading information on passengers entering or leaving the bus from the smart card which is read by the reader and the number of passengers on the bus is displayed on the LCD. In addition, the state of the bus can be monitored by the kernet in real time via the monitor.

Keywords: NodeMCU ESP8266, Smart Card, Infrared Sensor, Passenger Counter

## DAFTAR ISI

|                                                        |     |
|--------------------------------------------------------|-----|
| LEMBAR PENGESAHAN SKRIPSI.....                         | i   |
| PERNYATAAN KEASLIAN.....                               | ii  |
| KATA PENGANTAR .....                                   | iii |
| ABSTRAK .....                                          | v   |
| ABSTRACT.....                                          | vi  |
| DAFTAR ISI.....                                        | vii |
| DAFTAR GAMBAR .....                                    | ix  |
| DAFTAR TABEL .....                                     | x   |
| DAFTAR LAMPIRAN .....                                  | xi  |
| BAB I PENDAHULUAN.....                                 | 1   |
| 1.1 Latar Belakang.....                                | 1   |
| 1.2 Rumusan Masalah.....                               | 2   |
| 1.3 Tujuan Penelitian.....                             | 2   |
| 1.4 Manfaat Penelitian.....                            | 2   |
| 1.5 Batasan Masalah .....                              | 3   |
| BAB II TINJAUAN PUSTAKA.....                           | 4   |
| 2.1 NodeMCU ESP8266.....                               | 4   |
| 2.2 <i>Radio Frequency Identification</i> (RFID) ..... | 6   |
| 2.3 <i>Smart Card</i> .....                            | 14  |
| 2.4 LCD 16x2 .....                                     | 21  |
| 2.5 Modul I2C.....                                     | 26  |
| 2.6 Sensor IR FC-51 .....                              | 27  |
| 2.7 <i>Light Emitting Diode</i> (LED).....             | 31  |
| 2.8 PhpMyAdmin .....                                   | 32  |
| 2.8 Arduino IDE .....                                  | 33  |
| 2.9 XAMPP .....                                        | 34  |
| 2.10 Penelitian Terkait.....                           | 39  |
| BAB III METODE PERANCANGAN.....                        | 41  |
| 3.1 Perencanaan dan Perancangan.....                   | 41  |
| 3.2 Alat dan Bahan Penelitian .....                    | 43  |
| 3.3 Perancangan Perangkat Keras.....                   | 44  |

|                                              |                                     |    |
|----------------------------------------------|-------------------------------------|----|
| 3.4                                          | Perancangan Perangkat Lunak.....    | 46 |
| 3.5                                          | Perancangan Implementasi Alat ..... | 48 |
| 3.6                                          | Perancangan Pengujian.....          | 49 |
| BAB IV HASIL PENELITIAN DAN PEMBAHASAN ..... |                                     | 51 |
| 4.1                                          | Implementasi Alat.....              | 51 |
| 4.2                                          | Hasil Uji Coba .....                | 54 |
| BAB V KESIMPULAN DAN SARAN.....              |                                     | 62 |
| 5.1                                          | Kesimpulan.....                     | 62 |
| 5.2                                          | Saran .....                         | 63 |
| LAMPIRAN .....                               |                                     | 67 |

## DAFTAR GAMBAR

|                                                                                                                 |    |
|-----------------------------------------------------------------------------------------------------------------|----|
| Gambar 1 NodeMCU ESP8266 .....                                                                                  | 5  |
| Gambar 2 Konfigurasi <i>pin</i> NodeMCU .....                                                                   | 6  |
| Gambar 3 Cara kerja RFID <i>tag</i> pasif .....                                                                 | 9  |
| Gambar 4 Cara kerja RFID <i>tag</i> aktif.....                                                                  | 10 |
| Gambar 5 Cara kerja RFID <i>reader</i> sebagai <i>receiver</i> dan transfer data.....                           | 11 |
| Gambar 6 Konfigurasi <i>pin</i> modul MFRC522 RFID .....                                                        | 14 |
| Gambar 7 Ukuran standar <i>Smart Card</i> format ID-1 .....                                                     | 15 |
| Gambar 8 Arsitektur <i>Smart Card</i> dengan <i>microprocessor</i> .....                                        | 16 |
| Gambar 9 <i>Smart Card's pin contact</i> .....                                                                  | 17 |
| Gambar 10 LCD 16x2.....                                                                                         | 22 |
| Gambar 11 Modul I2C .....                                                                                       | 26 |
| Gambar 12 Sensor Inframerah FC-51 .....                                                                         | 30 |
| Gambar 13 Light Emitting Diode (LED) .....                                                                      | 31 |
| Gambar 14 Diagram Alir Penelitian .....                                                                         | 41 |
| Gambar 15 Blok Diagram Rancangan Sistem .....                                                                   | 42 |
| Gambar 16 Skematik Rancangan Perangkat Keras Alat Penghitung Penumpang                                          | 44 |
| Gambar 17 Proses Perakitan Alat Penghitung Penumpang Bus Otomatis .....                                         | 45 |
| Gambar 18 Diagram Alir Sistem Alat Penghitung Penumpang Bus Otomatis....                                        | 46 |
| Gambar 19 Software Arduino IDE .....                                                                            | 47 |
| Gambar 20 Tampilan Notepad++ .....                                                                              | 47 |
| Gambar 21 <i>Database Website</i> Sistem.....                                                                   | 48 |
| Gambar 22 Rancangan Instalasi Alat di Bus.....                                                                  | 48 |
| Gambar 23 Rancangan Perangkat Keras Sistem Penghitung.....                                                      | 49 |
| Gambar 24 Perangkat Keras Alat Penghitung Penumpang Bus Otomatis.....                                           | 51 |
| Gambar 25 Halaman Login <i>Website</i> .....                                                                    | 52 |
| Gambar 26 Homepage <i>Website</i> .....                                                                         | 52 |
| Gambar 27 Tampilan Keadaan Bus di Monitor .....                                                                 | 53 |
| Gambar 28 Grafik Perbandingan Lama Waktu Pembacaan <i>Reader</i> dan Pengiriman<br>Data ke <i>Website</i> ..... | 55 |
| Gambar 29 Tampilan Keadaan Bus Sebelum Pengujian .....                                                          | 57 |
| Gambar 30 Tampilan LCD Saat Percobaan 1 .....                                                                   | 58 |
| Gambar 31 Tampilan LCD Saat Percobaan 2 .....                                                                   | 58 |
| Gambar 32 Tampilan Monitor Saat Percobaan 3.....                                                                | 59 |
| Gambar 33 Tampilan LCD Saat diberi Halangan Pada Sensor Masuk .....                                             | 60 |
| Gambar 34 Tampilan LCD Saat diberi Halangan Pada Sensor Keluar .....                                            | 60 |

## DAFTAR TABEL

|                                                                                                                                                                   |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tabel 1 Jenis frekuensi RFID.....                                                                                                                                 | 12 |
| Tabel 2 Standarisasi ISO/IEC <i>contactless Smart Card</i> .....                                                                                                  | 20 |
| Tabel 3 Deskripsi <i>pin</i> pada LCD.....                                                                                                                        | 26 |
| Tabel 4 <i>Pin</i> Modul I2C .....                                                                                                                                | 27 |
| Tabel 5 Tabel spesifikasi sensor <i>infrared</i> .....                                                                                                            | 29 |
| Tabel 6 Jenis-jenis LED dan Tegangan .....                                                                                                                        | 32 |
| Tabel 7 Penelitian Terkait .....                                                                                                                                  | 39 |
| Tabel 8 Perangkat Keras yang digunakan.....                                                                                                                       | 43 |
| Tabel 9 Perangkat Lunak yang digunakan.....                                                                                                                       | 43 |
| Tabel 10 Hasil Pengujian Lama Waktu Pembacaan <i>Reader</i> Masuk dan <i>Reader</i><br>Keluar terhadap Smart Card dan Waktu Pengiriman Data ke <i>Website</i> ... | 54 |
| Tabel 11 Hasil Pengujian Jarak Baca <i>Reader</i> terhadap Smart Card .....                                                                                       | 56 |
| Tabel 12 Hasil Pengujian Smart Card yang Tidak Terdaftar di <i>Database</i> .....                                                                                 | 57 |

## DAFTAR LAMPIRAN

|             |                                                       |    |
|-------------|-------------------------------------------------------|----|
| Lampiran 1  | Source Code pada Arduino IDE .....                    | 68 |
| Lampiran 2  | Program Halaman Login.....                            | 72 |
| Lampiran 3  | Program Halaman Registrasi .....                      | 74 |
| Lampiran 4  | Program Halaman Index .....                           | 75 |
| Lampiran 5  | Program Halaman Kondisi Bus Terkini.....              | 76 |
| Lampiran 6  | Program Halaman Data Bus .....                        | 78 |
| Lampiran 7  | Program Halaman Data Kursi Bus.....                   | 79 |
| Lampiran 8  | Program Halaman Rekap Data Admin .....                | 80 |
| Lampiran 9  | Program Halaman Rekap Data Bus Terkini .....          | 81 |
| Lampiran 10 | Program Halaman Coming Soon .....                     | 83 |
| Lampiran 11 | Program Baca Kartu.....                               | 84 |
| Lampiran 12 | Program Reader Masuk .....                            | 85 |
| Lampiran 13 | Program Reader Keluar .....                           | 86 |
| Lampiran 14 | Program Update Kondisi Bus Terkini .....              | 87 |
| Lampiran 15 | Program Hapus Data .....                              | 87 |
| Lampiran 16 | Program Edit Data .....                               | 88 |
| Lampiran 17 | Program Baca Kartu Untuk Edit .....                   | 90 |
| Lampiran 18 | Program Tambah Data .....                             | 90 |
| Lampiran 19 | Program Konfirmasi Pembayaran.....                    | 92 |
| Lampiran 20 | Program Koneksi Website .....                         | 93 |
| Lampiran 21 | Program Header Halaman.....                           | 93 |
| Lampiran 22 | Program Menu Halaman .....                            | 94 |
| Lampiran 23 | Program Footer Halaman.....                           | 94 |
| Lampiran 24 | Program Logout .....                                  | 94 |
| Lampiran 25 | Pengeluaran Biaya 1 (Satu) Unit Alat Penghitung ..... | 95 |

## **BAB I PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam masa perkembangannya, Indonesia senantiasa berusaha memajukan segala aspek guna untuk menjadi negara yang lebih maju. Segala aspek tersebut melingkupi pendidikan, transportasi, ekonomi, sosial, politik dan masih banyak lainnya. Peningkatan transportasi sendiri bertujuan untuk memudahkan masyarakat dalam melakukan suatu *perpindahan* dari suatu wilayah ke wilayah lainnya. Baik menggunakan transportasi darat, laut ataupun udara.

Salah satu penunjuk dan pemicu keberhasilan pembangunan adalah keberhasilan bidang transportasi yang berarti kemudahan pergerakan (mobilitas) (Septiani, 2019). Untuk mendukung transportasi yang lebih maju, Indonesia berusaha meningkatkan penggunaan transportasi dalam negeri dan transportasi ke luar negeri namun tidak dapat dipungkiri masih banyak kekurangan yang memerlukan perbaikan lanjutan. Kebutuhan akan sarana transportasi dari waktu ke waktu terus mengalami peningkatan akibat semakin banyaknya kegiatan-kegiatan yang membutuhkan jasa transportasi sehingga bertambah pula intensitas pergerakan lalu lintas antar kota (Hariyadi, 2017). Untuk menuju ke daerah-daerah terpencil ataupun daerah-daerah yang sulit dijangkau tentu dibutuhkan transportasi tertentu untuk mencapainya, salah satu transportasi umum yang bisa digunakan yaitu bus antar daerah.

Dalam menjaga kenyamanan dan keamanan di bus, kernet atau pembantu supir akan melakukan pengecekan jumlah penumpang sebelum bus akan berangkat atau sebelum bus akan melanjutkan perjalanan kembali. Didorong dari kebiasaan manual maka dalam perhitungan tersebut juga digunakan manusia sebagai penghitung manual (Syahputra, 2016). Kadang kala ini menghambat ketepatan waktu tiba bus apabila pada saat pengecekan jumlah penumpang belum mencapai jumlah yang sesungguhnya dalam artian masih ada penumpang yang belum berada di dalam bus. Terkadang pula karena kelalaian kernet dalam melakukan pengecekan, ada penumpang yang tertinggal saat bus sedang melakukan pemberhentian dalam

sebuah perjalanan. Selain itu, kendala yang juga kadang terjadi yaitu penumpang yang salah menaiki bus antar daerah sesuai kota tujuan mereka yang tentu saja ini bisa menghambat pergerakan bus.

Oleh karena itu, pada penelitian ini penulis akan merancang sebuah alat yang dapat menghitung jumlah penumpang pada bus secara otomatis yang dapat dimonitoring oleh kernet atau pembantu supir melalui layar monitor yang disediakan di dalam bus. Adapun judul dari Tugas Akhir ini adalah **“ALAT PENGHITUNG PENUMPANG BUS OTOMATIS”**.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut, maka dapat dirumuskan masalah sebagai berikut.

1. Bagaimana merancang dan merealisasikan alat penghitung penumpang bus otomatis?
2. Bagaimana kinerja sistem kendali alat penghitung penumpang bus?
3. Bagaimana kinerja sistem monitoring alat penghitung penumpang bus?

## **1.3 Tujuan Penelitian**

Berdasarkan rumusan masalah yang ada, maka tujuan dari penelitian ini adalah sebagai berikut.

1. Untuk membuat alat penghitung penumpang bus secara otomatis.
2. Untuk mengetahui kinerja sistem kendali alat penghitung penumpang bus.
3. Untuk mengetahui kinerja sistem monitoring alat penghitung penumpang bus.

## **1.4 Manfaat Penelitian**

Manfaat dari penelitian ini adalah :

1. Bagi Pengguna

Hasil penelitian diharapkan dapat memudahkan pekerjaan kernet dalam mengecek jumlah penumpang dan menghindari masalah kenyamanan penumpang.

## 2. Bagi Peneliti

Dapat mengetahui kinerja alat penghitung penumpang bus otomatis dan menambah pengetahuan untuk pengaplikasian ilmu yang didapatkan selama masa kuliah.

## 3. Bagi Akademisi

Hasil penelitian ini diharapkan dapat menjadi referensi untuk penelitian yang serupa atau dapat dikembangkan lagi.

### 1.5 Batasan Masalah

Batasan-batasan masalah yang diberikan sesuai dengan permasalahan yang telah dijelaskan adalah sebagai berikut.

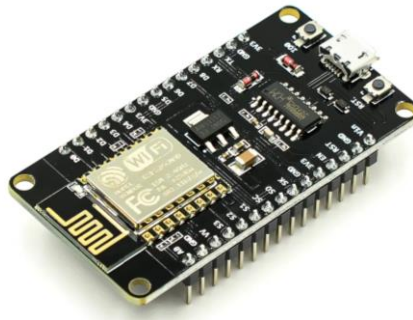
1. Pada sistem ini hanya membahas tentang kemampuan suatu alat untuk melakukan perintah yang telah dibuat dan dirancang, seperti memberi perintah *reader* untuk membaca *Smart Card* yang akan ditampilkan di LCD dan layar monitor serta mengirimkan data *realtime* ke *database* untuk ditampilkan pada *website*.
2. Sistem penghitung yang diusulkan diuji coba dengan hanya menggunakan satu pintu sebagai akses masuk dan akses keluar.
3. Alat penghitung diujicobakan di dalam ruangan dengan menggunakan skenario seperti pada saat di bus.
4. Proses pengujian hanya menggunakan 10 *Smart Card* sebagai identitas kursi penumpang.

## BAB II TINJAUAN PUSTAKA

### 2.1 NodeMCU ESP8266

Modul WiFi NodeMCU adalah *firmware* interaktif berbasis LUA Espressif ESP8266 WiFi SoC. Selain dapat diprogram menggunakan bahasa LUA dapat juga diprogram menggunakan bahasa C menggunakan Arduino IDE (Wicaksono & Hidayat, Mudah Belajar Mikrokontroler Arduino, 2017). NodeMCU dapat beroperasi pada frekuensi *clock* yang dapat disesuaikan 80 hingga 160 MHz. Modul ini memiliki 128 KB RAM dan 4MB Memori *flash* (untuk program dan penyimpanan data) yang cukup untuk mengatasi *string* besar yang membentuk halaman web, data JSON/XML, dan semua yang digunakan di perangkat IoT saat ini. ESP8266 mengintegrasikan 802.11b/g/n HT40 WiFi *Transceiver*, sehingga tidak hanya dapat terhubung ke jaringan WiFi dan berinteraksi dengan internet, tetapi juga dapat mengatur jaringan sendiri, memungkinkan perangkat lain untuk terhubung langsung ke modul. Hal ini membuat NodeMCU ESP8266 lebih fleksibel (Last Minute Engineers, n.d.).

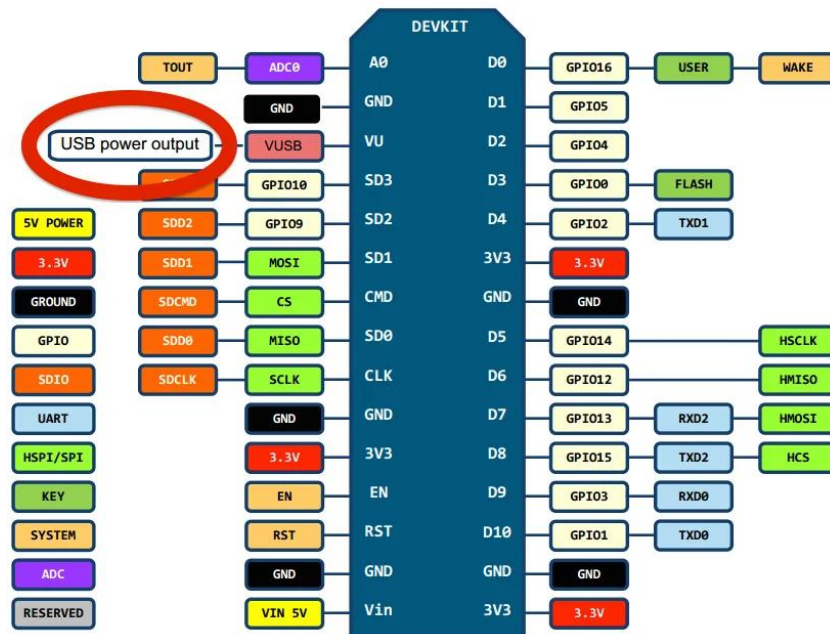
Pada NodeMCU dilengkapi dengan *micro usb port* yang berfungsi untuk pemrograman maupun *power supply*. Selain itu juga pada NodeMCU dilengkapi dengan *push button* yaitu tombol *reset* dan *flash*. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan paket dari ESP8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan C hanya berbeda *syntax*. Selain dengan bahasa Lua NodeMCU juga *support* dengan *software* Arduino IDE dengan melakukan sedikit perubahan *board manager* pada Arduino IDE. Sebelum digunakan *Board* ini harus di *Flash* terlebih dahulu agar *support* terhadap *tool* yang akan digunakan. Jika menggunakan Arduino IDE menggunakan *firmware* yang cocok yaitu *firmware* keluaran dari Ai- Thinker yang *support* AT Command. Untuk penggunaan *tool loader firmware* yang digunakan adalah *firmware* NodeMCU (Adibrata, 2020).



Gambar 1 NodeMCU ESP8266

NodeMCU berukuran panjang 4,83 cm, lebar 2,54 cm dan berat 7 gram. *Board* ini sudah dilengkapi dengan fitur WiFi dan *firmware*nya yang bersifat *opensource*. Spesifikasi yang dimiliki oleh NodeMCU sebagai berikut (Adibrata, 2020):

1. *Board* ini berbasis ESP8266 serial WiFi SoC (*Single on Chip*) dengan USB to TTL. *Wireless* yang digunakan adalah IEEE 802.11b/g/n.
2. 2 *tantalum capacitor* 100 *micro farad* dan 10 *micro farad*.
3. 3.3v LDO regulator.
4. *Blue led* sebagai indikator.
5. Cp2102 USB to UART *bridge*.
6. Tombol reset, *port* USB dan tombol *flash*.
7. Terdapat 9 GPIO yang di dalamnya ada 3 *pin* PWM, 1 x ADC *Channel* dan *pin* RX/TX.
8. 3 *pin ground*.
9. S3 dan S2 sebagai *pin* GPIO.
10. S1 MOSI (*Master Output Slave Input*) yaitu jalur data dari *master* dan masuk ke dalam *slave*, sc cmd/sc.
11. S0 MISO (*Master Input Slave Input*) yaitu jalur data keluar dari *slave* dan masuk ke dalam *master*.
12. SK yang merupakan SCLK dari *master* ke *slave* yang berfungsi sebagai *clock*.
13. *Pin* Vin sebagai masukan tegangan.
14. *Built in* 32-bit MCU.



Gambar 2 Konfigurasi *pin* NodeMCU

## 2.2 Radio Frequency Identification (RFID)

RFID adalah proses identifikasi seseorang atau objek dengan menggunakan frekuensi transmisi radio. RFID menggunakan frekuensi radio untuk membaca informasi dari sebuah *device* kecil yang disebut *tag* atau *transponder* (*transmitter* + *responder*). *Tag* RFID akan mengenali diri sendiri ketika mendeteksi sinyal dari *device* yang kompatibel, yaitu pembaca RFID (*Micro-Reader*) (Irawan dkk, 2016).

RFID merupakan teknologi identifikasi yang mudah digunakan, fleksibel dan sangat cocok untuk pengoperasian otomatis, serta mengkombinasikan keunggulan yang tidak tersedia pada teknologi identifikasi yang lain (Utama, 2010). RFID dapat disediakan dalam bentuk *tag* yang hanya dapat dibaca saja (*Read Only*) atau dapat dibaca dan ditulis (*Read/Write*), tidak memerlukan kontak langsung maupun jalur cahaya untuk dapat beroperasi, dapat berfungsi pada berbagai variasi kondisi lingkungan, dan menyediakan tingkat integritas data yang tinggi. Sebagai tambahan, karena teknologi ini sulit untuk dipalsukan, maka RFID dapat menyediakan tingkat keamanan yang tinggi (Irawan dkk, 2016).

### 2.2.1 Komponen utama RFID

Sistem RFID terdiri dari 4 komponen, yaitu RFID *tag* (*transponder*), antena, *reader*, dan *interface software* (Kurnia, 2017).

- a. RFID *tag* (*transponder*) memiliki *chip* yang dapat menyimpan data berupa nomor ID unik dan memiliki antena yang berfungsi untuk mentransmisikan data ke RFID *reader* melalui gelombang radio yang dipancarkan RFID *reader*.
- b. Antena terdapat pada RFID *tag* (*tag*-antena) dan RFID *reader* (*reader* antena) yang berfungsi mentransmisikan data dari *chip* RFID *tag* ke RFID *reader* melalui gelombang radio.
- c. RFID *reader* adalah perangkat yang kompatibel dengan RFID *tag*. RFID *reader* akan memancarkan gelombang radio dan menginduksi RFID *tag* yang kemudian akan mengirim data ID dari antena yang terdapat pada rangkaian RFID *tag* melalui gelombang radio yang dipancarkan RFID *reader*.
- d. *Interface Software* yang berfungsi untuk membaca data ID dan RFID *reader* dan mengolah data tersebut sehingga dapat digunakan menjadi *password*.

### 2.2.2 Jenis-jenis RFID *tag*

RFID *tag* adalah *device* yang dibuat dari rangkaian elektronika dan antena yang terintegrasi di dalam rangkaian tersebut (Utama, 2010). RFID *tag* memiliki *chip* yang dapat menyimpan data ID, *transponder* atau *tag*-antena yang berfungsi untuk mengirim data melalui gelombang radio yang dipancarkan RFID *reader* dan *encapsulation* yang berfungsi untuk melindungi *chip* agar tidak mudah rusak (Kurnia, 2017). RFID *tag* dibagi menjadi 2 berdasarkan catu dayanya, yaitu:

- a. *Tag* Pasif (*Passive Tags*)

Kartu RFID pasif tidak mempunyai *transmitter* maupun sumber daya. Kartu jenis ini biasanya lebih murah dari kartu RFID aktif dan juga tidak membutuhkan perawatan. *Transponder* RFID terdiri dari *microchip* yang menempel pada antena. Karena ukurannya yang kecil, *transponder* bisa saja dibungkus dalam berbagai macam bentuk, seperti di dalam lipatan kertas berlabel *barcode* atau di dalam kartu

plastik. Bentuk pembungkus yang digunakan tergantung pada jenis karakteristik aplikasi yang menggunakan RFID ini (Kurnia, 2017).

Kartu RFID pasif dapat menggunakan *low frequency* (124 kHz, 125 kHz atau 135 kHz), *high frequency* (13,56 MHz), atau UHF (860 MHz-960 MHz). Jenis frekuensi yang digunakan juga sangat bergantung pada karakteristik aplikasi karena tiap rentang frekuensi mempunyai karakteristik tertentu. Pada rentang frekuensi tertentu gelombang radio tidak dapat menembus benda logam atau air. Rentang frekuensi juga mempunyai karakteristik jarak maksimum pancaran gelombang radio yang berbeda-beda (Kurnia, 2017).

Perusahaan pengguna RFID umumnya banyak menggunakan RFID pasif berfrekuensi UHF dibandingkan dengan *low frequency* atau *high frequency*. Hal ini karena kartu RFID pasif yang menggunakan UHF berharga lebih murah dan jangkauannya lebih luas (jangkauannya sampai dengan 3,33 meter). Banyak aplikasi biasanya membutuhkan kartu RFID yang dapat dibaca pada jarak minimal 3 meter dari piranti pembaca. Aplikasi jenis ini misalnya aplikasi pengelolaan barang di gudang yang memerlukan kartu dapat dibaca ketika masuk *pintu* dan jangkauan kartu tentu saja minimal 3 meter. Sedangkan kartu RFID yang menggunakan *low frequency* hanya dapat dibaca pada jarak maksimal 0,3 meter dari piranti pembaca, sedang untuk *high frequency* dapat dibaca pada jarak 1 meter (Kurnia, 2017).

Metode pengiriman data kartu RFID pasif ke piranti pembaca dapat dibagi menjadi 2 macam, yaitu (Kurnia, 2017):

1. *Inductive Coupling*

Gulungan tembaga pada piranti pembaca membangkitkan medan elektromagnetik, kemudian gulungan yang ada di kartu RFID terinduksi oleh medan ini. Hasil induksi inilah yang menjadi sumber tenaga bagi kartu RFID untuk mengirimkan kembali sinyal yang berisi data ke piranti pembaca. Karena menggunakan prinsip induksi ini, maka jarak antara kartu RFID dengan piranti pembaca juga harus pendek agar induksi dapat ditangkap. *Inductive coupling* digunakan pada kartu RFID dengan *low frequency* dan *high frequency*.

## 2. *Propagation Coupling*

Pada sistem ini, energi yang digunakan berasal dari energi elektromagnetik (gelombang radio) yang dipancarkan oleh piranti pembaca. Kartu RFID kemudian akan mengumpulkan energi elektromagnetik ini untuk digunakan sebagai sumber daya mengirimkan data yang dimilikinya ke piranti pembaca.



Gambar 3 Cara kerja RFID tag pasif

### b. *Tag Aktif (Active Tags)*

Kartu RFID aktif mempunyai sumber daya sendiri dan mempunyai *transmitter*. Sumber daya yang digunakan bisa berasal dari baterai atau tenaga surya. Karena mempunyai sumber daya sendiri, RFID jenis ini mempunyai jangkauan yang lebih luas, yaitu antara 20 meter sampai 100 meter. Kartu ini akan melakukan *broadcast* sinyal untuk mengirimkan data dengan menggunakan *transmitter* yang dimilikinya. RFID jenis ini biasanya beroperasi pada frekuensi 455 MHz, 2,45 GHz atau 5,8 GHz. Kartu jenis ini digunakan pada aset bernilai besar (kargo, kontainer atau mobil) karena kartu jenis ini berharga relatif mahal (Kurnia, 2017).

Kartu RFID aktif dapat dibagi lagi menjadi 2 jenis, yaitu *transponder* dan *beacon*. *Transponder* hanya akan melakukan *broadcast* ketika mereka menerima sinyal dari piranti pembaca. Contoh umum dari sistem ini adalah pada sistem pembayaran di gerbang jalan tol. Pada saat mobil memasuki *pintu* keluar, maka piranti pembaca pada gerbang akan mengirim sinyal yang akan membangunkan *transponder* di kaca depan. *Transponder* kemudian akan melakukan *broadcast* data yang berisi identitas mobil tersebut. *Beacon* banyak digunakan pada *Real-Time Locating System* (RTLS), yaitu sistem untuk mengetahui lokasi suatu objek dengan cepat. Pada *beacon*, sinyal dikirimkan secara periodik pada selang interval tertentu. Frekuensi pengiriman sinyal bergantung pada tingkat kepentingan untuk mengetahui letak aset. Sinyal yang dipancarkan oleh *beacon* ditangkap dengan menggunakan minimal 3 buah piranti pembaca. Kartu RFID aktif ini juga dapat

ditambah dengan alat pembaca temperatur udara atau kelembaban udara, ada tidaknya sensor ini juga akan memengaruhi harga dari kartu (Kurnia, 2017).

Setiap bagian *Tag* RFID terdiri dari (Kurnia, 2017):

1. Silikon Mikroprosesor

Silikon Mikroprosesor adalah sebuah *chip* yang terletak dalam sebuah *tag* yang berfungsi sebagai penyimpan data.

2. *Metal Coil*

*Metal Coil* adalah sebuah komponen yang terbuat dari kawat aluminium yang berfungsi sebagai antena yang dapat beroperasi pada frekuensi 13,56 MHz. Jika sebuah *tag* masuk ke dalam jangkauan *reader* maka antena ini akan mengirimkan data yang ada pada *tag* kepada *reader* terdekat.

3. *Encapsulating Material*

*Encapsulating Material* adalah bahan yang membungkus *tag* yang terbuat dari bahan kaca. *Tag* RFID telah sering dipertimbangkan untuk digunakan sebagai *barcode* pada masa yang akan datang. Pembacaan informasi pada *tag* RFID tidak memerlukan kontak sama sekali. Karena kemampuan rangkain terintegrasi yang modern, maka *tag* RFID dapat menyimpan jauh lebih banyak informasi dibandingkan dengan *barcode*.



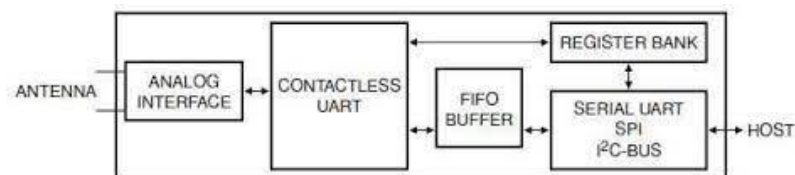
Gambar 4 Cara kerja RFID tag aktif

### 2.2.3 RFID reader

Agar sistem RFID berfungsi dengan baik diperlukan RFID *reader* yang dapat membaca RFID *tag* dan mengirim data yang dibaca ke *database*. Sebuah *reader* menggunakan antena untuk berkomunikasi dengan RFID *tag*. Ketika *reader* memancarkan gelombang radio seluruh RFID *tag* yang memiliki frekuensi sama dengan *reader* akan memberikan respon (Kurnia, 2017).

RFID *reader* memancarkan gelombang radio dan menginduksi RFID *tag*. Gelombang induksi tersebut berisi data ID dan jika dikenali oleh RFID *tag*, memori RFID *tag* (ID *chip*) aktif terbuka. Gelombang radio yang dipancarkan oleh *reader* juga berfungsi sebagai catu daya RFID *tag* (*tag* pasif). Kemudian RFID *tag* akan mengirimkan kode yang terdapat di memori ID *chip* melalui antena yang terpasang di RFID *tag*. RFID *reader* akan mengirim data tersebut ke mikrokontroler untuk diproses menjadi *password* sebagai pengaman (Kurnia, 2017).

RFID *reader* memiliki antena yang berfungsi untuk memancarkan gelombang radio ke RFID *tag* dan menerima data yang dikirim oleh RFID *tag*, data tersebut berupa sinyal analog yang kemudian akan diteruskan ke *contactless* UART yang berfungsi untuk membaca data ID dari RFID *tag* kemudian data ID tersebut akan dikirim ke register *bank* dan FIFO *buffer*. Register *bank* mengirim data ID ke serial UART kemudian akan mengirim data ID tersebut kepada HOST (mikrokontroler). FIFO *buffer* berfungsi mengirim data dari *contactless* UART kepada HOST (mikrokontroler) dan dari mikrokontroler ke *contactless* UART, data yang dikirim berupa data serial (Kurnia, 2017).



Gambar 5 Cara kerja RFID *reader* sebagai *receiver* dan transfer data

## 2.2.4 Frekuensi kerja RFID

Faktor penting yang harus diperhatikan dalam RFID adalah frekuensi kerja dari sistem RFID. Ini adalah frekuensi yang digunakan untuk komunikasi *wireless* antara pembaca RFID dengan *tag* RFID. Ada beberapa *band* frekuensi yang digunakan untuk sistem RFID. Pemilihan dari frekuensi kerja sistem RFID akan memengaruhi jarak komunikasi, interferensi dengan frekuensi sistem radio lain, kecepatan komunikasi data dan ukuran antena. Untuk frekuensi yang rendah umumnya digunakan *tag* pasif dan untuk frekuensi tinggi digunakan *tag* aktif. Pada frekuensi rendah, *tag* pasif tidak dapat mentransmisikan data dengan jarak yang jauh karena keterbatasan daya yang diperoleh dari medan elektromagnetik. Akan

tetapi komunikasi tetap dapat dilakukan tanpa kotak langsung. Pada kasus ini hal yang perlu mendapatkan perhatian adalah *tag* pasif harus terletak jauh dari objek logam karena logam secara signifikan mengurangi fluks dari medan magnet. Akibatnya *tag* RFID tidak bekerja dengan baik karena *tag* tidak menerima daya minimum untuk dapat bekerja. Pada frekuensi tinggi, jarak komunikasi antara *tag* aktif dengan pembaca RFID dapat lebih jauh, tetapi masih terbatas oleh daya yang ada. Sinyal elektromagnetik pada frekuensi tinggi juga mendapatkan pelemahan ketika *tag* tertutupi oleh es atau air. Pada kondisi terburuk, *tag* yang tertutup oleh logam tidak terdeteksi oleh pembaca RFID. Ukuran antena yang harus digunakan untuk transmisi data bergantung dari panjang gelombang elektromagnetik. Untuk frekuensi yang rendah, maka antena harus dibuat dengan ukuran yang lebih besar dibandingkan dengan RFID dengan frekuensi tinggi (Kurnia, 2017).

Tabel 1 Jenis frekuensi RFID

| No | Frekuensi RFID (Hz) | Jenis Frekuensi             | Manfaat                                            |
|----|---------------------|-----------------------------|----------------------------------------------------|
| 1  | 125 kHz - 134 kHz   | <i>Low Frequency</i>        | Menandai hewan ( <i>Animal tagging</i> )           |
| 2  | 13,56 MHz           | <i>High Frequency</i>       | <i>Smart Card</i>                                  |
| 3  | 860 MHz - 930 MHz   | <i>Ultra High Frequency</i> | Membuka otomatis bagasi, identifikasi suatu barang |
| 4  | 2,4 GHz             | <i>Micro-Wave</i>           | Akses kontrol bagasi pesawat terbang               |

Sumber: Kurnia (2017)

### 2.2.5 Tingkat akurasi RFID

Akurasi RFID dapat didefinisikan sebagai tingkat keberhasilan pembaca RFID melakukan identifikasi sebuah *tag* yang berada pada area kerjanya. Keberhasilan dari proses identifikasi sangat dipengaruhi oleh beberapa batasan fisik, yaitu posisi antena pada pembaca RFID, karakteristik dari material lingkungan yang mencakup sistem RFID, batasan catu daya, dan frekuensi kerja sistem RFID (Kurnia, 2017).

a. Akurasi Sistem RFID Frekuensi Rendah

Pada frekuensi rendah, contohnya pada frekuensi 13,56 MHz, komunikasi frekuensi radio antara *tag* dengan pembaca RFID sangat bergantung pada daya yang diterima *tag* dari antena yang terhubung dengan pembaca RFID. Pada ruang bebas, intensitas dari medan magnet yang diemisikan oleh antena berkurang terhadap jarak, maka terdapat batas jarak dimana *tag* tidak aktif dan komunikasi frekuensi radio tidak dapat terjadi. Pengurangan ukuran *tag* akan mengurangi juga batas jarak. Komunikasi radio berkurang jika medan magnet harus menembus material yang mengurangi daya elektromagnetik, contohnya pada kasus objek dengan bahan logam. *Tag* RFID tidak akan terdeteksi ketika ditaruh di dalam logam, karena material logam akan meredam fluks magnet yang melau *tag* secara drastis. Orientasi dari *tag* sangat penting dan dapat menyebabkan medan magnet bervariasi. Jika orientasi *tag* RFID sejajar dengan arah propagasi energi, maka fluks adalah nol dan komunikasi radio frekuensi tidak akan terjadi walaupun jarak antena dan *tag* sangat dekat (Kurnia, 2017).

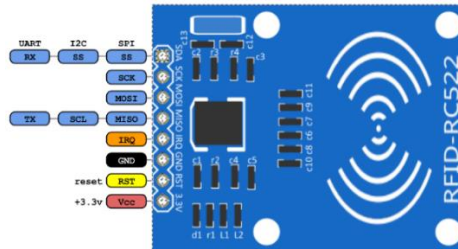
b. Akurasi Sistem RFID Frekuensi Tinggi

Pada frekuensi tinggi, performa dari sistem RFID sangat bergantung pada lingkungan dimana komunikasi di antara *tag* dan pembaca RFID terjadi. Pada jarak tanpa hambatan proses identifikasi dapat terjadi pada jarak pada orde 10 meter. Tetapi bila ada hambatan maka jarak ini akan berkurang secara drastis. Pada frekuensi tinggi, *tag* RFID bekerja secara aktif dengan daya dari baterai. Akurasi dari *tag* RFID dapat berkurang karena kekurangan daya. Akurasi dari sistem RFID pada umumnya sangat bergantung dari lingkungan dimana sistem RFID dioperasikan. Tantangan desain sistem RFID adalah melakukan desain infrastruktur RFID di antara lingkungan yang kurang bersahabat yang telah dijelaskan sebelumnya (Kurnia, 2017).

## 2.2.6 RFID MIFARE RC522

MIFARE RC522 adalah sebuah modul berbasis IC *Philips* MFRC522 yang dapat membaca RFID dengan penggunaan yang mudah dan harga yang murah, karena modul ini sudah berisi komponen-komponen yang diperlukan oleh

MFRC522 untuk dapat bekerja. Modul ini dapat digunakan langsung MCU dengan menggunakan *interface* SPI, dengan suplai tegangan sebesar 3,3V. MFRC522 merupakan produk dari NXP yang menggunakan *fully integrated* 13,56 MHz *non-contact communication card chip* untuk melakukan pembacaan atau penulisan (Kurnia, 2017).



Gambar 6 Konfigurasi *pin* modul MFRC522 RFID

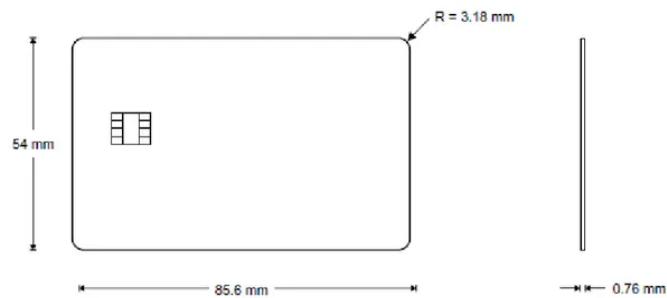
### 2.3 *Smart Card*

*Smart Card* adalah media yang dapat menyimpan data dalam satu kartu yang merupakan pengembangan dari kartu magnetik dan memiliki ukuran serupa dengan kartu pembayaran plastik masa kini (Wicaksono, 2011).

Standarisasi dari *Smart Card* ditentukan oleh Standar ISO/IEC yang merupakan standar internasional. ISO/IEC 7816 dan ISO/IEC 7810 adalah sebuah standar untuk *Smart Card* untuk menentukan hal berikut (Gintoro dan William, 2010):

1. Bentuk fisik dari kartu.
2. Karakteristik dari sirkuit elektronik.
3. Posisi dan ukuran kartu dari konektor listrik kartu.
4. Protokol komunikasi.
5. Ketahanan kartu.
6. Fungsionalitas kartu.

Berdasarkan ISO/IEC ukuran *Smart Card* dengan format ID-1 adalah 85,60mm x 54mm dan memiliki ketebalan kartu  $0.76\text{mm} \pm 0.08\text{mm}$  serta jari-jari sudut 3,18mm (Gintoro dan William, 2010).



Gambar 7 Ukuran standar *Smart Card* format ID-1

### 2.3.1 Tipe-tipe *Smart Card*

*Smart Card* terdiri dari dua tipe yaitu *microprocessor card* dan *memory card*. *Microprocessor card* adalah sebuah kartu yang memiliki memori dan mikroprosesor yang dapat melakukan kalkulasi data dan menyimpan data di dalam kartu secara aman. *Microprocessor Card* menawarkan sekuritas dan kemampuan multifungsional. Data yang disimpan di dalam *microprocessor card* tidak dapat diakses secara langsung melalui aplikasi di luar kartu. Selain itu, *microprocessor card* mengontrol data dan memori yang mengatur pengaksesan data menurut kondisi yang diberikan seperti *password*, enkripsi dan lainnya dan juga keuntungan dari *microprocessor card* dapat diintegrasikan ke lebih dari satu aplikasi (Gintoro dan William, 2010).

Sedangkan *Memory card* adalah kartu yang hanya menyimpan dan memproteksi data secara lokal, namun tidak mengandung sebuah prosesor untuk melakukan perhitungan komputer pada data atau memanipulasi data. Sebuah *memory card* dapat menyimpan data sebesar 4K dan keuntungan dari *memory card* terletak pada teknologinya yang sederhana (Gintoro dan William, 2010). Kartu ini bergantung pada alat pengakses kartu yang sesuai untuk dapat beroperasi.

### 2.3.2 Sistem memori *Smart Card*

*Smart Card* terdiri dari 3 macam jenis memori, yaitu (Gintoro dan William, 2010):

#### 1. ROM

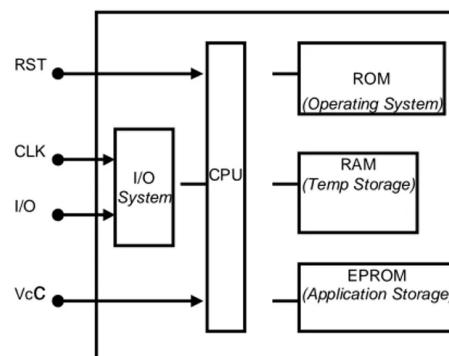
ROM (*Read-Only Memory*) merupakan tipe memori yang hanya dapat dibaca (*read*) dan tidak bisa ditulis (*write*), serta tidak membutuhkan tenaga listrik untuk menyimpan dan menjaga keutuhan data. ROM tidak dapat ditulis kembali setelah kartu dibuat dan berisikan sebuah sistem operasi. ROM dari *Smart Card* berisikan data dan aplikasi pengguna yang bersifat permanen.

#### 2. EEPROM

EEPROM (*Electrical Erasable Programmable Read Only Memory*) hampir serupa dengan ROM yang dapat menyebabkan data ketika tenaga dari memori dimatikan. Perbedaan di antara keduanya adalah isi dari memori yang dapat dimodifikasi selama penggunaan kartu secara normal. EEPROM dapat menerima setidaknya 100.000 kali penulisan, dan dapat menyimpan data selama 10 tahun. Membaca dari EEPROM sama cepatnya dengan membaca dari RAM, namun menulis pada EEPROM lebih lama dibandingkan menulis dari RAM.

#### 3. RAM

RAM (*Random Access Memory*) adalah sebuah memori yang digunakan untuk menampung data yang disimpan selama sebuah proses berlangsung. Jumlah akses pada suatu RAM tidak terbatas dan membutuhkan tenaga listrik untuk beroperasi. RAM merupakan memori yang bersifat *nonpersistent* yaitu informasi yang disimpan pada RAM akan hilang ketika tenaga listrik dari RAM hilang.



Gambar 8 Arsitektur *Smart Card* dengan *microprocessor*

### 2.3.3 Jenis Smart Card

Secara fisik *Smart Card* dapat dibedakan menjadi dua jenis yaitu:

#### 1. *Contact Smart Card*

*Contact Smart Card* memiliki *chip* kecil keemasan pada kartu yang saat dibaca oleh *reader*, *chip* tersebut melakukan kontak dengan konektor yang dapat membaca informasi dari *chip* dan dapat menuliskan informasi kembali ke dalam *chip* (Wicaksono, 2011). *Contact Smart Card* bekerja dengan cara berkomunikasi secara fisik antara *card reader* dan *Smart Card pin contact* yang berbentuk segi empat berukuran  $\pm 1$  cm. *Contact Smart Card* tidak membutuhkan baterai sebagai sumber tenaga dan akan aktif ketika terhubung dengan *card reader*. Saat terhubung dengan *card reader*, maka *chip* akan menunggu perintah *request* dari *client* dari aplikasi untuk membaca informasi dari *chip* atau menulis informasi ke *chip* (Gintoro dan William, 2010). *Contact Smart Card* memerlukan kontak fisik dengan mekanisme membaca daripada menggunakan *contactless interface*.



Gambar 9 *Smart Card's pin contact*

Berikut adalah fungsi dari masing-masing *contact* menurut ISO/IEC 7816, yaitu (Gintoro dan William, 2010):

#### a. VCC (*Supply Voltage*)

Berfungsi sebagai penyedia tegangan listrik. Tegangan listrik yang tersedia sebesar 3 atau 5 volt dengan toleransi maksimum  $\pm 10\%$ .

#### b. RST (*Reset Input*)

Berfungsi untuk mengirimkan sinyal untuk mereset mikroprosesor.

#### c. CLK (*Clock Input*)

Berfungsi sebagai *timing* atau *clocking signal* yang mengatur frekuensi waktu dan kecepatan mikroprosesor.

d. GND (*Ground*)

Merupakan *reference voltage* yang berlawanan dengan VCC di mana energi potensialnya diukur dan nilainya dianggap 0 volt.

e. VPP (*Programming Voltage*)

Berfungsi menyediakan tegangan listrik yang berbeda dari VCC. Biasa digunakan untuk aplikasi lain seperti konektivitas USB.

f. I/O (*Input/Output*)

Berfungsi untuk memonitor semua komunikasi yang keluar maupun yang masuk ke kartu.

g. C4 dan C8

*Contact* C4 dan C8 merupakan *contact* tambahan yang digunakan bila sewaktu-waktu diperlukan untuk operasi pada kartu atau penggunaan *interface* di masa yang akan datang.

2. *Contactless Smart Card*

*Contactless Smart Card* berkomunikasi dengan terminal melalui sinyal frekuensi radio dan tidak memiliki baterai sehingga kartu ini memiliki induktor yang dapat menangkap sinyal frekuensi radio sebagai daya elektronik bagi kartu (Gintoro dan William, 2010). Jenis *Smart Card* ini memiliki kemampuan untuk mengelola, menyimpan, dan menyediakan akses ke data pada kartu dengan aman, serta menjalankan fungsi pada kartu (misalnya enkripsi) dan berinteraksi secara cerdas dengan *reader*. *Contactless Smart Card* memerlukan jarak tertentu untuk melakukan pertukaran data dengan *card reader* (Secure Technology Alliance).

Berdasarkan pada standar ISO/IEC 14443, *Contactless Smart Card* memiliki frekuensi 13,56 MHz. *Reader* yang sesuai standar akan memiliki bidang aktivasi (rentang) sekitar 4 inci (sekitar 10 sentimeter). Dengan kata lain, kartu harus berada dalam jarak 10 sentimeter dari *reader* agar dapat diberi daya secara efektif. Namun, jangkauan komunikasi yang efektif untuk kartu yang akan dibaca akan tergantung pada sejumlah faktor seperti kekuatan *reader*, antena *reader* dan antena kartu. Jarak pendek ini memastikan bahwa pengguna melakukan tindakan sadar saat mendekatkan kartu dan *reader*. Hal ini sangat berguna dalam keperluan untuk

bertransaksi sehingga menghindari transaksi yang tidak disengaja atau penipuan. *Contactless Smart Card* harus diposisikan di area target yang sangat dekat dengan *reader* agar dapat berfungsi, sehingga mengurangi kemungkinan terbaca “*read*” tanpa sepengetahuan pengguna. Selain itu, informasi yang disimpan pada kartu biasanya dilindungi dari pencurian dengan enkripsi yang aman dan komunikasi antara kartu dan *reader* aman dan diautentikasi (Secure Technology Alliance).

*Contactless Smart Card* berisi antena yang tertanam di dalam badan kartu plastik. Ketika kartu berada dalam jangkauan medan elektromagnetik *reader* dan sesuai dengan frekuensi yang diperlukan, maka *chip* dalam kartu dihidupkan. Setelah *chip* dihidupkan, protokol komunikasi nirkabel dimulai dan dibuat antara kartu dan *reader* untuk transfer data atau dapat dikatakan kartu siap untuk berkomunikasi dengan *reader* (Secure Technology Alliance).

*Contactless Smart Card* memiliki memori dari 512 bit hingga 72 Kbytes. Selain itu, jenis kartu ini mampu menyimpan 72.000 byte atau karakter dengan aman dan menyediakan kemampuan baca (*read*) dan tulis (*write*) yang aman. Kemampuan membaca data dari kartu *pintar* tidak memungkinkan untuk menulis (*write*) atau memperbarui data, kecuali jika penyedia aplikasi merencanakannya demikian. Kedua elemen utama ini menempatkan *Contactless Smart Card* dibagian atas daftar opsi saat mengimplementasikan proyek yang memerlukan pengelolaan data dalam jumlah besar yang aman – jauh di depan alternatif lainnya (Secure Technology Alliance).

Teknologi *Contactless Smart Card* digunakan dalam aplikasi yang perlu melindungi informasi pribadi dan/atau mengirimkan transaksi yang cepat dan aman, seperti kartu pembayaran ongkos transit, kartu identitas pemerintah dan perusahaan, dokumen seperti paspor dan visa elektronik dan kartu pembayaran keuangan. *Contactless Smart Card* juga dapat mendukung berbagai algoritma enkripsi untuk meningkatkan keamanan. Hal ini penting untuk aplikasi yang bekerja di bidang perbankan, transportasi dan ID aman karena memberikan tingkat keamanan setinggi mungkin. *Contactless Smart Card* lebih aman dan lebih andal dalam menyimpan dan membawa informasi, serta memiliki masa pakai yang diharapkan lebih lama daripada sebagian besar opsi lain yang tersedia. Misalnya, karena

keamanan yang tinggi, keandalan, dan kenyamanan transaksi cepat, semua aplikasi kartu *pintar* di transportasi umum diimplementasikan menggunakan teknologi *Contactless Smart Card*. Selain itu, jenis *Smart Card* ini digunakan untuk mengelola sejumlah besar informasi variabel dengan aman yang biasanya menjadi perhatian besar otoritas penerbit kartu (Secure Technology Alliance).

Aplikasi yang menggunakan *Contactless Smart Card* dapat melindungi data yang disimpan dalam beberapa cara. Pertama, untuk mengakses data dari *Contactless Smart Card* aplikasi mungkin memerlukan pengetahuan tentang kunci rahasia tertentu. Secara umum, tanpa mengetahui kunci rahasia ini, mikrokontoler dan sirkuit kartu akan memblokir setiap upaya untuk mengakses data pada *chip*. Kedua, informasi yang disimpan pada kartu atau dokumen yang menggunakan teknologi *Contactless Smart Card* dapat dienkripsi. Selain itu, komunikasi antara *Contactless Smart Card* dan *reader* dapat dienkripsi untuk mencegah penyadapan. Aplikasi yang aman juga biasanya memerlukan “otentikasi timbal balik” yang mana *Contactless Smart Card* terlebih dahulu memverifikasi bahwa *reader* itu asli dan kemudian membuktikan keasliannya sendiri kepada *reader* sebelum memulai komunikasi lebih lanjut. Kemampuan aplikasi *Contactless Smart Card* untuk memverifikasi otoritas pemohon informasi dan memberikan keamanan *chip* dan data yang kuat menjadikannya sebagai penjaga informasi pribadi dan privasi individu yang sangat baik (Secure Technology Alliance).

Standar komunikasi *contactless Smart Card* adalah ISO/IEC 14443. Berikut adalah tabel standarisasi ISO/IEC 14443 (Gintoro dan William, 2010).

Tabel 2 Standarisasi ISO/IEC *contactless Smart Card*

| Standar       | Tipe <i>Contactless Smart Card</i>    | Jarak Komunikasi (cm) |
|---------------|---------------------------------------|-----------------------|
| ISO/IEC 10536 | <i>Close-Coupling card</i>            | $\pm 1$ cm            |
| ISO/IEC 14443 | <i>Proximity coupling card</i> (PICC) | $\pm 10$ cm           |
| ISO/IEC 15693 | <i>Vicinity coupling card</i> (VICC)  | $\pm 1$ m             |

Sumber: Gintoro dan William (2010)

### 2.3.4 Dampak positif dan negatif *Smart Card*

Adapun dampak positif dan negatif dari penggunaan *Smart Card*, yaitu (Yahya, 2019):

- a. Dampak Positif *Smart Card*
  1. Mempermudah berkomunikasi dengan cepat.
  2. Kemudahan bertransaksi dan berbisnis dalam berbagai bidang.
  3. Mempermudah mendapatkan informasi dimanapun dan kapanpun.
  4. Mempermudah pekerjaan secara cepat dan efisien.
  5. Mempermudah akses data dengan cepat dan mudah.
- b. Dampak Negatif *Smart Card*
  1. Selalu mencari jalan yang mudah atau instan.
  2. Membuat orang atau pengguna menjadi malas.
  3. Menimbulkan radiasi yang sangat besar.

### 2.3.5 *Smart Card reader*

*Smart Card reader* adalah sebuah perangkat elektronik yang diperlukan oleh *host computer* agar dapat membaca atau berkomunikasi dengan *Smart Card*. *Smart Card* bekerja bersama *reader* untuk dapat menyelenggarakan transaksi informasi dari dan ke kartu dengan aplikasi di dunia luar, misalnya kartu identitas kependudukan, passport, kartu asuransi, kartu catatan medis, visa, debit, *e-ticketing*, dan akses keamanan. Umumnya *Smart Card reader* yang banyak digunakan adalah yang bertipe *contacted* atau dengan hubungan kontak langsung karena lebih aman. Selain itu, konektivitasnya stabil karena kartu berada pada posisi tetap di dalam *card slot* dan dapat melakukan transaksi data berukuran besar (Yahya, 2019).

## 2.4 LCD 16x2

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama yang pengoperasiannya menggunakan sistem dot matriks. LCD *dot matrix* berfungsi untuk

menampilkan tulisan berupa angka, karakter, huruf, dan grafik sesuai dengan yang diinginkan (sesuai dengan program yang digunakan untuk mengontrolnya) (Arsul, 2020). LCD membutuhkan tegangan dan daya yang kecil sehingga banyak digunakan sebagai *display* dari alat-alat elektronika atau digunakan untuk aplikasi seperti kalkulator, multimeter digital, jam digital, dan sebagainya (Kurnia, 2017).

LCD memanfaatkan silikon dan galium dalam bentuk kristal cair sebagai pemancar cahaya. Pada layar LCD, setiap matrik adalah susunan dua dimensi piksel yang dibagi dalam baris dan kolom. Dengan demikian, setiap pertemuan baris dan kolom terdiri dari LED pada bidang datar (*backplane*), yang merupakan lempengan kaca bagian belakang dengan sisi dalam yang ditutupi oleh lapisan elektrode transparan. Dalam keadaan normal, cairan yang digunakan memiliki warna cerah. Kemudian daerah-daerah tertentu pada cairan tersebut warnanya akan berubah menjadi hitam ketika tegangan diterapkan antara bidang latar dan pola elektrode yang terdapat pada sisi dalam kaca bagian depan (Tanjung, 2015).

Keunggulan menggunakan LCD adalah konsumsi daya yang relatif kecil dan menarik arus yang kecil (beberapa mikroampere), sehingga alat atau sistem menjadi *portable* karena dapat menggunakan catu daya yang kecil. Keunggulan lainnya adalah ukuran LCD yang pas yakni tidak terlalu kecil dan tidak terlalu besar, kemudian tampilan yang diperlihatkan dari LCD dapat dibaca dengan mudah dan jelas (Tanjung, 2015).



Gambar 10 LCD 16x2

Spesifikasi pada LCD 16x2 adalah sebagai berikut (Tanjung, 2015) (Novaldy dkk, 2019).

1. Terdiri dari 16 kolom dan 2 baris.
2. Mempunyai 192 karakter yang tersimpan.
3. Terdapat karakter generator terprogram.

4. Dapat dialamati dengan mode 4-bit dan 8-bit.
5. Dilengkapi dengan *back light*.
6. Tegangan kerja 5V.
7. Memiliki ukuran yang praktis.

Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan register, memori yang digunakan adalah (Kurnia, 2017):

- a. DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang ditampilkan.
- b. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat berubah-ubah sesuai dengan keinginan.
- c. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD.

Register kontrol yang terdapat dalam suatu LCD diantaranya adalah (Kurnia, 2017):

- a. Register perintah yaitu register yang berisi perintah-perintah dari mikrokontroler ke panel LCD pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data.
- b. Register data yaitu register menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada register akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur sebelumnya.

*Pin*, kaki atau jalur input kontrol dalam suatu LCD diantaranya adalah (Kurnia, 2017):

- a. *Pin* data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD dapat dihubungkan dengan *bus* data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.

- b. *Pin RS (Register Select)* berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, baik data atau perintah. Logika *low* menunjukkan yang masuk dalam perintah sedangkan logika *high* menunjukkan data.
- c. *Pin R/W (Read Write)* berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.
- d. *Pin E (Enable)* digunakan untuk memegang data baik masuk atau keluar.
- e. *Pin VLCD* berfungsi mengatur kecerahan tampilan (kontras) dimana *pin* ini dihubungkan dengan *trimpot* 5 Kohm, jika tidak digunakan dihubungkan ke *ground* sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

#### 2.4.1 Prinsip kerja LCD 16x2

Prinsip kerja LCD 16x2 adalah dengan menggunakan lapisan film yang berisi kristal cair dan diletakkan di antara dua lempeng kaca yang telah dipasang elektrode logam transparan. Saat tegangan dicatukan pada beberapa pasang elektrode, molekul-molekul kristal cair akan menyusun agar cahaya yang mengenainya akan diserap. Dari hasil penyerapan cahaya tersebut akan terbentuk huruf, angka atau gambar sesuai bagian yang diaktifkan. Untuk membentuk karakter atau gambar pada kolom dan baris secara bersamaan digunakan metode *screening*. Metode *screening* adalah mengaktifkan daerah perpotongan suatu kolom dan baris secara bergantian dan cepat sehingga seolah-olah aktif semua (Tanjung, 2015).

Pada aplikasi umumnya RW diberi logika rendah “0”. *Bus* data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. *Interface* LCD merupakan sebuah *parallel bus*, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD seacara 4-bit atau 8-bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 *nibble* data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa *clock* EN setiap *nibble*-nya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroler mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus men-*set* EN ke kondisi *high* “1” dan

kemudian men-set dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data *bus*. Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat (tergantung pada *datasheet* LCD) dan set EN kembali ke *high* “1”. Ketika jalur RS berada dalam kondisi *low* “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan *layer*, posisi kursor dll). Ketika RS dalam kondisi *high* atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur kontrol R/W harus berada dalam kondisi *low* (0) saat informasi pada data *bus* akan dituliskan ke LCD. Apabila R/W berada dalam kondisi *high* “1”, maka program akan melakukan *query* (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu *Get* LCD status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diset ke “0”. Jalur data dapat terdiri atas 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara *parallel* baik 4-bit atau 8-bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi *interface* LCD, menentukan mode operasi merupakan hal yang paling penting (Adibrata, 2020).

Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 *pin* I/O (3 *pin* untuk kontrol, 8 *pin* untuk data). Sedangkan mode 4-bit minimal hanya membutuhkan 7-bit (3 *pin* untuk kontrol, 4 bit untuk data). Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroler dan LCD. Jika bit ini di set (RS=1), maka byte p (Fadil & Thamrin, 2020) ada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di reset (RS=0), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca (Adibrata, 2020).

## 2.4.2 Deskripsi *pin* LCD 16x2

Berikut ini tabel deskripsi *pin* pada LCD 16x2 (Tanjung, 2015).

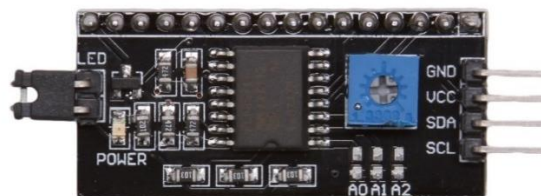
Tabel 3 Deskripsi *pin* pada LCD

| <i>Pin</i> | <b>Simbol</b> | <b>I/O</b> | <b>Deskripsi</b>                                                                            |
|------------|---------------|------------|---------------------------------------------------------------------------------------------|
| 1          | VSS           | --         | <i>Ground</i>                                                                               |
| 2          | VCC           | --         | <i>+5 V power supply</i>                                                                    |
| 3          | VEE           | --         | <i>Power supply source to control contrast</i>                                              |
| 4          | RS            | I          | Register select: RS = 0 to select intruksi.<br>Command register: RS = 1 to select data reg. |
| 5          | R/W           | I          | Read/Write: R/W=0 for write, R/W=1 for read                                                 |
| 6          | E             | I          | <i>Enable</i>                                                                               |
| 7          | DB0           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 8          | DB1           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 9          | DB2           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 10         | DB3           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 11         | DB4           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 12         | DB5           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 13         | DB6           | I/O        | <i>The 8-bit data bus</i>                                                                   |
| 14         | DB7           | I/O        | <i>The 8-bit data bus</i>                                                                   |

Sumber: Tanjung (2015)

## 2.5 Modul I2C

LCD memiliki banyak konfigurasi *pin* yang mesti dihubungkan ke *pin* NodeMCU ESP8266 sehingga menyebabkan banyaknya *pin* yang terpakai maka untuk mengurangi penggunaan *pin* mikrokontroler oleh LCD ditambahkan sebuah modul khusus yang dirancang untuk *driver* dari LCD yaitu modul I2C. Sehingga dengan penggunaan modul ini, maka *pin* yang akan digunakan menjadi 4 *pin* saja yaitu VCC, GND, SCL dan SDA (Fadil dan Thamrin, 2020).



Gambar 11 Modul I2C

*Inter Integrated Circuit* (I2C) merupakan standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data (Sarmidi dan Rahmat, 2019). Ini dikembangkan pada awal 1980-an oleh Philips dan jalur dua kabel ini telah menjadi standar untuk kontrol sistem dan digunakan untuk mengontrol segala sesuatu mulai dari sensor suhu dan penerjemah level tegangan hingga EEPROM, I/O tujuan umum, A/D dan D/A konverter, mikrokontroler dan mikroprosesor (Kamarudin, 2017).

I2C LCD adalah modul LCD yang dikendalikan secara serial sinkron dengan protokol I2C/IIC (*Inter Integrated Circuit*) atau TWI (*Two Wire Interface*). Normalnya, modul LCD dikendalikan secara *parallel* baik untuk jalur data maupun kontrolnya (Lukmanulhakim, 2020). Modul I2C ini diconverter menggunakan *Chip IC PCF8574* produk dari NXP untuk *controller*-nya pada sistem. IC ini adalah 8-bit I/O *expender for I2C bus* yang dasarnya adalah sebuah *shift register* (Saputra dan Hadijaja, 2019).

Modul I2C memiliki 4 *pin* yang menghubungkan antara modul I2C ke mikrokontroler, yaitu *pin* VCC, GND, SDA dan SCL. Pada tabel berikut menunjukan penjelasan dari *pin* yang terdapat pada modul I2C (Lukmanulhakim, 2020).

Tabel 4 *Pin* Modul I2C

| No. | Nama <i>Pin</i> | Fungsi                  |
|-----|-----------------|-------------------------|
| 1.  | GND             | <i>Ground</i>           |
| 2.  | VCC             | +5V                     |
| 3.  | SDA             | Mengirim/menerima data  |
| 4.  | SCL             | Mengirim/menerima clock |

Sumber: Lukmanulhakim (2020)

## 2.6 Sensor IR FC-51

*IR Obstacle* Sensor Inframerah merupakan sebuah modul yang berfungsi sebagai pendeteksi halangan atau objek di depannya. Sensor ini memiliki tiga kaki *pin* untuk *power supply*, *ground* dan digital input mikrokontroler (Putra dan Wagya, 2021). Pada perkembangannya, sensor jarak memiliki dua kelompok, yaitu sensor ultrasonik dan sensor inframerah.

Sensor ultrasonik untuk mengukur jarak dihasilkan dari gelombang ultrasonik yang dipancarkan atau dikeluarkan oleh *transmitter* atau alat pemancar gelombang ultrasonik. *Transmitter* mengeluarkan gelombang ultrasonik yang dihasilkan dari frekuensi di atas normal dari gelombang suara. Cara kerjanya yaitu *transmitter* akan mengeluarkan gelombang ultrasonik yang biasanya dikeluarkan secara berkala dalam beberapa detik sekali. Pancaran gelombang ultrasonik tersebut akan terus dipancarkan menyeluruh dan meluas dalam jangkauannya. Kemudian ketika pancaran gelombang ultrasonik tersebut menabrak sebuah objek tertentu, maka pancaran gelombang ultrasonik tersebut akan terhenti dan dengan kemudian berbalik arah menuju alat penerima sinyal ultrasonik atau lebih dikenal dengan istilah *receiver* yang terdapat pada sensor jarak. Pada saat itu juga *receiver* akan memberikan data dari hasil tangkapan gelombang ultrasonik tadi kepada mikrokontroler yang kemudian oleh mikrokontroler akan diproses menjadi sebuah data mengenai bentuk objek dan jarak dari objek yang tersentuh gelombang ultrasonik tadi. Jaraknya gelombang yang dipancarkan oleh *transmitter* tergantung pada alat yang digunakan (Adibrata, 2020).

Jenis selanjutnya dari sensor jarak adalah sensor inframerah. Perbedaan sensor inframerah dengan sensor ultrasonik sendiri sebenarnya sangat kecil, karena perbedaannya hanya terletak pada cara kerjanya. Apabila pada sensor ultrasonik mempergunakan gelombang ultrasonik untuk mendeteksi sebuah objek pada jarak tertentu. Maka pada sensor inframerah, untuk dapat mendeteksi sebuah objek dan mendapatkan gambaran serta jaraknya adalah dengan menggunakan panas tertentu dari sebuah benda atau objek. Setiap suhu panas dari suatu objek akan tertangkap oleh sensor inframerah karena inframerah menggunakan sumber utamanya yaitu radiasi panas atau juga radiasi termal (Adibrata, 2020). Sensor inframerah merupakan sensor yang bekerja berdasarkan sinyal inframerah. Sensor inframerah dapat digunakan untuk membaca halangan, pergerakan, dan mendeteksi panas emisi dari suatu benda. Komponen pendeteksi pada sensor ini menggunakan sinyal inframerah untuk mendeteksi adanya objek. Komponen pendeteksi ini terdiri dari IR *transmitter* yang berfungsi sebagai pemancar sinar inframerah dan IR *receiver* sebagai penerima sinar inframerah yang terpantul (Aminuddin, 2021).

Kelebihan sensor inframerah adalah mampu mengatasi tipuan-tipuan dalam bentuk cermin. Sedangkan kelemahan dari sensor ini adalah apabila objek yang dideteksi berupa dinding yang bergelombang dimana sinyal sonar akan dipantulkan ke arah lain sehingga jarak tidak terdeteksi. Berbeda dengan sensor ultrasonik, sensor inframerah tidak menghitung waktu pancaran sinar melainkan menghitung dibagian mana sinar inframerah yang dikembalikan diterima oleh rangkaian fototransistor. Semakin jauh jarak maka semakin ke kanan sinar inframerah yang diterima pada rangkaian fototransistor dan semakin kecil tegangan outputnya. Hasil output ini akan diterima terlebih dahulu sebelum diambil oleh mikrokontroler (Adibrata, 2020).

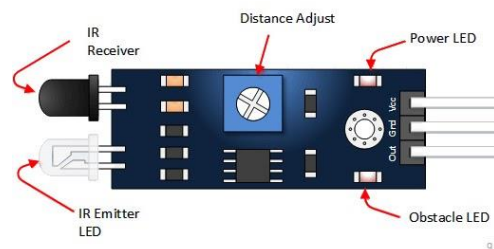
Sistem sensor inframerah pada dasarnya menggunakan inframerah sebagai media untuk komunikasi data antara *receiver* dan *transmitter*. Sistem dapat berfungsi bila sinar inframerah yang dipancarkan terhalang oleh suatu benda kerja yang mengakibatkan sinar inframerah tersebut tidak dapat terdeteksi atau terbaca oleh penerima/*receiver*. Pemancar pada sistem ini terdiri atas sebuah LED inframerah yang dilengkapi dengan rangkaian yang mampu membangkitkan data untuk dikirimkan melalui sinar inframerah, sedangkan pada bagian penerima biasanya terdapat fototransistor, fotodiode atau inframerah modul yang berfungsi untuk menerima sinar inframerah yang dikirimkan oleh pemancar (Novaldy dkk, 2019).

Tabel 5 Tabel spesifikasi sensor *infrared*

| Fitur                         | Spesifikasi                                                                                                                                                                                                    |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tegangan Masukan              | 3-5 Volt                                                                                                                                                                                                       |
| Konsumsi Arus                 | 23 mA saat 3.0V dan 43 mA saat 5.0V                                                                                                                                                                            |
| Jarak Pembacaan               | 2 – 30 cm (diatur dengan potensiometer)                                                                                                                                                                        |
| Keluaran Sensor               | Digital <i>LOW</i> ketika mendeteksi <i>obstacle</i>                                                                                                                                                           |
| Lampu LED Indikator           | Ada                                                                                                                                                                                                            |
| Sudut Pendeteksian            | 35°                                                                                                                                                                                                            |
| Pengaturan Jarak Pendeteksian | Jarak pendeteksian dapat diatur melalui potensiometer pada <i>board</i> . Putar potensiometer serah jarum jam untuk menambah jarak pendeteksian. Untuk mengurangi jarak pendeteksian putar ke arah sebaliknya. |

Sumber: Novaldy dkk (2019)

Salah satu contoh dari sensor inframerah adalah modul FC-51. Modul ini memiliki sumber generator sinyal IR dan sumber penerima pada papan yang sama. Sumber generator sinyal mengirimkan sinyal IR dan diterima oleh sumber penerima untuk mengetahui kedekatan atau hambatan yang ada di depan modul. Selain itu pada modul ini juga terdapat potensiometer yang berfungsi untuk mengatur jangkauan yang dapat dibaca sensor. Jangkauan jarak yang dapat dibaca modul FC-51 adalah 2 cm s.d. 30 cm. Terdapat 3 kaki *pin* dari modul FC-51. Kaki *pin* VCC berfungsi sebagai sumber tegangan +, kaki *pin* GND berfungsi sebagai sumber tegangan -, dan kaki OUT berfungsi sebagai masukan ke mikrokontroler (Aminuddin, 2021).



Gambar 12 Sensor Inframerah FC-51

Modul sensor inframerah FC-51 merupakan sebuah sensor yang bekerja untuk mendeteksi adanya hambatan yang berada di depan modul sensor. Modul sensor inframerah FC-51 ini memiliki dua bagian utama yang terdiri dari IR *transmitter* dan IR *receiver*. Fungsi dari IR *transmitter* adalah bagian yang bertugas untuk memancarkan radiasi inframerah kepada sebuah objek ataupun hambatan. Sedangkan IR *receiver* merupakan bagian yang berfungsi untuk mendeteksi radiasi yang telah dipantulkan oleh objek yang berasal dari IR *transmitter*. Pada bagian IR *transmitter* ini tampilannya tidak dapat terlihat oleh mata manusia. Bagian-bagian dari modul sensor inframerah ini juga terdapat beberapa bagian yang berupa potensiometer, IC LM393, LED *Obstacle* dan juga LED *power* (Adibrata, 2020).

Setelah cahaya inframerah ditembakkan pada objek dan terpantul, lalu Fototransistor akan mulai bekerja. Fototransistor akan bekerja dalam penerima cahaya inframerah. Pada fototransistor pantulan energi cahaya inframerah diubah menjadi arus listrik. Fototransistor adalah merupakan kombinasi fotodiode dan penguatan transistor. Fototransistor memiliki sensitifitas yang lebih tinggi

dibandingkan fotodiode, tetapi dengan waktu respon yang secara umum akan lebih lambat daripada fotodiode. Pada prinsip kerjanya, ketika cahaya inframerah diterima oleh fototransistor maka basis fototransistor akan mengubah energi cahaya inframerah menjadi arus listrik. Arus listrik tersebut merupakan basis *hole*. Pergerakan elektron disebut sebagai muatan listrik negatif dan pergerakan *hole* disebut sebagai muatan listrik positif (Hasibuan dkk, 2022).

Pemancar atau *transmitter* pada sensor inframerah ini terdiri dari sebuah LED inframerah yang dilengkapi dengan rangkaian yang dapat mengontrol data untuk mengirimkan melalui sinar inframerah. Dan pada bagian penerima atau *receiver* terdapat fotodiode, transistor, atau inframerah modul yang berfungsi untuk menerima sinar inframerah yang dapat dikirimkan dan dikontrol oleh pemancar atau *transmitter* (Saputra dan Hadijaja, 2019).

Manfaat serta keuntungan dari prinsip kerja di dalam penerapannya antara lain sebagai sistem keamanan dan pengendali jarak jauh, otomatisasi pada sistem kerja (Saputra dan Hadijaja, 2019). Contoh penggunaannya pada alarm yang berbunyi saat sesuatu mendekat atau mengubah arah robot ketika mendekati dinding dan bisa digunakan untuk sistem pendeteksian (Putra dan Wagyaana, 2021).

## 2.7 *Light Emitting Diode (LED)*

*Light Emitting Diode* atau disingkat LED adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju (Putra dan Wagyaana, 2021). LED merupakan keluarga diode yang terbuat dari bahan semikonduktor dan salah satu komponen elektronika yang mengubah listrik menjadi cahaya (Sarmidi dan Rahmat, 2019).



Gambar 13 Light Emitting Diode (LED)

Cara kerja LED yaitu akan memancarkan cahaya maju (*Bias Forward*) ketika dialiri tegangan dari kutub Positif/Anode (+) menuju kutub Negatif/Katode (-) (Lukmanulhakim, 2020). LED juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering dijumpai pada *remote control* perangkat elektronik (Putra dan Wagyana, 2021). Warna-warna cahaya yang dipancarkan oleh LED tergantung pada jenis bahan semikonduktor yang dipergunakannya (Natsir dkk, 2019). Pada Tabel 6 menunjukkan jenis-jenis LED dan tegangannya.

Tabel 6 Jenis-jenis LED dan Tegangan

| No. | Warna  | Tegangan Maju @20mA |
|-----|--------|---------------------|
| 1.  | Merah  | 1.8                 |
| 2.  | Jingga | 2.0                 |
| 3.  | Kuning | 2.2                 |
| 4.  | Hijau  | 3.5                 |
| 5.  | Biru   | 3.6                 |
| 6.  | Putih  | 3.4                 |

Sumber: Natsir dkk (2019)

## 2.8 PhpMyAdmin

PhpMyAdmin adalah aplikasi manajemen *database server* MySQL berbasis web. Dengan aplikasi phpMyAdmin kita bisa mengelola *database* sebagai *root* (pemilik *server*) atau juga sebagai user biasa, kita bisa membuat *database* baru, mengelola *database* dan melakukan operasi perintah-perintah *database* secara lengkap. PhpMyAdmin adalah *interface* web yang dibuat untuk mengelola *database* MySQL (Alakel dkk, 2019). PhpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*) dan lain-lain (Hartiwati, 2022).

PhpMyAdmin dibuat menggunakan bahasa PHP dan bersifat *open source* (Alakel dkk, 2019). Sesuai dengan namanya phpMyAdmin ini ditulis dalam Bahasa pemrograman PHP. *Software* ini juga berlisensi GNU (*General Public License*) dan sudah men-*support* multilingual (multi bahasa) dalam *user interface*-nya (Hartiwati, 2022). Dengan phpMyAdmin, administrator web *server* bisa mengelola *database* tanpa harus menguasai perintah berbasis baris teks (*command line*) dari SQL

(*Structure Query Language*). PhpMyAdmin sering digunakan pengembang web untuk menyiapkan *database* dari aplikasi web seperti CMS, Blog dsb (Alakel dkk, 2019).

Beberapa fitur penting dari phpMyAdmin antara lain (Alakel dkk, 2019):

1. Membuat, menghapus dan mengedit baik *database*, *table*, *record*, struktur.
2. Membuat pencarian sederhana dan kompleks.
3. *Import* CVS (bisa digunakan untuk mengimport data *spreadsheet*).
4. *Export* ke CVS, XML, Pdf, *spreadsheet*.

Perbedaan PhpMyAdmin dengan MySQL terletak pada fungsi. PhpMyAdmin merupakan alat untuk memudahkan dalam mengoperasikan *database* MySQL, sedangkan MySQL adalah *database* tempat penyimpanan data. PhpMyAdmin sendiri digunakan sebagai alat untuk mengolah/mengatur data pada MySQL (Standisyah dan Restu, 2017).

## 2.8 Arduino IDE

Untuk keperluan memprogram mikrokontroler NodeMCU ESP8266 diperlukan perangkat lunak bernama Arduino IDE. IDE (*Integrated Development Environment*) digunakan sebagai media pemrograman Arduino yang terintegrasi. Melalui software ini, Arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan 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 *Bootloader* yang berfungsi sebagai penengah antara compiler Arduino dengan mikrokontroler (Novaldy dkk, 2019).

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari

*software*. Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino (Novaldy dkk, 2019).

Pembuatan program (*coding*) menggunakan *software* Arduino. Arduino adalah *platform* dari *physical computing* yang bersifat *open source*. Arduino tidak hanya sebuah alat pengembangan tetapi kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih. IDE Arduino adalah *software* yang berfungsi untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memori mikrokontroler (Kurnia, 2017).

## 2.9 XAMPP

XAMPP adalah paket program web lengkap yang dapat dipakai untuk belajar pemrograman web, khususnya PHP dan MySQL (Alakel dkk, 2019). XAMPP adalah perangkat lunak yang bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program (Fauzi dkk, 2019). Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan X (empat sistem operasi apapun: Windows, Linux, Mac OS, Solaris), Apache, MySQL, PHP dan Perl. Program ini tersedia bebas, mudah digunakan, dan dapat melayani tampilan halaman web yang dinamis (Pradana, 2015).

Bagian penting dari XAMPP yang biasa digunakan (Alakel dkk, 2019):

1. Htdoc adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML, dan skrip lain.
2. PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada di komputer. Untuk membukanya, buka *browser* lalu ketikkan alamat <http://localhost/phpMyAdmin>, maka akan muncul halaman phpMyAdmin.
3. Kontrol panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan ataupun memulai (*start*).

Perangkat lunak komputer ini memiliki kelebihan untuk bisa berperan sebagai *server* web Apache untuk simulasi pengembangan *website*. Tool pengembangan web ini mendukung teknologi web populer seperti PHP dan MySQL. Melalui program ini, *programmer* web dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. XAMPP juga dilengkapi fitur manajemen *database* phpMyAdmin seperti pada *server hosting* sungguhan, sehingga pengembang web dapat mengembangkan web berbasis *database* dengan mudah (Pradana, 2015).

### **2.9.1. Apache web server**

Web *server* merupakan *server* internet yang mampu melayani koneksi transfer data dalam *protocol* HTTP. Web *server* telah dirancang untuk dapat melayani beragam jenis data, dari *text* sampai grafis. Kemampuan ini telah menyebabkan berbagai institusi seperti universitas maupun perusahaan dapat menerima kehadirannya dan juga sekaligus menggunakannya sebagai sarana di Internet (Pradana, 2015). Fungsi utama *Server* atau Web *Server* adalah untuk melakukan atau akan mentransfer bekas permintaan pengguna melalui *protocol* komunikasi yang telah ditentukan sedemikian rupa. Pemanfaatan web *server* berfungsi untuk mentransfer seluruh aspek pemberkasan dalam sebuah halaman web termasuk yang di dalam berupa teks, video, gambar dan banyak lagi.

Salah satu perangkat lunak web *server* yang biasa digunakan adalah Apache Web *Server*. Apache (Apache Web *Server* – The HTTP Web *Server*) merupakan web *server* yang paling banyak dipergunakan di Internet. Program ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Apache mempunyai program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunaannya (Yahya, 2019). Apache ini bersifat *open source* yang berarti gratis dan bisa diedit oleh penggunaannya. Tugas utama Apache adalah menghasilkan halaman web yang benar kepada *client* berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. Secara otomatis Apache akan menjalankan file *index.html* (halaman utamanya) untuk ditampilkan secara otomatis pada *client*. Jika diperlukan

juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu *database* diakses terlebih dahulu (misalnya dalam MySQL) untuk mendukung halaman web yang dihasilkan (Pradana, 2015).

Web server Apache mempunyai kelebihan dari beberapa pertimbangan di atas (Yahya, 2019):

1. Apache termasuk dalam kategori *freeware*.
2. Apache mudah sekali proses instalasinya.
3. Mampu beroperasi pada berbagai *platform* sistem operasi.
4. Mudah mengatur konfigurasinya. Apache mempunyai hanya empat file konfigurasi.
5. Mudah dalam menambahkan peripheral lainnya ke dalam *platform* web server-nya.

Beberapa dukungan Apache (Yahya, 2019):

1. Kontrol Akses

Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP CGI (*Common Gateway Interface*). Yang paling terkenal untuk digunakan adalah Perl (*Practical Extraction and Report Language*), didukung oleh Apache dengan menempatkannya sebagai modul (*mod\_perl*).

2. PHP (*Personal Home Page/PHP Hypertext Processor*)

Program dengan metode semacam CGI yang memproses teks dan bekerja di server. Apache mendukung PHP dengan menempatkannya sebagai salah satu modulnya (*mod\_php*). Hal ini membuat kinerja PHP menjadi lebih baik.

## 2.9.2. MySQL *database*

### a. Defenisi MySQL

MySQL (*My Structured Query Language*) adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (*Database Management System*) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia (Yahya, 2019). MySQL bersifat *open source*, *multiplatform* dan berbasis data relasional. MySQL dapat dipakai untuk *database* pribadi atau pada level korporat berskala

kecil hingga besar. MySQL menggunakan SQL untuk mendukung pengaksesan data (*query*) (Wicaksono, 2017).

Dalam konteks bahasa SQL, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logika merupakan struktur 2 dimensi yang terdiri atas baris-baris data yang berada dalam satu atau lebih kolom. Baris pada *table* sering disebut sebagai *instance* dari data, sedangkan kolom sering disebut sebagai *attribute* atau *field*. Keseluruhan *table* itu dihimpun dalam satu kesatuan yang disebut *database* (Pradana, 2015).

MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL) tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL (Yahya, 2019). MySQL adalah RDBMS (*Relational Database Management System*) yang didistribusikan secara gratis di bawah lisensi GPL. Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase* (Fauzi dkk, 2019).

#### **b. Fitur-fitur MySQL**

Tidak seperti PHP atau Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas

semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael “Monty” Widenius (Yahya, 2019).

Adapun fitur-fitur MySQL antara lain (Yahya, 2019).

1. *Relational Database System*. Seperti halnya *software database* lain yang ada di pasaran, MySQL termasuk RDBMS.
2. *Arsitektur Client-Server*. MySQL memiliki arsitektur *client-server* dimana *server database* MySQL terinstal di *server*. *Client* MySQL dapat berada di komputer yang sama dengan *server*, dan dapat juga di komputer lain yang berkomunikasi dengan *server* melalui jaringan bahkan internet.
3. Mengenal perintah SQL standar. SQL (*Structured Query Language*) merupakan suatu bahasa standar yang berlaku di hampir semua *software database*. MySQL mendukung SQL versi SQL:2003.
4. Mendukung *Sub Select*. Mulai versi 4.1 MySQL telah mendukung *select* dalam *select (sub select)*.
5. Mendukung *Views*. MySQL mendukung *views* sejak versi 5.0.
6. Mendukung *Stored Prosedured (SP)*. MySQL mendukung SP sejak versi 5.0.
7. Mendukung *Triggers*. MySQL mendukung *trigger* pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan *trigger* pada versi 5.1.
8. Mendukung *replication*.
9. Mendukung transaksi.
10. Mendukung *foreign key*.

Perintah SQL terbagi ke dalam dua bagian, yaitu DDL (*Definition Data Language*) dan DML (*Definition Manipulation Language*). DDL digunakan untuk kepentingan penciptaan *database*, *table*, hingga penghapusan *database* atau *table*, misalnya CREATE DATABASE, CREATE TABLE, DROP TABLE dan ALTER TABLE. DML digunakan untuk memanipulasi data contohnya SELECT untuk mengambil data, DELETE untuk menghapus data, INSERT untuk menyisipkan data dan UPDATE untuk mengubah data (Wicaksono, 2017).

## 2.10 Penelitian Terkait

Tabel 7 Penelitian Terkait

| No. | Nama Peneliti         | Judul Penelitian                                                                                        | Hasil Penelitian                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----|-----------------------|---------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.  | Toupik Jelani (2018)  | Perancangan Alat Penghitung Jumlah Penumpang Pada Bus Angkutan Umum Berbasis Mikrokontroler Arduino Uno | Penelitian ini menggunakan mikrokontroler Arduino Uno untuk mengontrol sensor <i>infrared</i> yang akan membaca jumlah penumpang yang masuk dan keluar yang akan ditampilkan di LCD. Jika penumpang mencapai batas maksimum maka <i>buzzer</i> akan menyala dan pada LCD akan ditampilkan pernyataan “PENUH”, serta pada sistem perhitungan data dikirim melalui SMS dengan modul GSM A6 dan ada juga data yang disimpan di microSD. |
| 2.  | Hadi Syahputra (2016) | Rancang Bangun Alat Penghitung Penumpang Bus Trans Padang Berbasis Mikrokontroler ATMEGA32              | Perancangan alat ini menggunakan mikrokontroler Atmega32 dan sensor untuk mendeteksi jumlah penumpang yang turun dan naik, serta frekuensi RF Radio sebagai pengirim data yang akan ditampilkan langsung di <i>Led Dotmatrik Digital</i> . Selain itu, modul suara akan berfungsi untuk menyampaikan informasi jumlah penumpang yang ada di bus.                                                                                     |
| 3.  | Didik Aribowo (2016)  | Sistem Penghitung Jumlah Penumpang Bus Way Berbasis Mikrokontroler AT89S51                              | Sistem penghitung ini menggunakan mikrokontroler AT89S51 sebagai pengendali seluruh rangkaian termasuk di dalamnya sensor inframerah sebagai pendeteksi jumlah penumpang yang masuk sehingga dapat ditampilkan di <i>seven segment</i> . Selain menghitung jumlah penumpang, penelitian ini menggunakan <i>driver motor</i> DC untuk membuka dan menutup <i>pintu</i> bus pada saat tiba di halte.                                   |

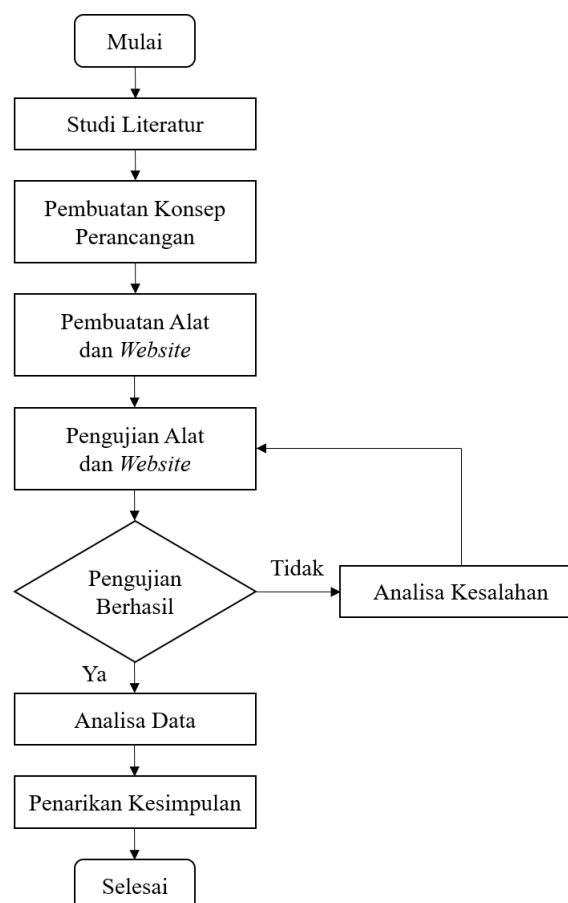
| No. | Nama Peneliti                   | Judul Penelitian                                                                                                   | Hasil Penelitian                                                                                                                                                                                                                                                                           |
|-----|---------------------------------|--------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4.  | Bangun Hardik Juniantoro (2007) | Pembuat Simulasi Penampil dan Penghitung Jumlah Penumpang <i>Bus Way</i> Menggunakan Mikrokontroler AT89S51        | Pada penelitian ini digunakan mikrokontroler AT89S51 untuk memproses data, sensor inframerah sebagai pendeteksi jumlah penumpang yang naik dan turun yang akan ditampilkan di <i>seven segment</i> . Selain itu, digunakan motor DC untuk menggerakkan pintu (membuka dan menutup).        |
| 5.  | Radhi Abdul Hakim (2019)        | <i>Prototype</i> Sistem Pemantau Lokasi Bus DAMRI dan Jumlah Penumpang Berbasis Arduino Menggunakan Mikrokontroler | Penelitian yang dilakukan menggunakan <i>smartphone</i> untuk akses lokasi, mikrokontroler NodeMCU, sensor LDR dan modul Laser untuk mendeteksi penumpang yang naik dan turun yang dikonversi menjadi data lalu ditampilkan pada aplikasi android yang terintegrasi dengan mikrokontroler. |

## BAB III

### METODE PERANCANGAN

#### 3.1 Perencanaan dan Perancangan

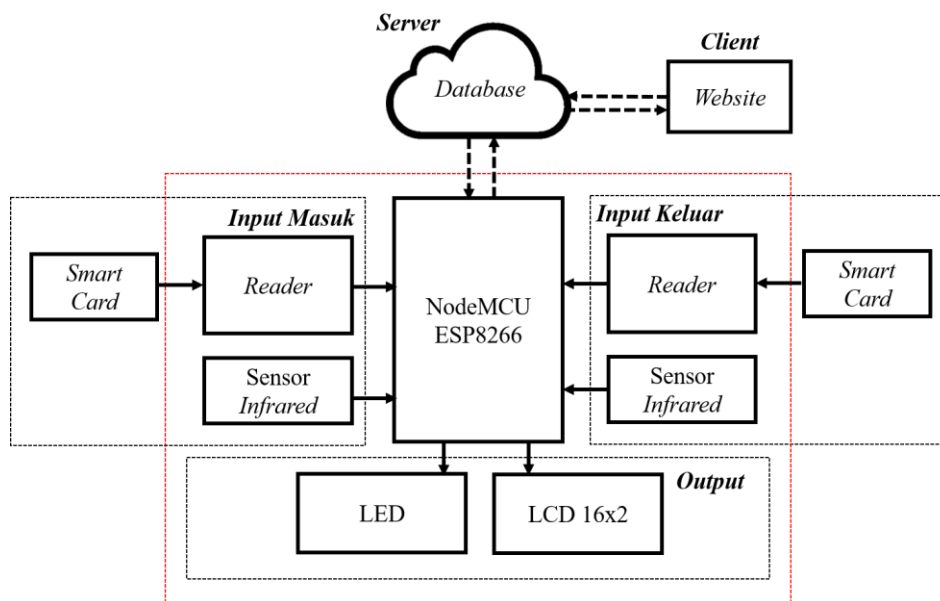
Dalam pembuatan alat penghitung penumpang bus otomatis ini dilakukan beberapa tahap perencanaan dan perancangan untuk menjelaskan proses dari awal hingga akhir proses penelitian tugas akhir ini. Dimulai dengan pengumpulan studi literatur hingga pada tahapan penarikan kesimpulan. Berikut pada Gambar 14 adalah diagram alir perencanaan penelitian dari alat penghitung penumpang bus otomatis.



Gambar 14 Diagram Alir Penelitian

Seperti yang diperlihatkan pada diagram alir penelitian pada Gambar 14, pada tahapan awal dilakukan dengan tahapan studi literatur, lalu membuat konsep

perancangan alat yang akan digunakan dalam penelitian ini. Kemudian dilanjutkan dengan proses pembuatan alat dan *website*. Setelah proses pembuatan alat dan *website*, maka dilanjutkan dengan pengujian alat dan *website* yang telah dirancang. Apabila alat dan *website* yang telah dibuat berhasil bekerja seperti yang dirancang maka akan dilanjutkan ke tahap selanjutnya yaitu pengambilan data dan analisis data. Namun, apabila belum berhasil maka akan dilakukan analisa kesalahan dan perbaikan alat dan/atau *website* kembali. Bila program telah bekerja sesuai yang telah diharapkan, maka tahapan selanjutnya yaitu penarikan kesimpulan dan pembuatan laporan dilakukan setelahnya



Gambar 15 Blok Diagram Rancangan Sistem

Pada Gambar 15, Blok Diagram sistem tersebut beberapa input dan output yang digunakan antara lain:

- a. NodeMCU ESP8266 sebagai proses mikrokontroler yang akan memproses seluruh input dan output.
- b. Input Masuk dan Keluar terdiri dari *Reader* dan *Sensor Infrared*.
  1. *Reader*, digunakan untuk membaca *Smart Card* penumpang yang berfungsi sebagai penanda *seat* penumpang di bus.
  2. *Sensor Infrared*, sebagai sensor yang mendeteksi jumlah penumpang yang masuk atau keluar.

- c. Output terdiri dari LED dan LCD 16x2.
1. LED, digunakan sebagai indikator untuk memberi informasi proses *tap Smart Card* telah berhasil terbaca dan telah terkirim ke *server*.
  2. LCD 16x2, digunakan sebagai media penampil yang akan menampilkan informasi *Smart Card* penumpang yang masuk atau keluar dan jumlah real penumpang yang berada di bus.

### 3.2 Alat dan Bahan Penelitian

Adapun Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*) yang digunakan dalam penelitian ini sebagai berikut.

Tabel 8 Perangkat Keras yang digunakan

| No. | Nama Perangkat Keras | Keterangan                                                                       |
|-----|----------------------|----------------------------------------------------------------------------------|
| 1   | NodeMCU ESP8266      | Sebagai Mikrokontroler dan modul WiFi.                                           |
| 2   | MRFC522              | Sebagai pembaca <i>Smart Card</i> .                                              |
| 3   | <i>Smart Card</i>    | Sebagai kartu identitas penumpang yang akan dibaca oleh mikrokontroler.          |
| 4   | Sensor IR FC-51      | Sebagai pendeteksi adanya penumpang.                                             |
| 5   | LCD 16x2 I2C         | Sebagai media penampil dari alat yang dibangun.                                  |
| 6   | LED                  | Sebagai indikator penunjuk <i>reader</i> berhasil membaca <i>id Smart Card</i> . |
| 7   | <i>Project Box</i>   | Sebagai <i>casing</i> modul yang sudah siap digunakan.                           |
| 8   | Kabel <i>Jumper</i>  | Sebagai Kabel penghubung                                                         |

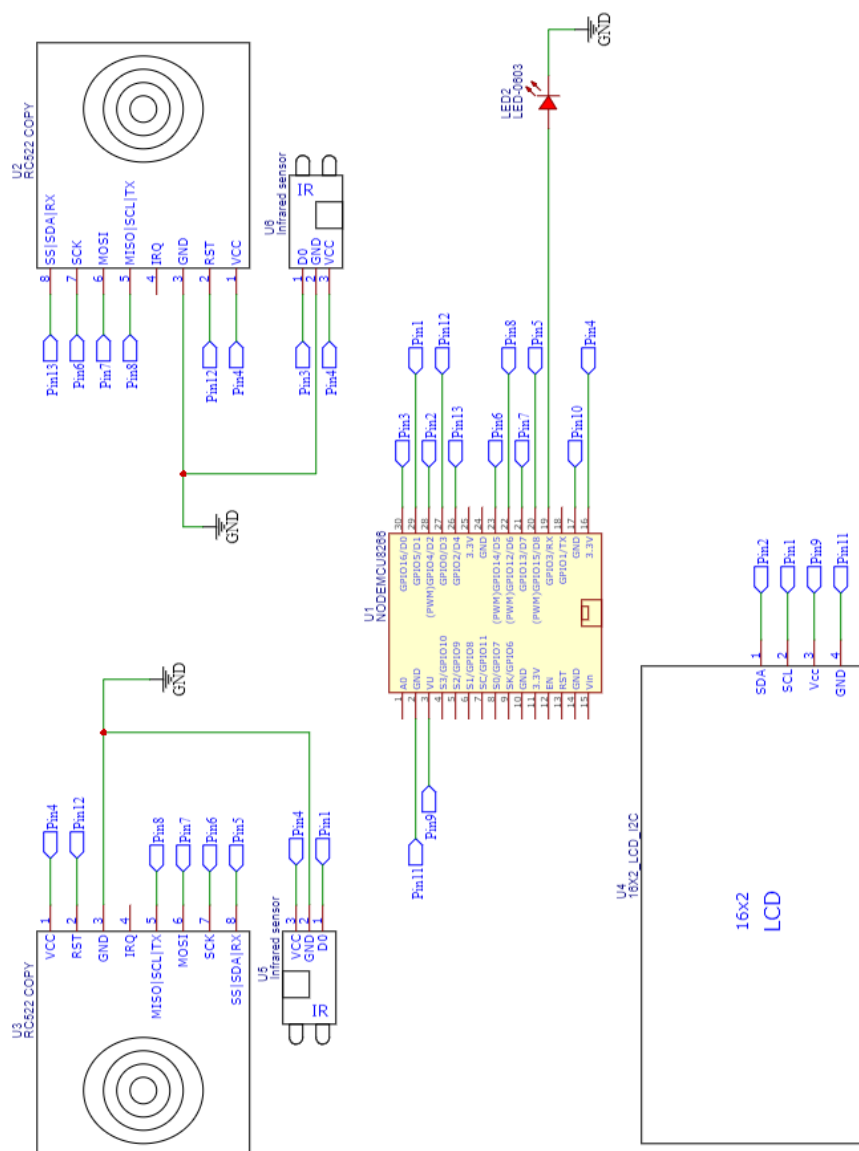
Tabel 9 Perangkat Lunak yang digunakan

| No. | Nama Perangkat Lunak | Keterangan                                                                                        |
|-----|----------------------|---------------------------------------------------------------------------------------------------|
| 1.  | Arduino IDE          | Untuk meng- <i>upload</i> program dan melihat hasil pembacaan <i>Reader</i> dari NodeMCU ESP8266. |
| 2.  | XAMPP                | Sebagai <i>server local</i> dan manajemen <i>database</i> .                                       |
| 3.  | Notepad++            | Untuk menuliskan program <i>website</i> dan <i>database</i> .                                     |

### 3.3 Perancangan Perangkat Keras

Perancangan perangkat keras dimulai dari membuat skematik rancangan alat dan merakit komponen-komponen yang dibutuhkan untuk membuat alat penghitung penumpang bus otomatis ini.

#### 3.3.1 Perancangan skematik alat

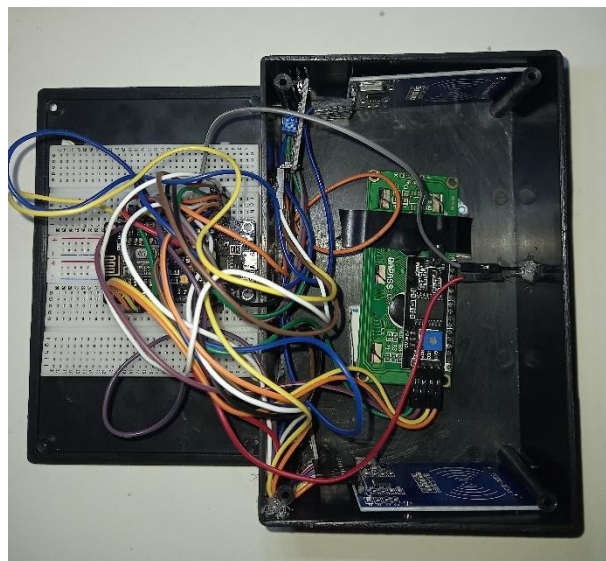


Gambar 16 Skematik Rancangan Perangkat Keras Alat Penghitung Penumpang

Hal pertama yang dilakukan sebelum merakit perangkat keras adalah membuat skematik rancangan perangkat keras seperti yang terlihat pada Gambar 16. Skematik rangkaian didesain menggunakan aplikasi easyEDA. Pengendali utama pada penelitian ini adalah NodeMCU ESP8266 yang dirancang sebagai mikrokontroler. ESP8266 dapat diaktifkan melalui koneksi USB atau dengan menggunakan catu daya eksternal. Pada penelitian ini daya yang didapatkan melalui koneksi USB. Sistem kontrol terdiri atas NodeMCU ESP8266, LCD 16x2, LED, dua MFRC522 dan dua sensor IR-FC51 yang berfungsi untuk mengontrol semua aktifitas input/output seperti mendeteksi jumlah penumpang yang masuk dan keluar, membaca *Smart Card* yang telah terdaftar di *database*, menampilkan jumlah penumpang yang berada di dalam bus dan mengirimkan informasi ke *server database*.

### 3.3.2 Perakitan alat

Pada proses perakitan alat, kabel *jumper* dihubungkan dari mikrokontroler ke komponen input dan output sesuai dengan Gambar 16. Proses perakitan alat dapat dilihat pada Gambar 17.

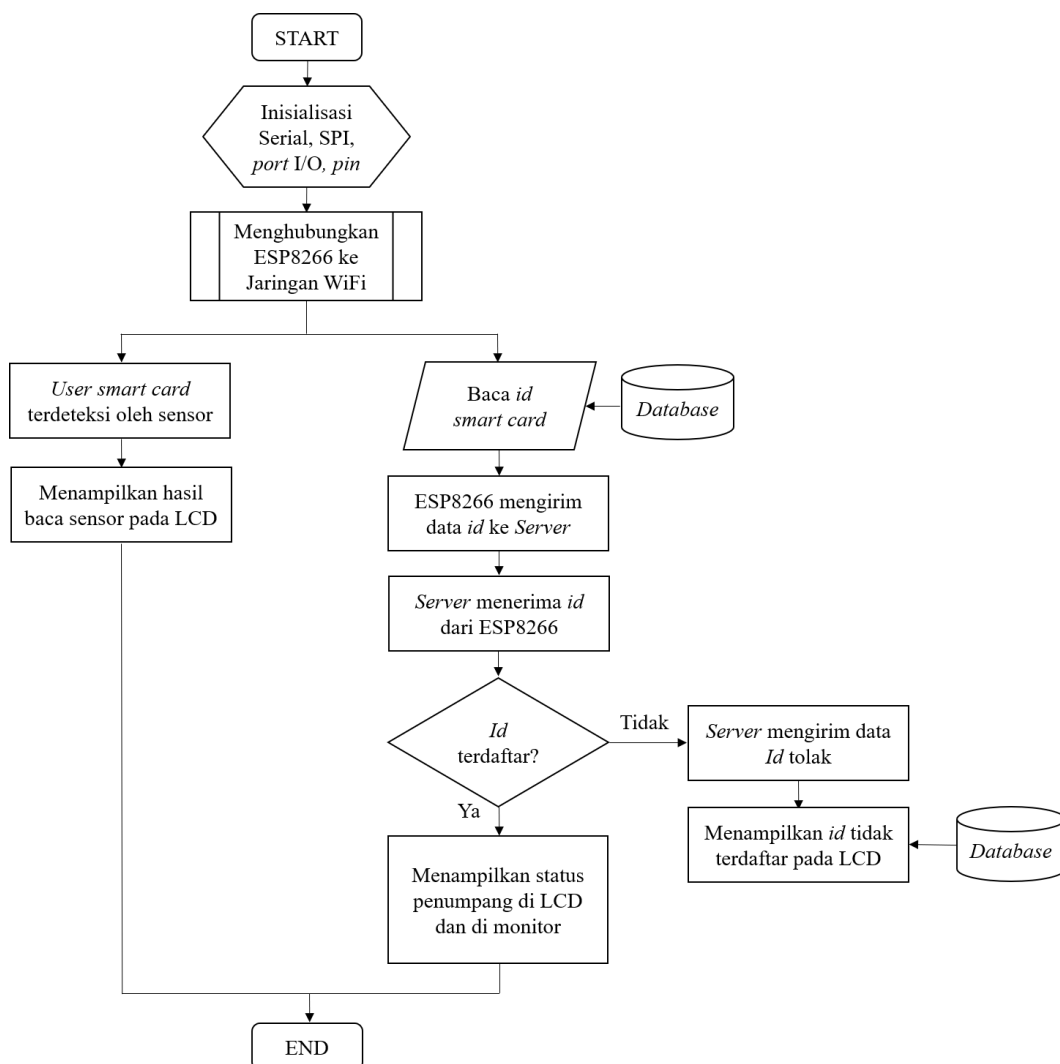


Gambar 17 Proses Perakitan Alat Penghitung Penumpang Bus Otomatis

### 3.4 Perancangan Perangkat Lunak

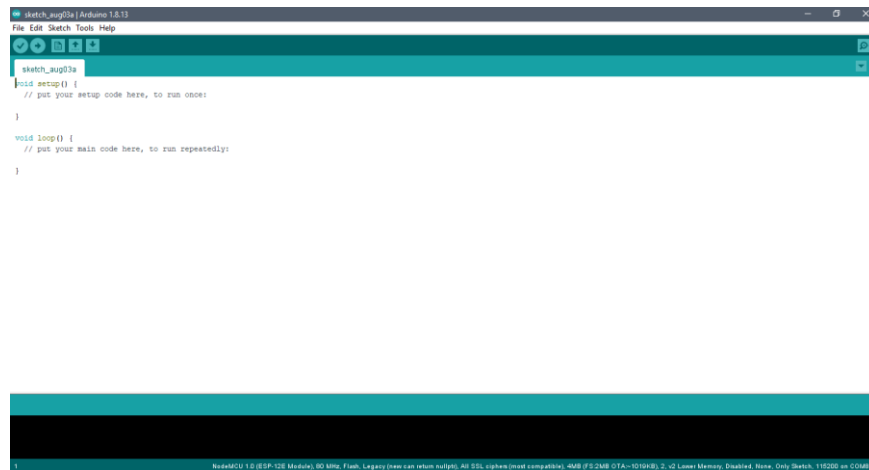
Untuk mengetahui jumlah penumpang di dalam bus secara *realtime* dibutuhkan aplikasi yang bisa menampilkan data *real* penumpang. *Listing* program pada Arduino IDE akan mengarahkan komunikasi NodeMCU ESP8266 langsung ke aplikasi *website* yang dibuat. Pada penelitian ini, *software* yang digunakan untuk membuat program dalam mengatur tampilan *database* pada *website* dibuat menggunakan *software* Notepad++.

#### 3.4.1 Perancangan program mikrokontroler



Gambar 18 Diagram Alir Sistem Alat Penghitung Penumpang Bus Otomatis

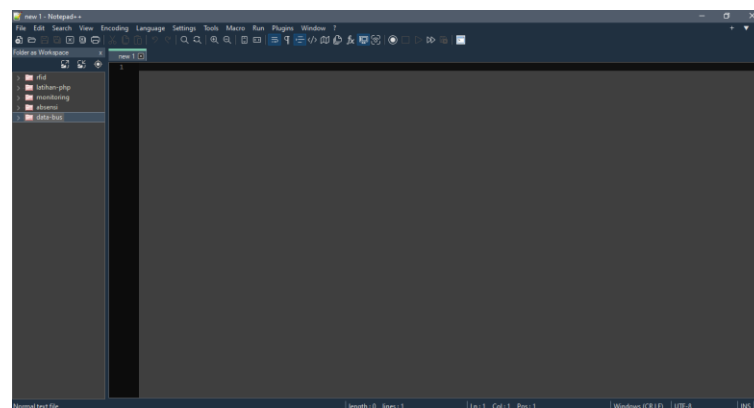
Proses *upload* program ke mikrokontroler NodeMCU ESP8266 dilakukan menggunakan Laptop atau PC menggunakan *software* Arduino IDE. Namun sebelum itu mikrokontroler harus dipastikan mendapatkan sumber daya yang memadai karena apabila tidak mendapatkan sumber daya yang dibutuhkan oleh mikrokontroler maka program tidak dapat di-*upload* ke mikrokontroler.



Gambar 19 Software Arduino IDE

### 3.4.2 Perancangan desain antarmuka *website*

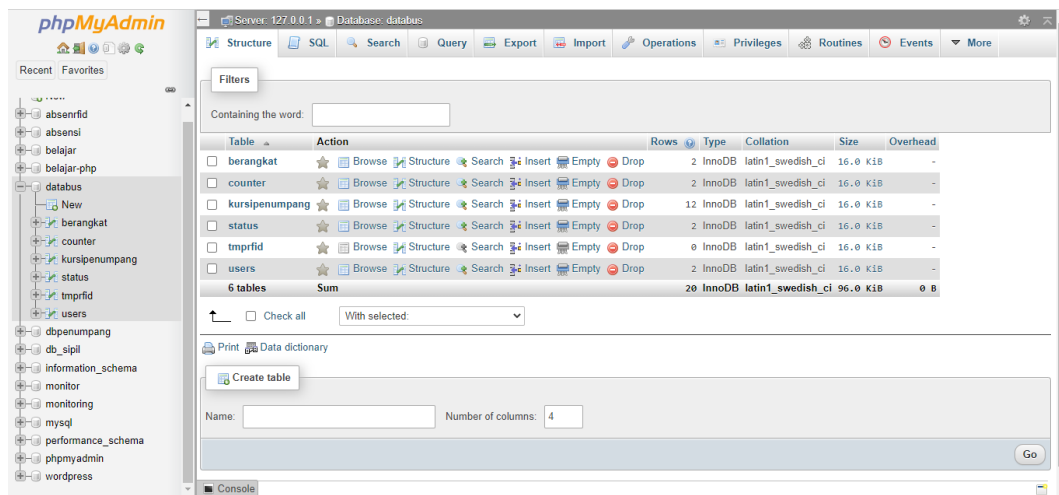
Antarmuka aplikasi disesuaikan dengan spesifikasi perangkat keras dan fungsinya yaitu menampilkan data penumpang bus yang masuk ataupun keluar. Setelah menulis kode program NodeMCU ESP8266, *website* akan terus bekerja dengan menunggu data yang masuk dari alat untuk memberi informasi data penumpang kepada pengguna. Dalam mendesain tampilan *website* digunakan perangkat lunak Notepad++ yang dapat dilihat pada Gambar 20.



Gambar 20 Tampilan Notepad++

### 3.4.3 Pembuatan database MySQL

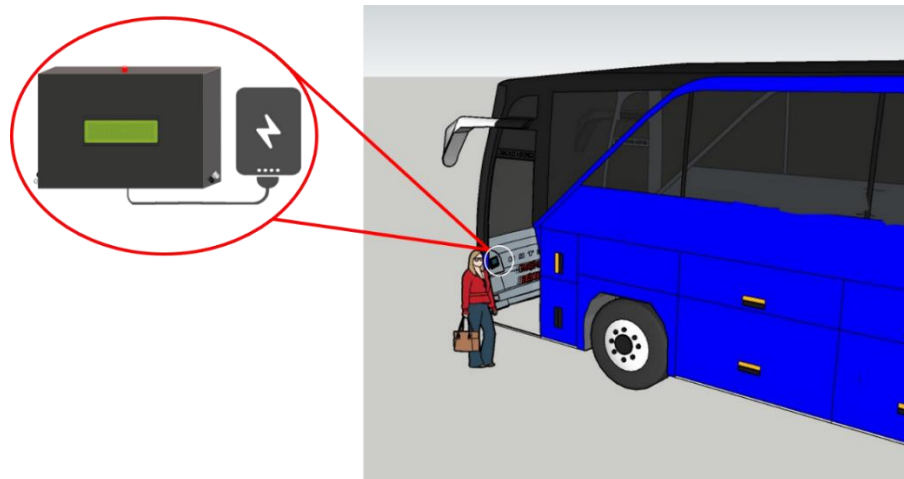
MySQL digunakan untuk kepentingan pengoperasian dan manajemen database. Adapun hal-hal yang dapat dilakukan seperti membuat database, membuat table, menambah data pada table, mengedit data pada table, hingga penghapusan database atau table. Database dari aplikasi website untuk alat penghitung penumpang bus otomatis dapat dilihat pada Gambar 21.



Gambar 21 Database Website Sistem

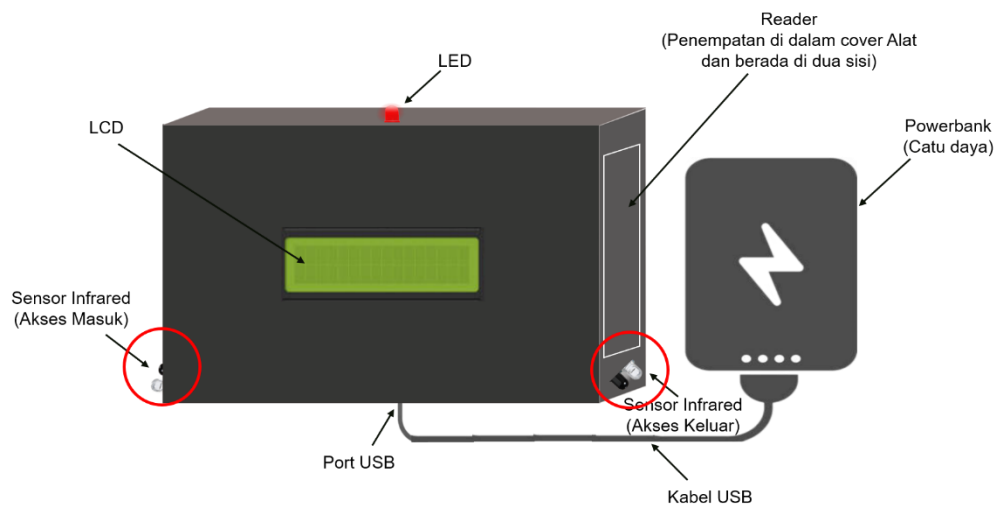
### 3.5 Perancangan Implementasi Alat

Perancangan pengujian ini dimaksudkan untuk mengetahui instalasi penempatan alat di bus dan penampilan perangkat keras pada saat telah dilakukan proses instalasi.



Gambar 22 Rancangan Instalasi Alat di Bus

Pada Gambar 22 menunjukkan letak penempatan instalasi alat dilakukan yaitu pada sisi kiri bagian depan bus setelah memasuki *pintu* bus. Adapun rancangan tampilan perangkat keras pada saat telah dilakukan proses instalasi ditunjukkan pada Gambar 23. Perangkat dihubungkan dengan catu daya berupa *powerbank*.



Gambar 23 Rancangan Perangkat Keras Sistem Penghitung

### 3.6 Perancangan Pengujian

Perancangan pengujian dimaksudkan untuk mengetahui kinerja alat yang dirancang telah bekerja sesuai dengan yang dibutuhkan pada penelitian ini.

#### 3.6.1 Perancangan pengujian RFID

Pada perancangan pengujian RFID dilakukan 3 jenis pengujian yaitu, pengujian waktu baca *reader* terhadap *Smart Card*, pengujian waktu pengiriman data dari *reader* ke *server* dan pengujian jarak baca *reader* ke *Smart Card*. Pengujian waktu baca RFID dilakukan untuk mengetahui respon RFID *reader* saat didekatkan dengan *Smart Card* dan untuk mengetahui lama waktu yang dibutuhkan *reader* untuk membaca *id Smart Card*. Pengujian ini dilakukan dengan mendekatkan *Smart Card* pada *reader*, apabila LED berkedip maka *id Smart Card* telah berhasil dibaca oleh *reader*. Pengujian dilakukan dengan menggunakan *stopwatch*.

Pengujian waktu pengiriman data dari *reader* ke *server* bertujuan untuk mengetahui lama pengiriman data dari Alat Penghitung Penumpang Bus Otomatis. Pengujian dilakukan dengan menghitung lama waktu data dikirim dan kemudian diterima oleh *server*. Pengujian ini menggunakan *timestamp* pada aplikasi Arduino IDE untuk mengetahui lama waktu pengiriman.

Pengujian jarak baca *reader* ke *Smart Card* dilakukan untuk mengetahui jarak baca *reader* terhadap *Smart Card*. Pengujian ini dilakukan dengan menggunakan penggaris.

### **3.6.2 Perancangan pengujian alat**

Pengujian ini dilakukan untuk mengetahui respon alat apabila diberikan *Smart Card* yang belum terdaftar pada *database website* dan untuk mengetahui sensor yang digunakan telah berfungsi sesuai dengan yang dirancang.

### **3.6.3 Perancangan pengujian performa alat penghitung penumpang bus otomatis**

Pengujian ini dilakukan untuk mengetahui performa dari sistem Alat Penghitung Penumpang Bus Otomatis. Pengujian dilakukan dengan menghitung reliabilitas perangkat keras yang telah dirancang.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

Bab ini akan menjelaskan hasil perencanaan alat yang telah diuji dan dianalisis untuk mengetahui kinerja sistem sesuai dengan yang telah dirancang. Proses analisis bertujuan untuk mengetahui kemampuan dan kekurangan dari Alat Penghitung Penumpang Bus Otomatis.

#### 4.1 Implementasi Alat

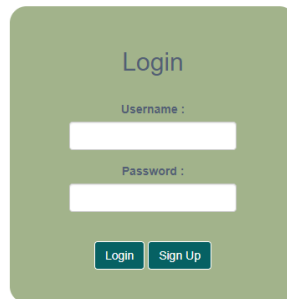
Implementasi alat terdiri dari rangkaian prototipe perangkat keras dan perangkat lunak dari Alat Penghitung Penumpang Bus Otomatis yang akan diuji dalam pengambilan data. Gambar 24 menunjukkan tampilan perangkat keras Alat Penumpang Bus Otomatis.



Gambar 24 Perangkat Keras Alat Penghitung Penumpang Bus Otomatis

*Website* dirancang dan didesain agar dapat menampilkan *database* bus, serta untuk membantu admin dalam mengubah, menghapus atau menambahkan id *Smart Card* untuk keperluan sistem *monitoring* alat penghitung penumpang otomatis di bus. Selain untuk mengubah dan mengatur *database*, *website* dapat digunakan

admin untuk keperluan informasi status pembayaran penumpang bus. Tampilan halaman *login* dan *homepage* ditampilkan sesuai pada Gambar 25 dan Gambar 26.



Gambar 25 Halaman Login Website



Gambar 26 Homepage Website

Sistem *monitoring* alat penghitung penumpang bus otomatis ditampilkan pada layar monitor sehingga memudahkan kernet atau pembantu supir dalam mengetahui keadaan terkini dari bus. Pada layar monitor ditampilkan denah bus dan informasi mengenai penumpang yang berada di bus ataupun sedang berada di luar bus ditandai dengan perubahan warna pada posisi kursi penumpang. Apabila kursi penumpang berwarna merah menandakan penumpang sedang tidak ada di dalam bus, sebaliknya apabila kursi berwarna hijau menandakan penumpang sedang berada di dalam bus. Tampilan pada layar monitor dapat dilihat pada Gambar 27.



Gambar 27 Tampilan Keadaan Bus di Monitor

#### 4.1.1 Prosedur penggunaan

Tahapan prosedur penggunaan adalah sebagai berikut.

1. *Id Smart Card* didaftarkan pada aplikasi *website*.
2. Men-Tap *Smart Card* pada *reader* alat dan sensor membaca objek.
3. LCD menampilkan jumlah penumpang *realtime* dan informasi tentang kursi penumpang, serta layar monitor menunjukkan keadaan terkini kondisi bus.

#### 4.1.2 Prosedur pengujian

Tahapan prosedur pengujian Alat Penghitung Penumpang Bus Otomatis adalah sebagai berikut.

1. Pengujian RFID
2. Pengujian Alat
3. Pengujian Performa Alat Penghitung Penumpang Bus Otomatis.

## 4.2 Hasil Uji Coba

### 4.2.1 Pengujian RFID

Pada pengujian RFID dilakukan 3 pengujian yaitu untuk mengetahui lama waktu yang dibutuhkan RFID *reader* untuk merespon *Smart Card*, mengetahui lama waktu yang dibutuhkan *server* untuk menerima dan mengirim kembali data yang dibaca oleh *reader* dan untuk mengetahui jarak baca maksimal *reader* terhadap *Smart Card*.

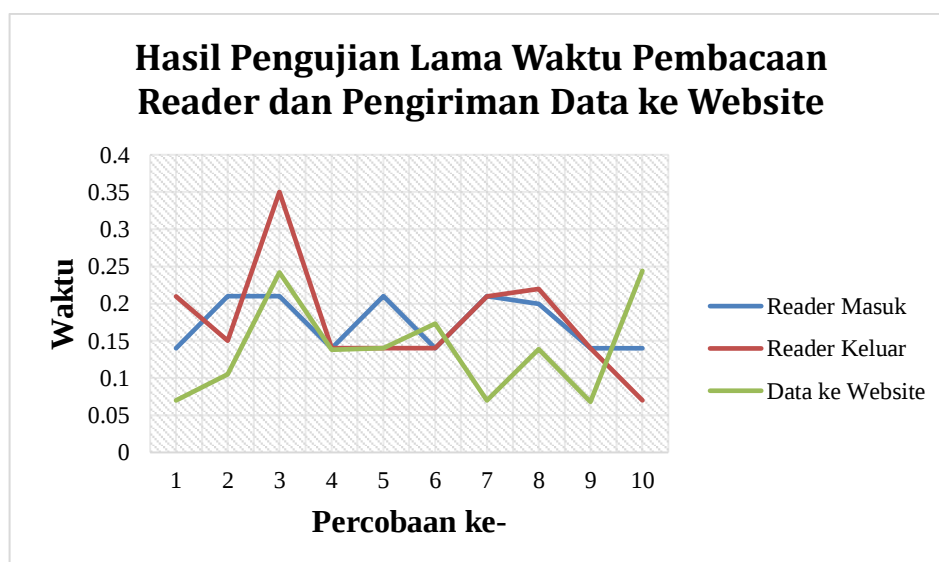
Hasil pengujian lama waktu pembacaan *reader* pada *Smart Card* dilakukan dengan menghitung lama waktu yang dibutuhkan *reader* untuk membaca *id Smart Card*. Tabel 10 menampilkan hasil uji coba pada *reader* masuk dan keluar yang dilakukan sebanyak 10 kali percobaan tiap pengujian. Dari Tabel 10 dapat dilihat bahwa waktu rata-rata yang diperoleh dari pengujian pembacaan *reader* masuk yaitu 0.174s dan waktu rata-rata yang diperoleh dari pengujian pembacaan *reader* keluar yaitu 0.177s.

Tabel 10 Hasil Pengujian Lama Waktu Pembacaan *Reader* Masuk dan *Reader* Keluar terhadap *Smart Card* dan Waktu Pengiriman Data ke *Website*

| Percobaan | Lama Waktu (s)      |                      |                        |
|-----------|---------------------|----------------------|------------------------|
|           | <i>Reader</i> Masuk | <i>Reader</i> Keluar | Data ke <i>Website</i> |
| 1         | 0.14                | 0.21                 | 0.07                   |
| 2         | 0.21                | 0.15                 | 0.105                  |
| 3         | 0.21                | 0.35                 | 0.242                  |
| 4         | 0.14                | 0.14                 | 0.138                  |
| 5         | 0.21                | 0.14                 | 0.14                   |
| 6         | 0.14                | 0.14                 | 0.173                  |
| 7         | 0.21                | 0.21                 | 0.07                   |
| 8         | 0.20                | 0.22                 | 0.139                  |
| 9         | 0.14                | 0.14                 | 0.068                  |
| 10        | 0.14                | 0.07                 | 0.244                  |
| Rata-rata | 0.174               | 0.177                | 0.1389                 |

Hasil pengujian lama waktu pengiriman data dari hasil baca *reader* ke *server* hingga pada saat *server* memberikan umpan balik dilakukan dengan menghitung lama waktu yang dibutuhkan *server* untuk menerima dan mengirim data. Pada Tabel 10 menampilkan hasil uji coba lama waktu pengiriman data ke *server* hingga *server* memberi umpan balik. Pengujian ini dilakukan sebanyak 10 kali percobaan dengan waktu rata-rata yang didapatkan yaitu 0.1389s.

Dari hasil pengujian yang dilakukan sebanyak 30 kali, 10 kali dilakukan pada pengujian lama waktu baca *reader* masuk, 10 kali dilakukan pada pengujian lama waktu baca *reader* keluar dan 10 kali dilakukan pada pengujian lama waktu pengiriman data ke *server*. Pada pengujian waktu baca *reader* masuk, waktu tercepat yang dibutuhkan *reader* untuk membaca *Smart Card* yaitu 0.14s dan waktu terlama yang dibutuhkan *reader* yaitu 0.21s. Pada pengujian waktu baca *reader* keluar, waktu tercepat yang dibutuhkan *reader* yaitu 0.07s dan waktu terlama yaitu 0.35s. Pengujian lama waktu pengiriman data ke *server* hingga *server* memberi umpan balik, waktu tercepat yang tercatat yaitu 0.07s dan waktu terlama yaitu 0.244s. Hasil pengujian tersebut dapat dilihat pada Gambar 28. Kecepatan waktu baca maupun pengiriman tidak dapat dipungkiri dipengaruhi oleh adanya *delay*. *Delay* yang terjadi diakibatkan oleh kestabilan koneksi jaringan pada saat melakukan pengujian.



Gambar 28 Grafik Perbandingan Lama Waktu Pembacaan *Reader* dan Pengiriman Data ke *Website*

Hasil pengujian jarak pembacaan *reader* pada *Smart Card* dilakukan dengan menghitung jarak antara *reader* dan *Smart Card*. Data diperoleh dengan mencatat berhasil tidaknya *reader* dalam membaca *Smart Card* pada jarak tertentu. Pada Tabel 11 menampilkan batas jarak maksimum *reader* untuk membaca *Smart Card*.

Tabel 11 Hasil Pengujian Jarak Baca *Reader* terhadap *Smart Card*

| Percobaan | Jarak (cm) | Keterangan     |
|-----------|------------|----------------|
| 1         | 1          | Berhasil       |
| 2         | 2          | Berhasil       |
| 3         | 3          | Berhasil       |
| 4         | 4          | Berhasil       |
| 5         | 5          | Tidak Berhasil |
| 6         | 6          | Tidak Berhasil |
| 7         | 7          | Tidak Berhasil |
| 8         | 8          | Tidak Berhasil |
| 9         | 9          | Tidak Berhasil |
| 10        | 10         | Tidak Berhasil |

#### 4.2.2 Pengujian alat

Pengujian alat dilakukan untuk mengetahui respon Alat Penghitung Penumpang Bus Otomatis dalam berbagai kondisi. Pengujian pertama dilakukan dengan melakukan proses tap pada *reader* saat diberi *Smart Card* yang tidak terdaftar pada *database*. Pengujian kedua dilakukan dengan menguji sensor untuk melihat hasil baca sensor sesuai yang akan ditampilkan pada LCD atau tidak.

##### 4.2.2.1 Pengujian *Smart Card* yang tidak terdaftar di *database*

Hasil pengujian *Smart Card* yang tidak terdaftar di *database* dilakukan dengan melakukan proses *tap* dengan mencoba 4 *Smart Card* yang terdiri dari 2 data yang terdaftar di *database* dan 2 lainnya tidak terdaftar di *database*. Pengujian ini dilakukan untuk mengetahui alat yang dirancang telah berfungsi dalam membaca *Smart Card* yang telah didaftar untuk keperluan alat. Hasil pengujian pada Tabel 12 menunjukkan alat telah dapat membedakan id *Smart Card* yang telah terdaftar dan belum terdaftar.

Tabel 12 Hasil Pengujian Smart Card yang Tidak Terdaftar di *Database*

| Percobaan | <i>Id Smart Card</i> | Keterangan      |
|-----------|----------------------|-----------------|
| 1         | 17019719318          | Berhasil        |
| 2         | 3323312466           | Tidak Terdaftar |
| 3         | 1714110466           | Berhasil        |
| 4         | 335511066            | Tidak Terdaftar |

Pada pengujian ini, 2 *Smart Card* yang terdaftar di *database* dan telah disimpan dengan data sebagai Kursi 10 dengan *id tag* 1714110466 dan Kursi 12 dengan *id tag* 17019719318. Keadaan awal kondisi bus sebelum pengujian dilakukan yang ditampilkan di monitor ditunjukkan pada Gambar 29. Pada monitor terdapat 2 kondisi pada letak kursi di bus. Kondisi yang pertama yaitu kursi berwarna hijau yang menandakan penumpang pada kursi tersebut sedang berada di dalam bus. Sedangkan, kondisi yang kedua yaitu kursi berwarna merah yang menandakan penumpang pada kursi tersebut tidak berada di dalam bus.



Gambar 29 Tampilan Keadaan Bus Sebelum Pengujian

Pada percobaan yang pertama dilakukan proses tap dengan menggunakan *Smart Card* yang telah terdaftar sebagai Kursi 12 dan tampilan pada LCD menunjukkan pesan “Kursi 12 Masuk!” seperti yang ditunjukkan pada Gambar 30.



Gambar 30 Tampilan LCD Saat Percobaan 1

Pada percobaan 2, 3, dan 4 dilakukan tahap yang sama yaitu melakukan proses tap *Smart Card*. Percobaan 2 menggunakan *Smart Card* yang belum terdaftar dengan *id tag* 3323312466 sehingga pada LCD ditampilkan pesan “Tidak Terdaftar!” dan pada tampilan monitor tidak terjadi perubahan. Tampilan LCD ditunjukkan pada Gambar 31.



Gambar 31 Tampilan LCD Saat Percobaan 2

Kemudian pada percobaan 3 digunakan *Smart Card* yang sudah terdaftar sebagai Kursi 10 sehingga pada LCD menampilkan pesan seperti pada Percobaan

1 sesuai dengan data kursi pada *database*. Tampilan kondisi bus pada saat percobaan 3 menunjukkan perubahan pada warna kursi 10. Pada Gambar 32 kursi 10 berwarna hijau menandakan penumpang pada kursi tersebut masih berada di dalam bus, setelah percobaan dilakukan kursi 10 berwarna merah yang menandakan penumpang telah keluar atau sedang tidak berada di dalam bus.



Gambar 32 Tampilan Monitor Saat Percobaan 3

Percobaan 4 menggunakan *Smart Card* yang belum terdaftar di *database* dengan *id tag* 335511066 dan pada LCD ditampilkan pesan “Tidak terdaftar!” seperti pada percobaan 2 dan pada tampilan di monitor tidak terjadi perubahan.

#### 4.2.2.2 Pengujian sensor

Hasil pengujian sensor dilakukan dengan memberi objek halangan untuk mengetahui hasil baca sensor sesuai dengan yang ditampilkan pada LCD. Pada Gambar 33 menunjukkan saat keadaan sensor telah membaca ada 5 objek yang telah melewati sensor masuk sehingga ditampilkan jumlah 5 orang yang berada di dalam bus. Sedangkan Gambar 34 menunjukkan saat keadaan sensor telah membaca ada 2 objek yang melewati sensor keluar sehingga menampilkan jumlah 3 orang yang berada di dalam bus pada saat itu.



Gambar 33 Tampilan LCD Saat diberi Halangan Pada Sensor Masuk



Gambar 34 Tampilan LCD Saat diberi Halangan Pada Sensor Keluar

#### 4.2.3 Pengujian performa alat penghitung penumpang bus otomatis

Pengujian ini dilakukan dengan menguji performa sistem Alat Penghitung Penumpang Bus Otomatis secara keseluruhan. Performa mengacu pada tingkat keberhasilan dari Alat Penghitung Penumpang Bus Otomatis. Pengujian performa yang digunakan yaitu perhitungan rata-rata *reliability*.

*Reliability* didefinisikan sebagai probabilitas suatu komponen atau sistem untuk melakukan fungsi yang ditentukan dalam periode waktu tertentu di bawah kondisi yang dirancang untuk beroperasi. Nilai *reliability* pada pengujian Alat Penghitung Penumpang Bus Otomatis didapatkan dengan menggunakan persamaan sebagai berikut.

$$Reliability = \frac{jumlah\ kartu\ terbaca - jumlah\ kartu\ gagal}{jumlah\ kartu} \times 100\%$$

Dari persamaan tersebut nilai *reliability* rata-rata yang didapatkan dalam penelitian ini adalah 100% dari pengujian 10 *Smart Card*.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah proses penelitian yang telah dilakukan, kesimpulan yang didapatkan pada penelitian ini adalah:

1. Perancangan alat penghitung penumpang bus otomatis pada penelitian ini dilakukan dengan perakitan dan pembuatan alat serta *website*, menggunakan sensor *infrared* sebagai pendeteksi penumpang dan menggunakan *Smart Card* sebagai *tag* penumpang sesuai posisi kursi di bus. Uji performa keberhasilan alat rata-rata yang diperoleh yaitu *reliability* 100%.
2. Sistem kendali alat penghitung penumpang bus otomatis bekerja sesuai dengan yang dirancang. Ketika penumpang bus melakukan proses tap pada *reader* dengan menggunakan *Smart Card* yang telah terdaftar maka data akan ditampilkan pada LCD dan layar monitor yang ditandai dengan perubahan warna kursi pada denah. Selain itu, saat terdeteksi penumpang melakukan proses tap masuk ataupun keluar maka jumlah penumpang di dalam bus dan informasi mengenai kursi penumpang akan tampil pada LCD. Jarak pembacaan *Smart Card* menggunakan *reader* maksimal hanya 4 cm terhadap *reader* dengan rata-rata waktu pembacaan pada tiap *reader* yaitu 0.174 detik dan 0.177 detik.
3. Sistem *monitoring* alat penghitung penumpang bus otomatis bekerja memantau keadaan bus secara *real time*. Kondisi bus ditampilkan pada layar monitor dan dimonitor oleh kernet atau pembantu supir. Selain itu, jumlah penumpang terkini di bus juga ditampilkan pada LCD. Waktu yang dibutuhkan alat untuk mengirim dan menerima hasil baca *reader* dari *server* yaitu 0.1389 detik.

## 5.2 Saran

Adapun saran-saran untuk penelitian ini, yaitu:

1. Sistem dapat dikembangkan dengan menggunakan 2 pintu pada bus untuk menjadi jalur masuk dan jalur keluar bus.
2. Sistem *monitoring* dapat dikembangkan dengan menambahkan sistem pembayaran pada *website*.
3. Posisi sensor *infrared* dapat dipindahkan agar dapat membaca aktivitas penumpang yang masuk ataupun keluar bus tanpa melakukan proses *tap* terlebih dahulu. Hal tersebut untuk mengantisipasi apabila ada penumpang yang tidak melakukan proses *tap*.
4. Pengujian alat dilakukan langsung pada bus antar daerah untuk mengetahui kondisi *real* di lapangan.

## DAFTAR PUSTAKA

- Adibrata, M. F. (2020). *Monitoring Sistem Penghitungan Barang Otomatis Menggunakan NodeMCU ESP8266*. Medan: Universitas Sumatera Utara.
- Alakel, W., Ahmad, I., & Santoso, E. B. (2019). Sistem Informasi Akuntansi Persediaan Obat Metode First In First Out (Studi Kasus: Rumah Sakit Bhayangkara Polda Lampung). *Jurnal TEKNOKOMPAK*, 13(1), 39.
- Aminuddin, M. (2021). *Perancangan Pintu Otomatis Menggunakan Sensor Temperatur Berbasis Arduino Untuk Melaksanakan Protokol Kesehatan Covid-19 di Ruang Kuliah*. Banjarmasin: Universitas Islam Kalimantan Muhammad Arsyad Al Banjari.
- Arsul, A. A. (2020). *Pengembangan Teknologi Smart Hybrid Reader Untuk Perbaikan Tata Kelola Perpustakaan di Fakultas Teknik Universitas Hasanuddin*. Makassar: Universitas Hasanuddin.
- Fadil, M., & Thamrin. (2020). Perancangan Alat Ukur Indeks Massa Tubuh (IMT) Digital Berbasis Mikrokontroler. *Jurnal Vocational Teknik Elektronika dan Informatika*, 8(1).
- Fauzi, N. A., Hapsari, G. I., & Rosmiati, M. (2019). Prototipe Sistem Monitoring Berat Muatan Truk. *e-Proceeding of Applied Science*, 5(3), 2436.
- Gintoro, T. A., & William. (2010). Sistem Smart Class Room Berbasis Smart Card dan Bahasa Pemrograman C++. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*. Yogyakarta.
- Hariyadi. (2017). *Kajian Kinerja Pelayanan Angkutan Umum Bus Antar Kota Antar Provinsi Trayek Kota Medan - Kota Subulussalam*. Medan: Universitas Muhammadiyah Sumatera Utara.
- Hartiwati, E. N. (2022). Aplikasi Inventori Barang Menggunakan Java dengan phpMyAdmin. *Cross-border*, 5(1), 603.
- Hasibuan, H. A., Kristyawati, D., Syukriah, F., & Jamilah. (2022). Rancang Bangun Prototipe Monitoring Parkir Otomatis Menggunakan Sensor Infrared Berbasis Arduino Uno. *Syntax Literate: Jurnal Ilmiah Indonesia*, 7(6), 6821.
- Irawan, J. D., Prasetio, S., & Adi, S. (2016). Pengembangan Kunci Elektronik Menggunakan RFID dengan Sistem IoT. *Industri Inovatif: Jurnal Teknik Industri*, 6(2), 28-32.
- Kamarudin, N. H. (2017). *Air Quality Sensing Device for Murdoch Library*. Perth: Murdoch University.

- Kurnia, M. (2017). *Implementasi Sistem Pengaman Sepeda Motor Menggunakan Radio Frequency Identification (RFID) dan E-KTP Berbasis Mikrokontroler*. Pekanbaru: Universitas Islam Negeri Sultan Syarif Kasim Riau.
- Last Minute Engineers. (n.d.). Retrieved Juni 8, 2022, from <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/>
- Lukmanulhakim, A. (2020). *Alat Peraga Interaktif Blok Huruf, Angka dan Kata untuk Anak Usia Dini*. Bandung: Universitas Komputer Indonesia.
- Natsir, M., Rendra, D. B., & Anggara, A. D. (2019). Implementasi IoT untuk Sistem Kendali AC Otomatis Pada Ruang Kelas di Universitas Serang Raya. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 6(1), 69-72.
- Novaldy, F., P, A. T., R, M. G., & Awan, N. (2019). *Rancang Bangun Prototype Counter Barang Berbasis Arduino Uno*. Tegal: Politeknik Harapan Bersama.
- Pradana, P. A. (2015). *Perekaman Data Akses Kamar Hotel dengan RFID Berbasis Web*. Yogyakarta: Universitas Sanata Dharma.
- Putra, D. P., & Wagyaana, A. (2021). Rancang Bangun Sistem Pendeteksi Tinggi Badan dan Antrian Otomatis Pada Pintu Masuk Wahana Bermain. *TESLA : Jurnal Teknik Elektro*, 23(1), 73.
- Saputra, D. D., & Hadijaja, D. (2019). *Sistem Kontrol Kuantitas Pakan Ayam Berbasis Arduino Mega*. Sidoarjo: Universitas Muhammadiyah Sidoarjo.
- Sarmidi, & Rahmat, S. I. (2019). Sistem Peringatan Dini Banjir Menggunakan Sensor Ultrasonik Berbasis Arduino Uno. *Jurnal Manajemen dan Teknik Informatika (JUMANTAKA)*, 3(1), 33.
- Secure Technology Alliance. (n.d.). Retrieved Juni 26, 2022, from [https://www.securetechalliance.org/resources/pdf/RFID\\_and\\_Contactless\\_Smart\\_Cards\\_FAQ\\_FINAL\\_042105.pdf](https://www.securetechalliance.org/resources/pdf/RFID_and_Contactless_Smart_Cards_FAQ_FINAL_042105.pdf)
- Septiani, M. (2019). *Efektivitas Pengelolaan dan Operasional Angkutan Trans Siginjai di Provinsi Jambi*. Jambi: Universitas Islam Negeri Sultan Thaha Saifuddin.
- Standisyah, R. E., & Restu N.S, I. (2017). Implementasi phpMyAdmin Rancangan Sistem Pengadministrasian. *Unisda Journal of Mathematics and Computer Science*, 3(2), 39-40.
- Syahputra, H. (2016). Rancang Bangun Alat Penghitung Penumpang Bus Trans Padang Berbasis Mikrokontroler Atmega32. *Majalah Ilmiah*, 23(2), 42-46.
- Tanjung, A. (2015). *Aplikasi Liquid Crystal Display (LCD) 16x2 Sebagai Tampilan Pada Coconut Milk Auto Machine*. Palembang: Politeknik Negeri Sriwijaya.

- Utama, A. D. (2010). *Perancangan Sistem Perpakiran Kendaraan Roda Empat Menggunakan Teknologi RFID di Universitas Sebelas Maret*. Surakarta: Universitas Sebelas Maret.
- Wicaksono, F. A. (2011). Pembuatan Report dan Pengaksesan Presensi Smart Card Melalui SMS Gateway. *EEPIS final project*.
- Wicaksono, M. F. (2017). Implementasi Modul Wifi NodeMCU ESP8266 Untuk Smart Home. *KOMPUTIKA-Jurnal Sistem Komputer*, 6(1), 3.
- Wicaksono, M. F., & Hidayat. (2017). *Mudah Belajar Mikrokontroler Arduino*. Bandung: Informatika Bandung.
- Yahya, R. (2019). *Pembangunan Smart Card Student dengan Teknologi NFC (Near Field Communication) di SMK ICB CT*. Bandung: Universitas Komputer Indonesia.

# LAMPIRAN

## Lampiran 1 Source Code pada Arduino IDE

```

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal_I2C.h>

#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>

//Network SSID
const char* ssid = "Galaxy A313396";
const char* password = "qvvg7555";

//pengenalan host (server)
const char* host = "192.168.137.156";

//variabel untuk LCD dan IR
LiquidCrystal_I2C lcd(0x27, 16, 2);
int irPin1 = 16;
int irPin2 = 5;
int count = 0;
boolean state1 = true;
boolean state2 = true;
boolean insideState = false;
boolean outsideIr = false;
boolean isPeopleExiting = false;
int a=1;

//sediakan variabel untuk RFID
#define SDA1_PIN 15 //D8
#define SDA2_PIN 2 //D4
#define PIN_RST 0 //D3
#define LED 3
#define JML_READER 2

//array yang menampung pin SDA
byte PIN_SDA[] = {SDA1_PIN, SDA2_PIN} ;

//variabel untuk RFID
MFRC522 mfrc522[JML_READER] ;

unsigned long previousMillis_1 = 0;
unsigned long previousMillis_2 = 0;
const long interval = 100;

void setup() {

```

```

SPI.begin();
lcd.begin(16, 2);
lcd.init();
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Selamat Datang!");

Serial.begin(9600);
pinMode(irPin1, INPUT);
pinMode(irPin2, INPUT);
pinMode(LED, OUTPUT);
digitalWrite(LED, HIGH);

//setting koneksi wifi
WiFi.hostname("NodeMCU");
WiFi.begin(ssid, password);
//cek koneksi wifi
while (WiFi.status() != WL_CONNECTED){
    //progress sedang mencari wifi
    Serial.print(".");
    digitalWrite(LED, LOW);
    delay(100);
    digitalWrite(LED, HIGH);
    delay(100);}

digitalWrite(LED, HIGH);
Serial.println("Wifi Connected");
Serial.println("IP Address : ");
Serial.println(WiFi.localIP());

for(int reader=0; reader < JML_READER; reader++){
    mfrc522[reader].PCD_Init(PIN_SDA[reader], PIN_RST);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Tap Kartu!");
    Serial.println("Dekatkan Kartu RFID Anda ke Reader");
    Serial.println();}
}

void loop() {
    for(int reader=0; reader < JML_READER; reader++){
        digitalWrite(LED, LOW);
        if(mfrc522[reader].PICC_IsNewCardPresent() &&
mfrc522[reader].PICC_ReadCardSerial()){

            //sediakan sebuah variabel untuk menampung kode tag rfid

```

```

String IDTAG = "";
for(byte i=0; i < mfrc522[reader].uid.size; i++){
    IDTAG += mfrc522[reader].uid.uidByte[i];}

digitalWrite(LED, HIGH);
digitalWrite(LED, LOW);
Serial.println(IDTAG);

//kirim nomor kartu RFID untuk disimpan ke tabel tmprfid & tmprfid
WiFiClient client;
const int httpPort = 80;
if(!client.connect(host, httpPort)){
    Serial.println("Connection Failed!");
    return;}

//uji reader
unsigned long currentMillis_1 = millis();
if (currentMillis_1 - previousMillis_1 >= interval){
    previousMillis_1 = currentMillis_1;

    if(reader==0){
        //terkirim ke link kartu masuk
        String Link;
        HTTPClient http;
        Link = "http://192.168.137.156/data-bus/kartumasuk.php?nomasuk=" +
IDTAG;
        http.begin(Link);
        int httpCode = http.GET();
        String payload = http.getString();
        Serial.println(payload);
        lcd.setCursor(0,0);
        lcd.print(payload);
        http.end();}
    }

    unsigned long currentMillis_2 = millis();
    if (currentMillis_2 - previousMillis_2 >= interval){
        previousMillis_2 = currentMillis_2;

        if(reader==1){
            //terkirim ke link kartu masuk
            String Link;
            HTTPClient http;
            Link = "http://192.168.137.156/data-bus/kartukeluar.php?nokartu=" +
IDTAG;
            http.begin(Link);

```

```

        int httpCode = http.GET();
        String payload = http.getString();
        Serial.println(payload);
        lcd.setCursor(0,0);
        lcd.print(payload);
        http.end();}
    }

    digitalWrite(LED, HIGH);

    //halt PICC / istirahatkan
    mfrc522[reader].PICC_HaltA();
    //hentikan proses enkripsi
    mfrc522[reader].PCD_StopCrypto1();}
}

if (!digitalRead(irPin1) && a==1 && state1){
    outsideIr = true;
    a++;
    state1 = false;}

if (!digitalRead(irPin2) && a==2 && state2 && count <=31){
    //Serial.println("Masuk ruangan");
    outsideIr = true;
    a = 1;
    count++;
    Serial.print("Jumlah Penumpang di Bus: ");
    Serial.println(count);
    lcd.setCursor(0,1);
    lcd.print("Jmlh di Bus: ");
    lcd.setCursor(14,1);
    lcd.print(count);
    state2 = false;}

if (!digitalRead(irPin2) && a==1 && state2){
    outsideIr = true;
    a = 2;
    state2 = false;}

if (!digitalRead(irPin1) && a==2 && state1){
    if(count > 0){
        //Serial.println("Keluar ruangan");
        outsideIr = true;
        count--;
        Serial.print("Jumlah penumpang di Bus: ");
        Serial.println(count);
    }
}

```

```

        lcd.setCursor(0,1);
        lcd.print("Jmlh di Bus: ");
        lcd.setCursor(14,1);
        lcd.print(count);
        lcd.print(" ");
        a = 1;
        state1 = false;}
    }
    if (digitalRead(irPin1)){
        state1 = true;}

    if (digitalRead(irPin2)){
        state2 = true;}
}

```

## Lampiran 2 Program Halaman Login

```

<?php
session_start();
if (isset($_SESSION['login'])){
    header("Location: index.php");
    exit;}

require 'koneksi.php';
if (isset($_POST['login'])){
    $username = $_POST['username'];
    $password = $_POST['password'];
    $hasil = mysqli_query($koneksi, "select * from users where
    username='$username'");
    if (mysqli_num_rows($hasil) === 1){
        $row = mysqli_fetch_assoc($hasil);
        if (password_verify($password, $row['password'])){
            $_SESSION['login'] = true;
            header("Location: index.php");
            exit; }
        }
    $error = true;}
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <title>Halaman Login</title>
</head>
<body>

```

```

<?php if (isset($error)){
    echo "
    <script>
        alert('Username/Password Salah');
    </script>";}
?>
<div align="right" style="padding-top: 10px; padding-right: 12px">
<a href="keberangkatan.php">
<button style="align:right; background-color: #CC704B; border-color:white"
class="btn btn-primary">Cek Bus</button></a>
</div>

<div style="padding-top: 50px">
</div>
<div class="container-fluid" align="center" style="padding-top: 30px;
background-color:#A2B38B; color: #525E75; width: 25%; min-width: 300px;
border-radius: 20px">
    <h2>Login</h2>
    <br>

    <form method="POST">
        <div class="form-group">
            <label>Username :</label>
            <input type="text" name="username" id="username" class="form-control"
            style="width: 200px" required="">
        </div>
        <div class="form-group">
            <label>Password :</label>
            <input type="password" name="password" id="password" class="form-
            control" style="width: 200px" required="">
        </div>
        <br>
        <button class="btn btn-primary" style="background-color:#066163; border-
        color:white" type="submit" name="login">Login</button>
        <input type="button" class="btn btn-primary" style="background-
        color:#066163; border-color:white" value="Sign Up"
        onclick="window.location.href='registrasi.php'" />
    </form>
    <br><br>
</div>
</body>
</html>

```

### Lampiran 3 Program Halaman Registrasi

```

<?php
session_start();
if (isset($_SESSION['login'])){
    header("Location: login.php");
    exit;}

require 'koneksi.php';
if (isset($_POST['register'])){
    if(registrasi($_POST) > 0){
        echo "
        <script>
            alert('User baru berhasil ditambahkan.');

```

```

</div>
<div class="form-group">
  <label>Konfirmasi Password :</label>
  <input type="password" name="password2" id="password2" class="form-
    control" style="width: 200px" required="">
</div>
<br>
<button class="btn btn-primary" style="background-color:#066163; border-
  color:white" type="submit" name="register">Register</button>
<input type="button" class="btn btn-primary" style="background-
  color:#066163; border-color:white" value="Login"
  onclick="window.location.href='login.php'" />

</form>
<br><br>
</div>
</body>
</html>

```

#### Lampiran 4 Program Halaman Index

```

<?php
session_start();
if(!isset($_SESSION['login'])){
  header("Location: login.php");
  exit;}
?>

<!DOCTYPE html>
<html>
<head>
  <?php include "header.php"; ?>
  <title>Menu Utama</title>
</head>
<body>
  <?php include "menu.php"; ?>

  <!-- isi -->
  <div class="container-fluid" style="padding-top: 10%; text-align: center">
    <h1>
    Selamat Datang <br>
    SISTEM PENGHITUNG PENUMPANG BUS OTOMATIS <br>
    </h1>
  </div>

  <br><br>

```

```

    <?php include "footer.php"; ?>
</body>
</html>

```

## Lampiran 5 Program Halaman Kondisi Bus Terkini

```

<?php
session_start();
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <meta http-equiv="refresh" content="3">
    <title>Cek Keberangkatan</title>

    <!-- scanning membaca kartu RFID -->
    <script type="text/javascript">
        $(document).ready(function() {
            setInterval(function() {
                $("#cekawal").load('bacakartu.php'), 0);
            });
        });
    </script>
</head>
<style>
.depanbus{width: 420px; height: 30px; background: #843f36; margin-top: 10px;
margin-right: 10px; color: white; text-align: center;}
.box1-1{width: 50px; height: 50px; background: #B73E3E; margin-right: 10px;
color: white; text-align: center;}
.box1-2{width: 50px; height: 50px; background: #54B435; margin-right: 10px;
color: white; text-align: center;}
.circle{width: 70px; height: 70px; background: #ccc; margin-top: 10px; margin-
left: 210px; border-radius: 50%; float: left; border: 8px solid #6a633e;}
</style>
<body>

    <!-- isi -->
    <div class="container-fluid" style="padding-top: 20px">
        <a href="login.php" style="padding-left: 12px">back &#8594;</a>

        <div id="cekawal"></div>
        <h3>Kondisi Bus Terkini</h3>
        <div style="display: inline-flex; ">
            <div class="depanbus">Depan Bus</div>
        </div>

```

```

<br>
<div style="display: inline-flex; padding-left: 80px">
  <div class="circle"></div>
</div>
<br>
<br>
<table class="table table-bordered" style="width: 420px">

  <?php
    error_reporting(0);
    include "koneksi.php";

    //baca tanggal saat ini
    date_default_timezone_set('Asia/Makassar');
    $tanggal = date("Y-m-d");

    $kolom = 4;
    $i = 1;

    $sql = mysqli_query($koneksi, "select a.letak, b.nokartu, b.keterangan,
    c.tanggal from counter c, berangkat b, kursipenumpang a where
    a.nokartu=b.nokartu and b.nokartu=c.nokartu order by a.letak");

    while($data = mysqli_fetch_array($sql)){
      if(($i) % $kolom==1){
        echo'<tr style="text-align: center">';}
    ?>
      <td>
        <?php

          if($data['keterangan']==1){
            echo '<p class="box1-2">'.$data['letak'].'</p>';}
          else{
            echo '<p class="box1-1">'.$data['letak'].'</p>';
          } ?>
        </td>
        <?php

          if(($i) % $kolom==0){
            echo '</tr>';}
            $i++;}
        ?>
      </table>
    </div>

    <br><br>

```

```
<?php include "footer.php"; ?>
```

```
</body>
```

```
</html>
```

## Lampiran 6 Program Halaman Data Bus

```
<?php
```

```
session_start();
```

```
if (!isset($_SESSION['login'])){
```

```
    header("Location: login.php");
```

```
    exit;}
?>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <?php include "header.php"; ?>
```

```
    <title>Data Bus</title>
```

```
</head>
```

```
<body>
```

```
    <?php include "menu.php"; ?>
```

```
<!-- isi -->
```

```
<div class="container-fluid">
```

```
    <h3>Data Bus</h3>
```

```
    <table class="table table-bordered" style="width: 25%">
```

```
        <thead>
```

```
            <tr style="background-color: #9FC088; color: white">
```

```
                <th style="width: 10px; text-align: center">No.</th>
```

```
                <th style="width: 200px; text-align: center">Daftar Bus</th>
```

```
            </tr>
```

```
        </thead>
```

```
        <tbody>
```

```
            <tr>
```

```
                <td style="text-align: center">1.</td>
```

```
                <td style="text-align: center">
```

```
                <a href="datakursi.php">Bus O1</a>
```

```
            </tr>
```

```
            <tr>
```

```
                <td style="text-align: center">2.</td>
```

```
                <td style="text-align: center">
```

```
                <a href="comingsoon.php">Bus O2</a>
```

```
            </tr>
```

```
        </tbody>
```

```

        <td style="text-align: center">3.</td>
        <td style="text-align: center">
        <a href="comingsoon.php">Bus O3</a>
    </tr>
</tbody>
</div>

```

```

<br><br>
<?php include "footer.php"; ?>

```

```

</body>
</html>

```

### Lampiran 7 Program Halaman Data Kursi Bus

```

<?php
session_start();
if (!isset($_SESSION['login'])){
    header("Location: login.php");
    exit;}
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <title>Data Kursi Bus</title>
</head>
<body>
    <?php include "menu.php"; ?>
    <a href="databus.php" style="padding-left: 12px">back &#8594;</a>
    <!-- isi -->
    <div class="container-fluid">
        <h3>Data Kursi Bus</h3>
        <table class="table table-bordered" style="width: 50%">
            <thead>
                <tr style="background-color: #9FC088; color: white">
                    <th style="width: 10px; text-align: center">No.</th>
                    <th style="width: 300px; text-align: center">No. Kartu</th>
                    <th style="width: 300px; text-align: center">Letak Kursi</th>
                    <th style="width: 200px; text-align: center">Aksi</th>
                </tr>
            </thead>
            <tbody>

                <?php

```

```

//koneksi ke database
include "koneksi.php";
//baca data penumpang
$sql = mysqli_query($koneksi, "select * from kursipenumpang order by
letak");
$no = 0;
while($data = mysqli_fetch_array($sql)){
    $no++;
    ?>

    <tr>
        <td style="text-align: center"> <?php echo $no; ?>. </td>
        <td style="text-align: center"> <?php echo $data['nokartu']; ?> </td>
        <td style="text-align: center"> <?php echo $data['letak']; ?> </td>
        <td style="text-align: center">
            <a href="edit.php?id=<?php echo $data['id']; ?>">Edit</a> |
            <a href="hapus.php?id=<?php echo $data['id']; ?>" onclick="return
            confirm('Yakin menghapus data ini?')">Hapus</a>
        </td>
    </tr>
    <?php } ?>
</tbody>
</table>

<!-- tombol tambah data karyawan -->
<a href="tambah.php"> <button class="btn btn-primary" style="background-
color:#614124; border-color:white">Tambah Data Kursi</button></a>
</div>

<br><br>
<?php include "footer.php"; ?>

</body>
</html>

```

## Lampiran 8 Program Halaman Rekap Data Admin

```

<?php
session_start();
if (!isset($_SESSION['login'])){
    header("Location: login.php");
    exit;}
?>

<!DOCTYPE html>
<html>

```

```

<head>
  <?php include "header.php"; ?>
  <title>Rekapitulasi Penumpang Bus</title>
</head>
<body>

  <?php include "menu.php"; ?>

  <!-- isi -->
  <div class="container-fluid">
    <h3>Rekap Data Bus</h3>
    <table class="table table-bordered" style="width: 25%">
      <thead>
        <tr style="background-color: #9FC088; color: white">
          <th style="width: 10px; text-align: center">No.</th>
          <th style="width: 200px; text-align: center">Daftar Bus</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td style="text-align: center">1.</td>
          <td style="text-align: center">
            <a href="counter.php">Bus O1</a>
          </td>
        </tr>
        <tr>
          <td style="text-align: center">2.</td>
          <td style="text-align: center">
            <a href="comingsoon.php">Bus O2</a>
          </td>
        </tr>
        <tr>
          <td style="text-align: center">3.</td>
          <td style="text-align: center">
            <a href="comingsoon.php">Bus O3</a>
          </td>
        </tr>
      </tbody>
    </table>

    <br><br>
    <?php include "footer.php"; ?>

  </div>
</body>
</html>

```

## Lampiran 9 Program Halaman Rekap Data Bus Terkini

```
<?php
```

```

session_start();
if (!isset($_SESSION['login'])) {
    header("Location: login.php");
    exit;
}
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <meta http-equiv="refresh" content="5">
    <title>Rekapitulasi Terkini</title>

    <!-- scanning membaca kartu RDID -->
    <script type="text/javascript">
        $(document).ready(function(){
            setInterval(function(){
                $("#cekkartu").load('bacakartu.php'), 50);
            });
        });
    </script>
</head>
<body>

    <?php include "menu.php"; ?>
    <a href="databus.php" style="padding-left: 12px">back &#8594;</a>
    <!-- isi -->
    <div class="container-fluid">
        <div id="cekkartu"></div>

        <h3>Rekap Data Penumpang Terkini</h3>
        <table class="table table-bordered" style="width: 75%">
            <thead>
                <tr style="background-color: #9FC088; color: white">
                    <th style="width: 10px; text-align: center">No.</th>
                    <th style="width: 150px; text-align: center">Letak Kursi</th>
                    <th style="width: 250px; text-align: center">Tanggal</th>
                    <th style="width: 250px; text-align: center">Jam Masuk</th>
                    <th style="width: 200px; text-align: center;">Keterangan</th>
                </tr>
            </thead>
            <tbody>
                <?php
                    include "koneksi.php";

                    $shapus = mysqli_query($koneksi, "update counter set tanggal=null,
                    jam_masuk=null where timediff(now(), waktu) >= '08:00:00' ");

```

```

date_default_timezone_set('Asia/Makassar');
$tanggal = date('Y-m-d');

$sql = mysqli_query($koneksi, "select b.id, b.tanggal, a.letak,
b.jam_masuk, b.keterangan from kursipenumpang a, counter b where
a.nokartu=b.nokartu order by a.letak");

$no = 0;
while($data = mysqli_fetch_array($sql)){
    $no++;

    ?>
    <tr>
        <td style="text-align: center"> <?php echo $no; ?> </td>
        <td style="text-align: center"> <?php echo $data['letak']; ?> </td>
        <td style="text-align: center"> <?php echo $data['tanggal']; ?> </td>
        <td style="text-align: center"> <?php echo $data['jam_masuk']; ?>
        </td>
        <td style="text-align: center">
            <?php
            if($data['keterangan']==1){
                echo '<p><a href="action.php?id='.$data['id'].'&keterangan=0">
                Lunas</a></p>';}
            else{
                echo '<p><a href="action.php?id='.$data['id'].'&keterangan=1">
                Belum Bayar</a></p>';}
            ?>
        </td>
    </tr>
    <?php } ?>
</tbody>
</table>
</div>

<br><br><br>
<?php include "footer.php"; ?>

</body>
</html>

```

## Lampiran 10 Program Halaman Coming Soon

```

<?php
session_start();
if (!isset($_SESSION['login'])){
    header("Location: login.php");
}

```

```

    exit;}
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <title>Data Bus</title>
</head>
<body>

    <?php include "menu.php"; ?>
    <a href="databus.php" style="padding-left: 12px">back &#8594;</a>
    <!-- isi -->
    <div class="container-fluid" style="width:100%; text-align: center">
        <h1>Coming Soon</h1>
    </div>

    <br><br>
    <?php include "footer.php"; ?>

</body>
</html>

```

### Lampiran 11 Program Baca Kartu

```

<?php
error_reporting(0);
include "koneksi.php";

//baca tabel tmprfid
$baca_kartu = mysqli_query($koneksi, "select * from tmprfid");
$data_kartu = mysqli_fetch_array($baca_kartu);
$nomasuk = $data_kartu['nomasuk'];
?>

<div class="container-fluid" style="text-align: center">
    <?php

        if($nomasuk==""){
        }
        else{
            $cari_kursi = mysqli_query($koneksi, "select * from counter where
            nokartu='$nomasuk'");

            date_default_timezone_set('Asia/Makassar');

```

```

$tanggal = date('Y-m-d');
$jam = date('H:i:s');
$waktu = date('Y-m-d H:i:s');

$jumlah_kursi = mysqli_num_rows($cari_kursi);
if($jumlah_kursi!=0){
    mysqli_query($koneksi, "update counter set tanggal='$tanggal',
    jam_masuk='$jam', waktu='$waktu' where nokartu='$nomasuk'");

    $data=mysqli_query($koneksi, "select * from berangkat where
    nokartu='$nomasuk'");

    if(mysqli_num_rows($data)!=0){
        mysqli_query($koneksi, "update berangkat set keterangan=1 where
        nokartu='$nomasuk'");}
    }
    //kosongkan tabel tmprfid
    mysqli_query($koneksi, "delete from tmprfid");}
?>
</div>

```

## Lampiran 12 Program Reader Masuk

```

<?php
error_reporting(0);
include "koneksi.php";

//baca nomor kartu dari NodeMCU
$nomasuk = $_GET['nomasuk'];
//kosongkan tabel tmprfid
mysqli_query($koneksi, "delete from tmprfid");

//simpan nomor kartu yang baru ke tabel tmprfid
$simpan = mysqli_query($koneksi, "insert into tmprfid(nomasuk)
values('$nomasuk')");
if($simpan){
    $carikartu = mysqli_query($koneksi, "select * from tmprfid");
    $datakartu = mysqli_fetch_array($carikartu);
    $nomasuk = $datakartu['nomasuk'];

    $cari_kursi = mysqli_query($koneksi, "select letak from kursipenumpang where
    nokartu=$nomasuk");
    $datakursi = mysqli_fetch_array($cari_kursi);
    $letak = $datakursi['letak'];

    if(mysqli_num_rows($cari_kursi)){

```

```

        echo "$letak Masuk!";}
    else{
        echo "Tidak Terdaftar!";}
}
else
    echo "Gagal Masuk";
?>

```

### Lampiran 13 Program Reader Keluar

```

<?php
error_reporting(0);
include "koneksi.php";

//baca nomor kartu dari NodeMCU
$nokartu = $_GET['nokartu'];
//kosongkan tabel tmprfid
mysqli_query($konek, "delete from tmprfid");

//simpan nomor kartu yang baru ke tabel tmprfid
$simpan = mysqli_query($konek, "insert into tmprfid(nokartu)
values('$nokartu')");
if($simpan){
    $carikartu = mysqli_query($konek, "select * from tmprfid");
    $datakartu = mysqli_fetch_array($carikartu);
    $nokartu = $datakartu['nokartu'];

    $cari_kursi = mysqli_query($konek, "select letak from kursipenumpang where
nokartu=$nokartu");
    $datakursi = mysqli_fetch_array($cari_kursi);
    $letak = $datakursi['letak'];

    if(mysqli_num_rows($cari_kursi)){
        echo "$letak Keluar!";

        $scaridata = mysqli_query($konek, "select * from berangkat where
nokartu='$nokartu'");
        $data = mysqli_num_rows($scaridata);
        if($data!=0){
            mysqli_query($konek, "update berangkat set keterangan='0' where
nokartu='$nokartu'");}
        }
    else{
        echo "Tidak Terdaftar!";}
}
else

```

```
    echo "Gagal Keluar";
?>
```

#### **Lampiran 14** Program Update Kondisi Bus Terkini

```
<?php
error_reporting(0);
include "koneksi.php";

$nokartu = $_GET['nokartu'];

$hapus = mysqli_query($koneksi, "delete from berangkat where
nokartu='$nokartu'");

if($hapus){
    echo "
    <script>
        alert('Terhapus');
        location.replace('keberangkatan.php');
    </script>";
}
else{
    echo "
    <script>
        alert('Gagal Terhapus');
        location.replace('keberangkatan.php');
    </script>";
}
?>
```

#### **Lampiran 15** Program Hapus Data

```
<?php
include "koneksi.php";

//baca id data yang akan dihapus
$id = $_GET['id'];

//hapus data
$hapus = mysqli_query($koneksi, "delete from kursipenumpang where id='$id'");

//jika berhasil terhapus tampilkan pesan Terhapus
//kemudian kembali ke data kursipenumpang
if($hapus){
    echo "
    <script>
        alert('Terhapus');
        location.replace('datakursi.php');
```

```

        </script>";}
    else{
        echo "
        <script>
            alert('Gagal Terhapus');
            location.replace('datakursi.php');
        </script>";}
?>

```

## Lampiran 16 Program Edit Data

```

<!-- proses penyimpanan -->

<?php
    error_reporting(0);
    include "koneksi.php";

    //baca ID data yang akan di edit
    $id = $_GET['id'];
    //baca data kursipenumpang berdasarkan id
    $cari = mysqli_query($koneksi, "Select * from kursipenumpang where id='$id'");
    $hasil = mysqli_fetch_array($cari);

    //jika tombol simpan diklik
    if(isset($_POST['btnSimpan'])){
        //baca isi inputan form
        $nokartu = $_POST['nokartu'];
        $letak = $_POST['letak'];

        $cekletak = mysqli_num_rows(mysqli_query($koneksi, "select letak from
        kursipenumpang where letak='$letak'"));

        if($cekletak > 0){
            echo"
            <script>
                alert('Data tidak disimpan. Sudah ada data sebelumnya.');

```

```

        echo "
        <script>
            alert('Tersimpan');
            location.replace('datakursi.php');
        </script>";}
    else{
        echo "
        <script>
            alert('Gagal Tersimpan');
            location.replace('datakursi.php');
        </script>";}
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <title>Tambah Data Kursi</title>
</head>
<body>
    <?php include "menu.php"; ?>

    <!-- isi -->
    <div class="container-fluid">
        <h3>Tambah Data Kursi</h3>

        <!-- form input -->
        <form method="POST">
            <div class="form-group">
                <label>No. Kartu</label>
                <input type="text" name="nokartu" id="nokartu" placeholder="nomor kartu
                RFID" class="form-control" style="width: 200px" value="<?php echo
                $hasil['nokartu']; ?>">
            </div>
            <div class="form-group">
                <label>Letak Kursi</label>
                <input type="text" name="letak" id="letak" placeholder="letak kursi"
                class="form-control" style="width: 200px" value="<?php echo
                $hasil['letak']; ?>">
            </div>

            <a href="datakursi.php" class="btn btn-primary">Back</a>
            <button class="btn btn-primary" name="btnSimpan"
            id="btnSimpan">Simpan</button>

```

```

        </form>
    </div>

    <br><br>
    <?php include "footer.php"; ?>
</body>
</html>

```

### Lampiran 17 Program Baca Kartu Untuk Edit

```

<?php
    error_reporting(0);
    include "koneksi.php";
    //baca isi tabel tmprfid
    $sql = mysqli_query($koneksi, "select nokartu from tmprfid");
    $data = mysqli_fetch_array($sql);
    //baca nokartu
    $nokartu = $data['nokartu'];
?>
<div class="form-group">
    <label>No. Kartu</label>
    <input type="text" name="nokartu" id="nokartu" placeholder="Tempelkan kartu
    rfid Anda" class="form-control" style="width: 200px" value="<?php echo
    $nokartu; ?>">
</div>

```

### Lampiran 18 Program Tambah Data

```

<!-- proses penyimpanan -->

<?php
    include "koneksi.php";

    //jika tombol simpan diklik
    if(isset($_POST['btnSimpan'])){
        //baca isi inputan form
        $nokartu = $_POST['nokartu'];
        $letak = $_POST['letak'];

        $cekkartu = mysqli_num_rows(mysqli_query($koneksi, "select nokartu from
        kursipenumpang where nokartu='$nokartu'"));

        if($cekkartu > 0){
            echo"
            <script>
                alert('Kartu sudah terdaftar.');
```

```

        location.replace('tambah.php');
    </script>";}
else {
    $cekletak = mysqli_num_rows(mysqli_query($koneksi, "select letak from
    kursipenumpang where letak='$letak'"));

    if($cekletak >0){
        echo "
        <script>
            alert('Sudah ada data. ');
            location.replace('tambah.php');
        </script>";}
    else{
        //simpan ke tabel kursipenumpang
        $simpan = mysqli_query($koneksi, "insert into kursipenumpang (nokartu,
        letak) values('$nokartu', '$letak')");

        //jika berhasil tersimpan, tampilkan pesan Tersimpan,
        //kembali ke data karyawan
        if($simpan){
            echo "
            <script>
                alert('Tersimpan');
                location.replace('datakursi.php');
            </script>";}
        else{
            echo "
            <script>
                alert('Gagal Tersimpan');
                location.replace('datakursi.php');
            </script>";}
        }
    }
}

//kosongkan tabel tmprfid
mysqli_query($koneksi, "delete from tmprfid");
?>

<!DOCTYPE html>
<html>
<head>
    <?php include "header.php"; ?>
    <title>Tambah Data Kursi</title>

```

```

<!-- pembacaan no kartu otomatis -->
<script type="text/javascript">
    $(document).ready(function(){
        setInterval(function(){
            $("#norfid").load('nokartu.php'), 0);});
    </script>
</head>
<body>
    <?php include "menu.php"; ?>

    <!-- isi -->
    <div class="container-fluid">
        <h3>Tambah Data Kursi</h3>

        <!-- form input -->
        <form method="POST">
            <div id="norfid"></div>
            <div class="form-group">
                <label>Letak Kursi</label>
                <input type="text" name="letak" id="letak" placeholder="letak kursi"
                    class="form-control" style="width: 200px">
            </div>

            <a href="datakursi.php" class="btn btn-primary">Back</a>
            <button class="btn btn-primary" name="btnSimpan"
                id="btnSimpan">Simpan</button>
            </form>
        </div>

        <br><br>
        <?php include "footer.php"; ?>
    </body>
</html>

```

### Lampiran 19 Program Konfirmasi Pembayaran

```

<?php
    include "koneksi.php";
    $id = $_GET['id'];
    $keterangan = $_GET['keterangan'];
    $sql = "update counter set keterangan=$keterangan where id=$id";
    $update = mysqli_query($koneksi, $sql);
    if($update){
        echo "
            <script>
                alert('Pembayaran Terkonfirmasi!');
            </script>
        "
    }

```

```

        location.replace('counter.php');
    </script>";}
else{
    echo "
    <script>
        alert('Gagal Konfirmasi!');
        location.replace('counter.php');
    </script>";}
?>

```

## Lampiran 20 Program Koneksi Website

```

<?php
//urutan = server, userdb, passdb, namadb
$konek = mysqli_connect("localhost", "root", "", "databus");

function registrasi($data){
    global $konek;
    $username = strtolower(stripslashes($data['username']));
    $password = mysqli_real_escape_string($konek, $data['password']);
    $password2 = mysqli_real_escape_string($konek, $data['password2']);
    $result = mysqli_query($konek, "select username from users where
    username='$username'");
    if (mysqli_fetch_assoc($result)){
        echo "
        <script>
            alert('Username sudah terdaftar');
        </script>";
        return false;}

    if ($password !== $password2){
        echo "
        <script>
            alert('Konfirmasi password tidak sesuai');
        </script>";
        return false;}

    $password = password_hash($password, PASSWORD_DEFAULT);
    mysqli_query($konek, "insert into users values(", '$username', '$password')");
    return mysqli_affected_rows($konek);}
?>

```

## Lampiran 21 Program Header Halaman

```

<meta charset="utf-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
<script type="text/javascript" src="jquery/jquery.min.js"></script>
<script type="text/javascript" src="js/bootstrap.min.js"></script>

```

### Lampiran 22 Program Menu Halaman

```

<nav class="navbar navbar-inverse" style="background-color: #525E75; border-color:#247881">
  <div class="container-fluid" style="color: white">
    <div class="navbar-header">
      <a href="#" class="navbar-brand">DATA BUS</a>
    </div>
    <ul class="nav navbar-nav">
      <li> <a href="index.php"> HOME </a> </li>
      <li> <a href="databus.php"> Data Bus </a> </li>
      <li> <a href="rekapbus.php"> Rekapitulasi Penumpang Terkini </a> </li>
    </ul>
  </div>
</nav>
<div align="right" style="padding-right: 12px">
  <a href="logout.php">
    <button style="align:right; background-color: #CC704B; border-color:white"
      class="btn btn-primary">Log Out</button></a>
</div>

```

### Lampiran 23 Program Footer Halaman

```

<div class="footer" style="width: 100%; height: 30px; display: flex; justify-content: center; align-items: center; background: #E4E9BE; font-size: 10px; position: fixed; bottom: 0px; z-index: 1">
  Counter Penumpang Bus Otomatis
</div>

```

### Lampiran 24 Program Logout

```

<?php
session_start();
unset($_SESSION['login']);
session_destroy();
header("Location: login.php");
?>

```

**Lampiran 25** Pengeluaran Biaya 1 (Satu) Unit Alat Penghitung

| Komponen           | Harga    | Jumlah     | Total     |
|--------------------|----------|------------|-----------|
| <i>Project Box</i> | Rp11.500 | 1          | Rp11.500  |
| NodeMCU ESP8266    | Rp40.000 | 1          | Rp40.000  |
| Sensor IR FC-51    | Rp5.500  | 2          | Rp11.000  |
| MRFC522            | Rp20.000 | 2          | Rp40.000  |
| <i>Smart Card</i>  | Rp1.690  | 10         | Rp16.900  |
| LCD 16x2 I2C       | Rp34.000 | 1          | Rp34.000  |
| LED                | Rp600    | 1          | Rp600     |
| Jumper             | Rp12.000 | Secukupnya | Rp12.000  |
| Breadboard         | Rp9.700  | 2          | Rp19.400  |
| Biaya Total        |          |            | Rp185.400 |