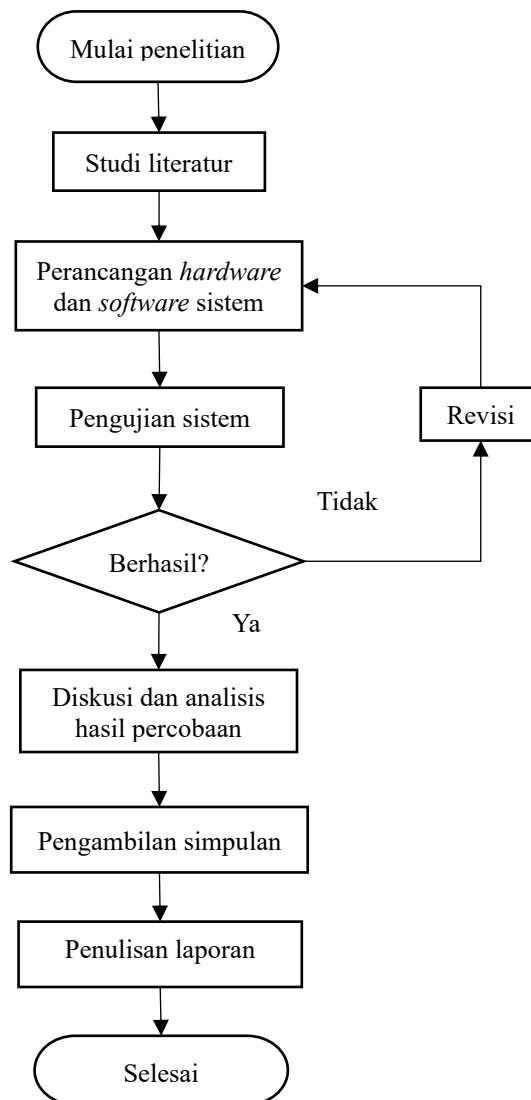


BAB III METODE PENELITIAN

3.1 Tahapan Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode eksperimen. Adapun sistem yang diharapkan ialah sebuah sistem parkir berbasis IoT untuk gedung parkir motor UPI yang dapat memberikan informasi ketersediaan lahan parkir di setiap lantai secara *real-time* melalui *website*. Sistem ini juga diharapkan dapat memberikan informasi pengendara sepeda motor yang parkir sembarangan dan akan dikenakan sanksi berupa pembebanan biaya yang ditampilkan pada sebuah *website*. Diagram alir penelitian disajikan pada Gambar 3.1.



Gambar 3.1 Diagram alir penelitian

Pada tahap awal penelitian, dilakukan studi literatur untuk mengumpulkan informasi dan teori-teori yang berkaitan dengan penelitian ini agar bisa dikaji lebih dalam. Studi literatur mencakup pencarian informasi dari buku, jurnal, publikasi dan sumber daya lain yang relevan. Tujuannya untuk mengetahui penelitian-penelitian serupa yang pernah dibuat dengan menggunakan berbagai komponen dan metode, sehingga bisa diperoleh pemahaman yang mendalam tentang bagaimana sistem tersebut dapat bekerja. Hal tersebut juga dilakukan untuk menunjang pengembangan sistem agar lebih baik.

Setelah melakukan studi literatur, langkah selanjutnya adalah perancangan sistem. Komponen yang digunakan terdiri dari komponen perangkat keras (*hardware*) dan komponen perangkat lunak (*software*). Perancangan *hardware* dilakukan dengan merangkai komponen-komponen seperti RFID *Reader* MFRC522, Mikrokontroler ESP32, Sensor Ultrasonik HC-SR04, Motor DC Servo, *Buzzer*, dan LED pada sebuah *breadboard*. Setelah dirangkai, setiap komponen dihubungkan melalui kabel *jumper*. Hal tersebut dilakukan agar semua komponen saling terintegrasi untuk membentuk purwarupa sistem. Sementara itu, perancangan *software* dilakukan dengan merancang *web server* menggunakan PHP, MySQL, Bootstrap dan Arduino IDE.

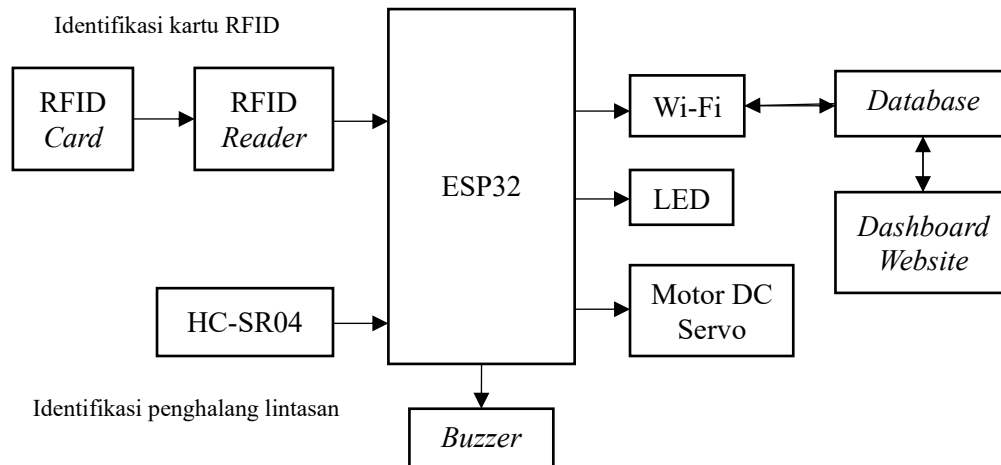
Setelah perancangan *hardware* dan *software* selesai, kemudian dilakukan pengujian pada keseluruhan sistem parkir sepeda motor pada bangunan multilantai di UPI. Pengujian dilakukan untuk memastikan sistem dapat bekerja dengan baik sesuai dengan hasil yang diharapkan. Kemudian penelitian ini diakhiri dengan menganalisis dan mengkaji sistem yang telah dibuat agar dapat ditarik kesimpulan berdasarkan rumusan masalah dan tujuan penelitian. Semua hal tersebut dikemukakan dan ditulis pada sebuah laporan akhir penelitian.

3.2 Perangkat Penunjang Penelitian

Terdapat dua perangkat penunjang penelitian yang digunakan, yaitu perangkat keras dan perangkat lunak. Perangkat keras yang digunakan di antaranya adalah RFID *Card* dan *Reader* MFRC522, Mikrokontroler ESP32, Sensor Ultrasonik HC-SR04, Motor DC Servo, *Buzzer*, Kabel *Jumper*, *Breadboard*, LED, dan Baterai. Adapun perangkat lunak yang digunakan yaitu Arduino IDE, MySQL, bahasa pemrograman PHP, Visual Studio Code dan *framework* Bootstrap.

3.3 Prinsip Kerja

Prinsip kerja dari sistem parkir untuk sepeda motor berbasis IoT untuk bangunan multilantai di UPI diperlihatkan oleh diagram blok pada Gambar 3.2. Rancangan sistem pada diagram blok tersebut akan ada di setiap palang parkir, yaitu palang parkir gerbang utama dan per lantai, baik itu pintu masuk maupun pintu keluar. Pada purwarupa sistem ini akan dibuat untuk dua lantai parkir saja.



Gambar 3.2 Diagram blok prinsip kerja sistem

Diagram blok pada Gambar 3.2 ini memperlihatkan bahwa di setiap palang parkir akan dipasang RFID Reader MFRC522, LED, dan Motor DC Servo. RFID berfungsi untuk melakukan verifikasi ID pada kartu RFID. Sedangkan LED berfungsi sebagai indikator apakah ID tersebut terdaftar atau tidak. Sementara itu, Motor DC Servo digunakan untuk membuka dan menutup palang parkir.

Pada tahap pertama dilakukan pemasukan data diri terlebih dahulu ke dalam setiap kartu RFID yang dilakukan setelah ID terbaca. Data yang dimasukkan berupa nama lengkap, NIM, program studi, fakultas, dan nomor kendaraan. Selanjutnya, kartu RFID tersebut akan dibaca oleh RFID Reader dalam sistem. Ketika kartu RFID terdaftar, maka sistem akan menyalakan LED hijau dan menggerakkan Motor DC Servo sejauh 90 derajat selama 30 detik untuk membuka palang parkir. Namun, jika ID tersebut tidak terdaftar, maka LED merah akan menyala dan Motor DC Servo tidak akan bergerak. Selain itu, setiap lantai juga akan ditentukan terlebih dahulu kapasitasnya sebagai data awal yang akan diteruskan dan diproses pada

pemrograman mikrokontroler ESP32. Ketika ada data masuk maka akan mengurangi jumlah data awal tadi.

Data yang terbaca oleh RFID *Reader* MFRC522 di palang parkir utama dan palang parkir di setiap lantai akan diteruskan ke mikrokontroler ESP32. Apabila terdapat dua data yang sama yang terbaca pada RFID *Reader* di palang parkir utama dan palang parkir di setiap lantai, maka dapat disimpulkan bahwa motor tersebut parkir sesuai tempatnya. Namun, jika hanya salah satu saja yang terbaca maka dapat disimpulkan bahwa motor tersebut tidak parkir di lahan parkir yang benar dan akan dikenakan tarif biaya parkir sebagai sanksi. Proses tersebut juga terjadi sebaliknya ketika motor hendak keluar parkir. Data dari sensor tersebut akan diolah oleh Mikrokontroler ESP32 kemudian akan langsung dikirimkan dengan konektivitas Wi-Fi menuju *database* yang di-*parsing* pada *website*. Data pengendara yang parkir beserta jumlah lahan parkir yang tersedia di setiap lantai akan dipantau secara *real time* di *website* termasuk identitas pengendara yang parkir sembarangan beserta tarif sanksi yang harus dibayarkan.

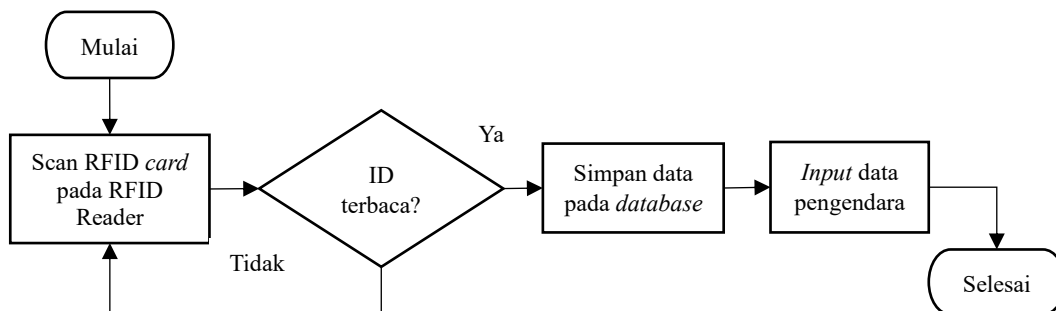
Sementara itu, Sensor Ultrasonic HC-SR04 akan dipasang di area tanjakan lintasan parkir untuk mendeteksi keberadaan objek yang menghalangi lintasan. Jika terdapat objek yang terdeteksi selama sepuluh detik, maka *Buzzer* akan menyala. *Buzzer* tersebut berfungsi sebagai peringatan apabila yang terdeteksi bukan merupakan motor melainkan orang yang sedang berada di area tersebut agar bisa menyingkir.

3.4 Algoritma

Dalam melakukan perancangan sistem parkir sepeda motor berbasis IoT pada bangunan multilantai di UPI ini, ada beberapa algoritma yang digunakan untuk menampilkan data. Beberapa algoritma tersebut sebagai berikut.

3.4.1 Algoritma pemasukan data pengendara pada kartu RFID tag

Gambar 3.3 menjelaskan algoritma dalam melakukan *input* data pengendara pada kartu RFID Tag.



Gambar 3.3 Algoritma pemasukan data pengendara pada kartu RFID

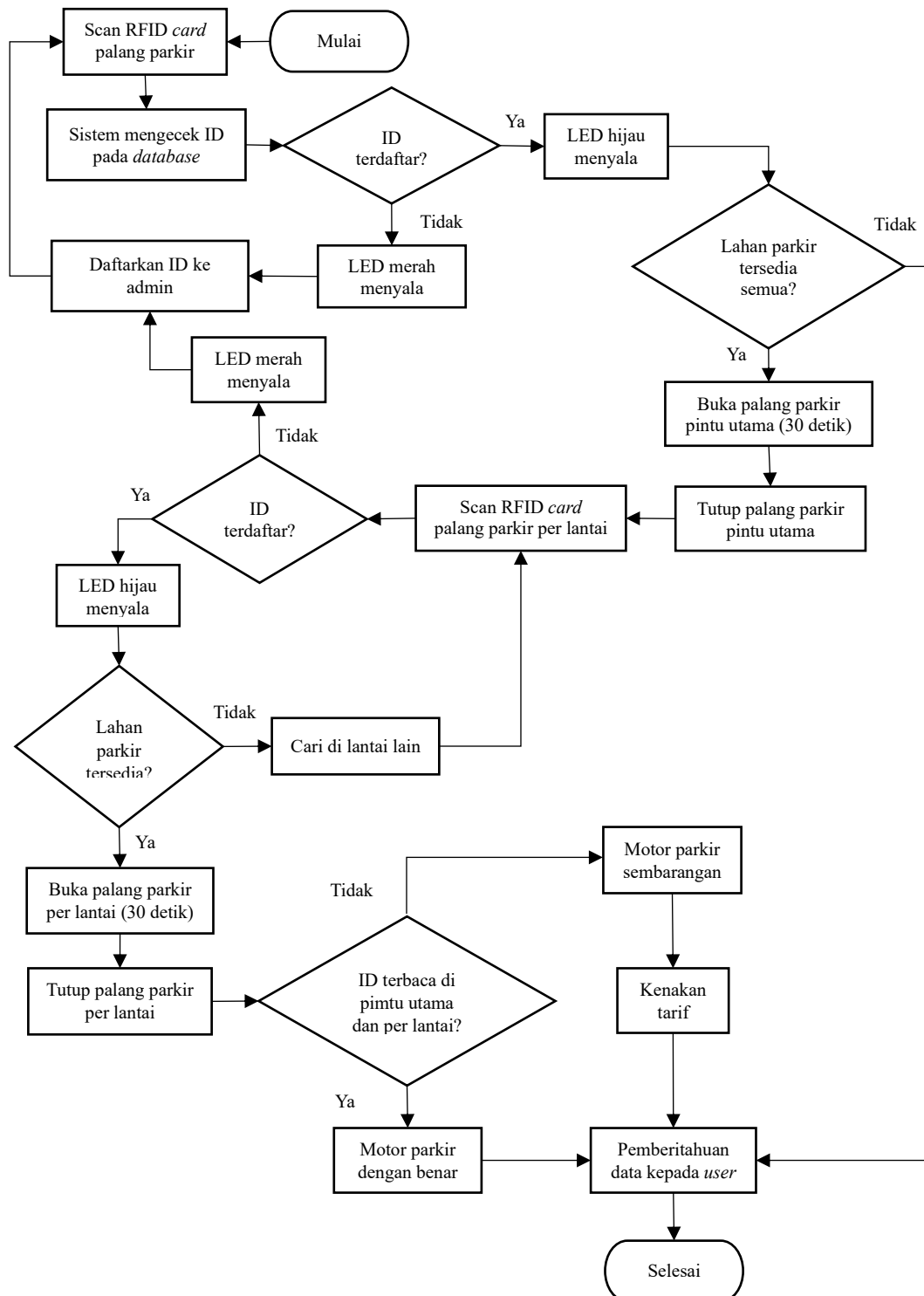
- Tahap pertama, RFID Reader MFRC522 yang terhubung pada Mikrokontroler ESP32 akan membaca ID pada kartu RFID.
- ID yang terbaca akan dilengkapi dengan identitas pengendara seperti: Nama Lengkap, NIM, Program Studi, Fakultas, dan Nomor Kendaraan.
- Data tersebut akan disimpan pada sistem pengelolaan *database* MySQL.

3.4.2. Algoritma pemrosesan sistem parkir

Algoritma pemrosesan sistem parkir dalam membuka palang parkir dan mendeteksi kendaraan yang parkir sembarangan dijelaskan pada Gambar 3.4.

- Tahap pertama, kartu RFID yang telah terdaftar akan di-*scan* pada Reader MFRC522 di pintu utama untuk dilakukan proses verifikasi *database* oleh sistem.
- Jika ID terverifikasi, maka LED hijau akan menyala. Namun, jika ID gagal terverifikasi, maka lampu LED merah akan menyala sehingga perlu dilakukan pendaftaran ID terlebih dahulu kepada petugas.
- Jika ID terverifikasi dan slot tersedia, maka Motor DC Servo akan bergerak atau palang parkir di pintu utama akan terbuka selama 30 detik, setelah itu palang parkir akan tertutup kembali.
- Setelah proses pintu utama di pintu utama selesai, maka dilanjutkan dengan proses di setiap lantai dengan cara yang sama dengan palang parkir pintu utama.
- Apabila ID terbaca pada palang parkir di pintu utama dan palang parkir per tiap lantai, maka dapat disimpulkan bahwa sepeda motor parkir dengan benar.
- Sebaliknya, apabila ID hanya terbaca di salah satu palang parkir saja baik itu di pintu parkir utama ataupun pintu parkir per lantai, maka dapat disimpulkan sepeda motor parkir sembarangan (bukan di lahan parkir yang seharusnya).

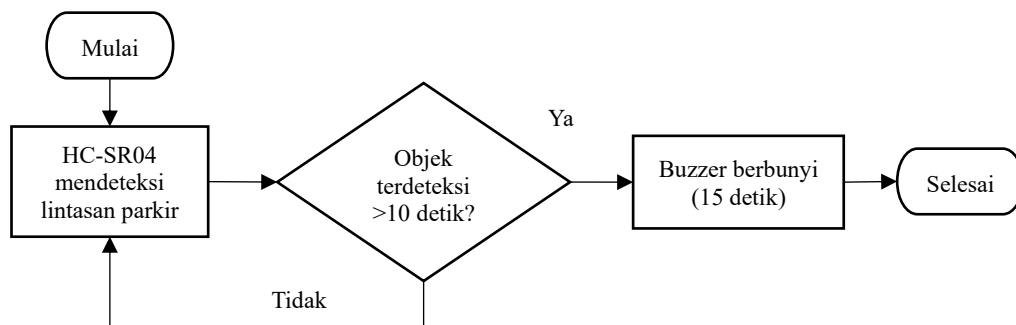
- g. Sepeda motor yang terindikasi parkir sembarangan, akan dikenakan tarif biaya sebanyak Rp 1.000,- per jam.



Gambar 3.4 Algoritma pemrosesan sistem parkir

3.4.2 Algoritma pemrosesan peringatan untuk objek yang menghalangi lintasan parkir

Pada Gambar 3.5 ditunjukkan proses pendeteksian objek yang menghalangi lintasan parkir untuk diberi peringatan.



Gambar 3.5 Algoritma peringatan untuk objek yang menghalangi lintasan parkir

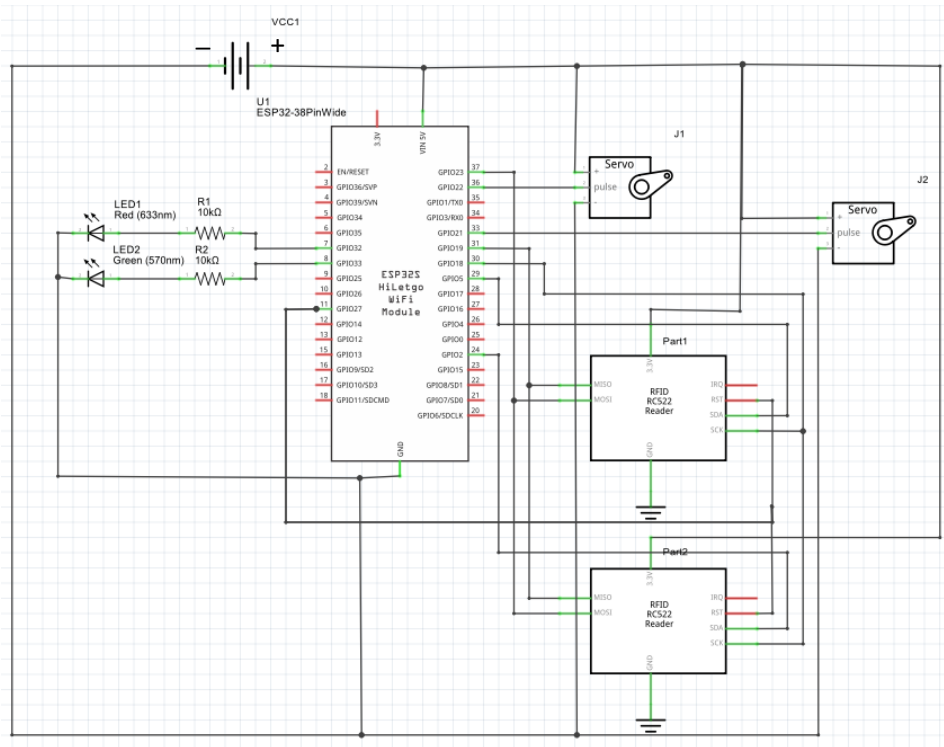
- Pada tahap awal, Sensor Ultrasonik HC-SR04 akan mendeteksi keberadaan objek di daerah lintasan parkir yaitu daerah tanjakan.
- Apabila terdapat objek yang terdeteksi selama lebih dari 10 detik, maka *Buzzer* akan menyala selama 15 detik sebagai peringatan apabila objek tersebut adalah seseorang untuk segera berpindah tempat.

3.5 Perancangan Sistem

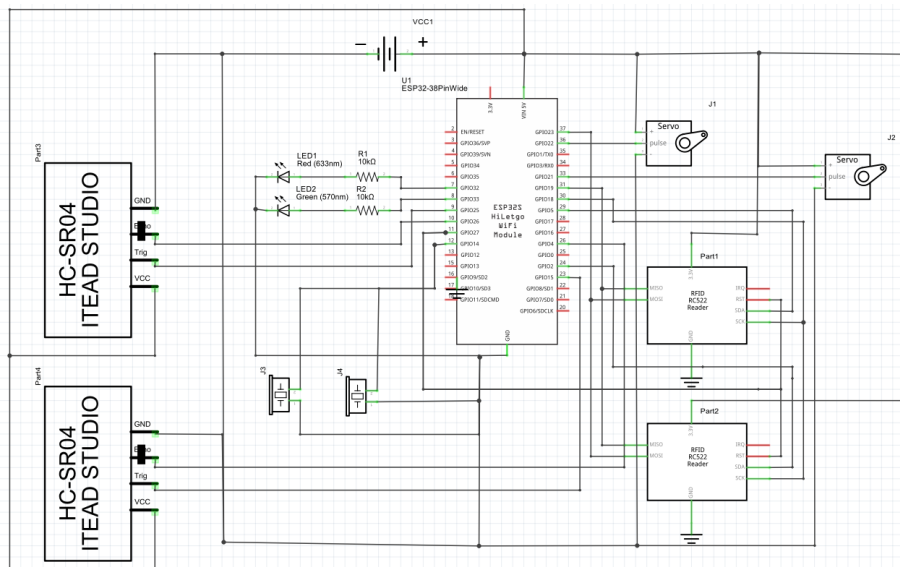
Perancangan sistem parkir ini terdiri perancangan perangkat keras dan perangkat lunak sebagai berikut.

3.5.1 Perancangan perangkat keras

Perancangan dilakukan dengan merangkai perangkat seperti *Reader* MFRC522, Mikrokontroler ESP32, Sensor Ultrasonik HC-SR04, Motor DC Servo, *Buzzer*, dan LED. Perangkat tersebut dirancang pada sebuah PCB dan saling terhubung menggunakan kabel. Rangkaian skematik perangkat dijelaskan pada Gambar 3.6 dan Gambar 3.7.



Gambar 3.6 Rangkaian skematik perangkat untuk gerbang utama



Gambar 3.7 Rangkaian skematik perangkat untuk gerbang per lantai

Tabel 3.1 menyajikan komponen perangkat keras yang digunakan dalam perancangan sistem parkir untuk sepeda motor berbasis IoT pada bangunan multilantai di UPI beserta spesifikasinya.

Tabel 3.1 Komponen perangkat keras sistem beserta spesifikasinya

No.	Komponen	Spesifikasi
1	ESP32	Mikrokontroler <i>opensource</i> (DEVKIT V1) dengan 30 GPIO dilengkapi Wi-Fi dan <i>dual-mode</i> bluetooth terintegrasi
2	MFRC522	RFID Reader modul serial 13.56 MHz untuk mengidentifikasi data
3	Micro Motor Servo SG90	Motor DC sebagai simulator palang parkir otomatis
4	LED	Lampu indikator untuk menunjukkan kebenaran ID Kartu RFID
5	Sensor Ultrasonic HC-SR04	Sensor untuk mendeteksi objek yang menghalangi lintasan parkir untuk <i>trigger</i> Buzzer
6	Buzzer	Sebagai peringatan suara apabila ada orang yang menghalangi lintasan parkir.
7	Resistor	Pembatas arus pada LED

3.5.2 Perancangan perangkat lunak

Perancangan perangkat lunak pada sistem ini menggunakan MySQL, Arduino IDE, pemrograman PHP, dan Bootstrap pada Visual Studio Code. MySQL digunakan membagi, menyimpan, dan mengolah data. Arduino IDE berfungsi sebagai media pemrograman dari sistem. Sementara *website*, akan dibuat menggunakan bahasa pemrograman PHP dan *framework* Bootstrap. *Database* yang tersimpan di *server* akan di tampilkan pada *website* yang sebelumnya di-*parsing* terlebih dahulu.

Perancangan menggunakan Arduino IDE dibuat dengan menuliskan program pada *sketch* untuk diunggah ke dalam sistem yang dapat dilihat pada Gambar 3.8. Program yang dibuat sebanyak tiga disesuaikan dengan jumlah mikrokontroler ESP32. Program yang dibuat pada *sketch* merupakan kesatuan dari keseluruhan sistem dan algoritma agar dapat bekerja sesuai dengan yang diharapkan. Hal tersebut sudah termasuk kode program untuk pengiriman data ke *server*.

```

unsigned long sekarang = 0;
unsigned long interval = 3000;
// === VARIABEL LED ===
#define led1 12
#define led2 13
#define led3 32
#define led4 33
#define buzzer 14
// ===
// === LIBRARY RFID ===
#include <SPI.h>
#include <MFRC522.h>
#include <WiFi.h>
#include <HTTPClient.h>
// === VARIABEL RFID ===
const char* ssid = "Utama"; const char* pass = "elis123456"; const String server =
"http://192.168.1.142/post.php/"; unsigned long lastTime = 0;
unsigned long timerDelay = 5000;
int gate;
String id;
String serverReal = "http://192.168.1.142";
#define PIN_RST 27
#define PIN_SDA1 2
#define PIN_SDA2 5
#define JML_READER 2
byte SDA_PIN[] = { PIN_SDA1, PIN_SDA2 };
MFRC522 mfrc522[JML_READER];
// ID yang sudah bisa di akses oleh RFID
char *UID[] = { "14419523232", "83342014", "83263016" };
int i;
// String IDTAG = "";
// === VOID SETUP RFID ===
void rfid() {
  SPI.begin(); //SPI bus
  //aktifkan reader
  Serial.println("aktifkan Reader");
  delay(2000);
  for (int reader = 0; reader < JML_READER; reader++) {
    mfrc522[reader].PCD_Init(SDA_PIN[reader], PIN_RST);
    Serial.print("reader " + String(reader));
    Serial.print(": ");
    mfrc522[reader].PCD_DumpVersionToSerial();
  }
}

```

Gambar 3.8 Cuplikan salah satu *sketch* program sistem di Arduino IDE

Sementara itu, perancangan *website* dilakukan melalui Visual Studio Code. Kode program dibuat agar menghasilkan tampilan yang berisi informasi untuk pengguna. Kode program dibuat dengan bahasa pemrograman PHP dan *framework* Bootstrap agar tampilannya lebih dinamis. Perancangan juga dilakukan untuk mengetahui komunikasi antara sistem, *web server*, dan *database* menggunakan arsitektur REST API. Mikrokontroler ESP32 dapat mengirimkan data menggunakan *http request* melalui komunikasi menggunakan *http get*. Data tersebut kemudian diterima oleh *web server* dan dilanjutkan ke *database*. Gambar 3.9 memperlihatkan salah satu kode program untuk mengambil parameter “*id*” dan “*gate*” untuk memerintahkan terbuka tidaknya palang parkir.

```

$exist = "SELECT ID FROM L1 WHERE ID = $id;";
$resmhs = $mysql -> query($exist);
$countmhs = mysqli_num_rows($resmhs);
if($countmhs>0){
    if($lantai === 'dat1'){

        $mysql -> query("ALTER TABLE 1A AUTO_INCREMENT = 1");

        $qmhs = "SELECT * FROM 1A";
        $resmhs = $mysql -> query($qmhs);
        $countmhs = mysqli_num_rows($resmhs);
        if($countmhs <500){
            $insertmhs = "INSERT INTO 1A VALUES('', '$id', current_timestamp());";

            $querymhs = $mysql -> query($insertmhs);

            if ($querymhs)
            {
                echo "In";
            }else{
                echo "InGagal";
            }
        }
    }
}

```

Gambar 3.9 Kode program untuk mengambil paramater “id” dan “gate”