

BAB 3

METODE PENELITIAN

Pada Bab III ini membahas mengenai metode penelitian yang akan dilakukan oleh penulis. Metode penelitian ini dilakukan berdasarkan tahapan-tahapan penting yang dikerjakan dengan berorientasikan kepada indikator keberhasilan dalam menghubungkan modul NodeMCU ESP8266 dan *device* lainnya sehingga dapat digunakan untuk menyelesaikan permasalahan multiobjektif. Untuk memudahkan pengawasan dan juga perawatan, digunakanlah fitur *Wifi* ESP8266. Modul ini bertujuan untuk komunikasi alat dengan telepon pintar teknisi.

3.1 ALAT DAN BAHAN

Pada penelitian yang akan dilakukan oleh penulis adanya alat dan bahan serta *software* yang akan digunakan. Alat dan bahan yang digunakan ditampilkan pada Tabel 3.1.

Tabel 3. 1 Alat Dan ahan

No	Item	Fungsi
1	<i>Software Arduino IDE</i>	<i>Software</i> yang digunakan untuk membuat <i>coding program</i> .
2	<i>Fritzing</i>	Sebagai desain skematik <i>system</i> berupa gambar skema rangkaian.
3	Laptop	Untuk memasukan <i>coding program</i> melalui <i>software Arduino IDE</i> .
4	NodeMcu ESP8266	Sebagai modul <i>Wifi</i> .
5	PCF8574	Sebagai Pin tambahan.
6	Sensor <i>ultrasonic</i> HC-SR04	Digunakan untuk mengukur jarak.
7	LCD	Sebagai mengevaluasi kemampuan <i>display</i> .
8	Motor Servo	Sensor yang menghasilkan sinyal responsif kepada sistem pengendalian guna menetapkan lokasi targetnya.

No	Item	Fungsi
9	<i>Buzzer</i>	<i>Buzzer</i> digunakan untuk memberikan peringatan suara.
10	Relay	Sebagai kontrol <i>ON</i> atau <i>OFF</i> Servo.
11	<i>Radio Frequency Identification</i> (RFID)	Sebagai pembaca Uid Tag <i>Card</i> .
12	<i>Blynk</i>	Aplikasi android sebagai kontrol dan informasi <i>output</i> perangkat.

Berikut merupakan penjelasan penggunaan alat dan bahan:

a. *Software Arduino IDE*

Berfungsi sebagai perantara mengunggah dan mengelola kode program yang akan dijalankan oleh mikrokontroler Arduino atau modul ESP8266. Melalui Arduino IDE pengguna dapat mengakses berbagai fungsi dan *library* yang mempermudah dalam mengatur sensor ultrasonik untuk mengukur jarak parkir dan dapat mengkonfigurasi aplikasi *Blynk* agar dapat saling terhubung.

b. *Laptop*

Laptop digunakan dalam penelitian ini untuk menulis kode atau memberikan instruksi kepada sistem yang telah dibuat agar dapat dijalankan. Ini dilakukan dengan menggunakan perangkat lunak Arduino IDE untuk mengatur pengaturan dan konfigurasi ke dalam modul yang digunakan.

c. *PCF8574*

PCF8574 digunakan untuk menyediakan fleksibilitas lebih dalam desain sirkuit yang memerlukan lebih banyak I/O daripada yang tersedia pada mikrokontroler.

d. *NodeMCU ESP8266*

NodeMCU ESP8266 memiliki suatu bagian yang disebut modul *WiFi* yang membuatnya bisa berhubungan dengan internet dan dapat menghubungkan aplikasi *Blynk*.

e. *Sensor Ultrasonik HC-SR04*

Sensor ini berfungsi untuk mengukur jarak antara sensor dan objek dengan menggunakan gelombang ultrasonik. Dengan menggunakan waktu tempuh

gelombang ultrasonic. Informasi jarak yang diperoleh dari sensor ini akan digunakan sebagai masukan untuk sistem parkir berbasis IoT.

f. *Liquid Crystal Display (LCD)*

Liquid Crystal Display (LCD) dipergunakan untuk memvisualisasikan hasil pengukuran jarak yang diperoleh dari sensor ultrasonik yang terintegrasi dalam sistem.

g. *Buzzer*

Buzzer digunakan untuk memberikan peringatan suara ketika kendaraan mendekati objek atau dinding di sekitarnya. Semakin mendekat ke objek, frekuensi bunyi *buzzer* dapat meningkat.

h. *Motor Servo*

Motor Servo digunakan untuk menggerakkan dan mengatur posisi objek Servo bekerja dengan menerima sinyal kontrol yang menentukan sudut putaran porosnya, kemudian servo dapat menggunakan mekanisme umpan balik untuk menyesuaikan dan mempertahankan posisi yang diinginkan.

i. *Relay*

Relay digunakan untuk mengaktifkan atau mematikan perangkat listrik yang beroperasi pada tegangan dan arus yang lebih tinggi daripada yang bisa langsung ditangani oleh mikrokontroler.

j. *Radio Frequency Identification (RFID)*

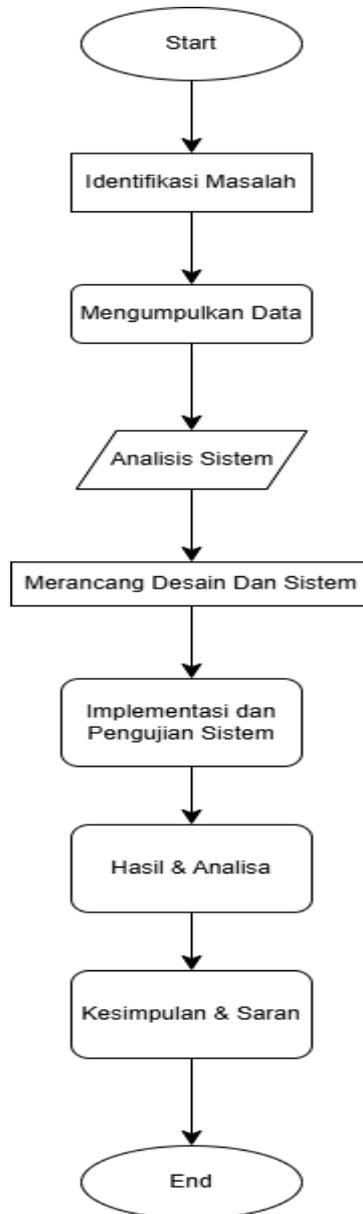
RFID digunakan sebagai pembaca Id card yang memancarkan sinyal radio dan mengaktifkan tag yang memungkinkan pembaca untuk memperoleh data dari tag tersebut. RFID juga menyediakan metode cepat dan akurat dalam mengidentifikasi dan melacak objek.

k. *Blynk*

Sebagai aplikasi penampil *output* dari *coding* program alat yang akan dibuat yang mana berupa tingkat sudut dan peringatan terjadinya tanah longsor *Blynk server*.

3.2 ALUR PENELITIAN

Pada alur penelitian ini akan disusun dengan beberapa tahap yang dilakukan agar mempermudah proses penelitian. Alur penelitian ini dapat dilihat pada gambar 3.1.



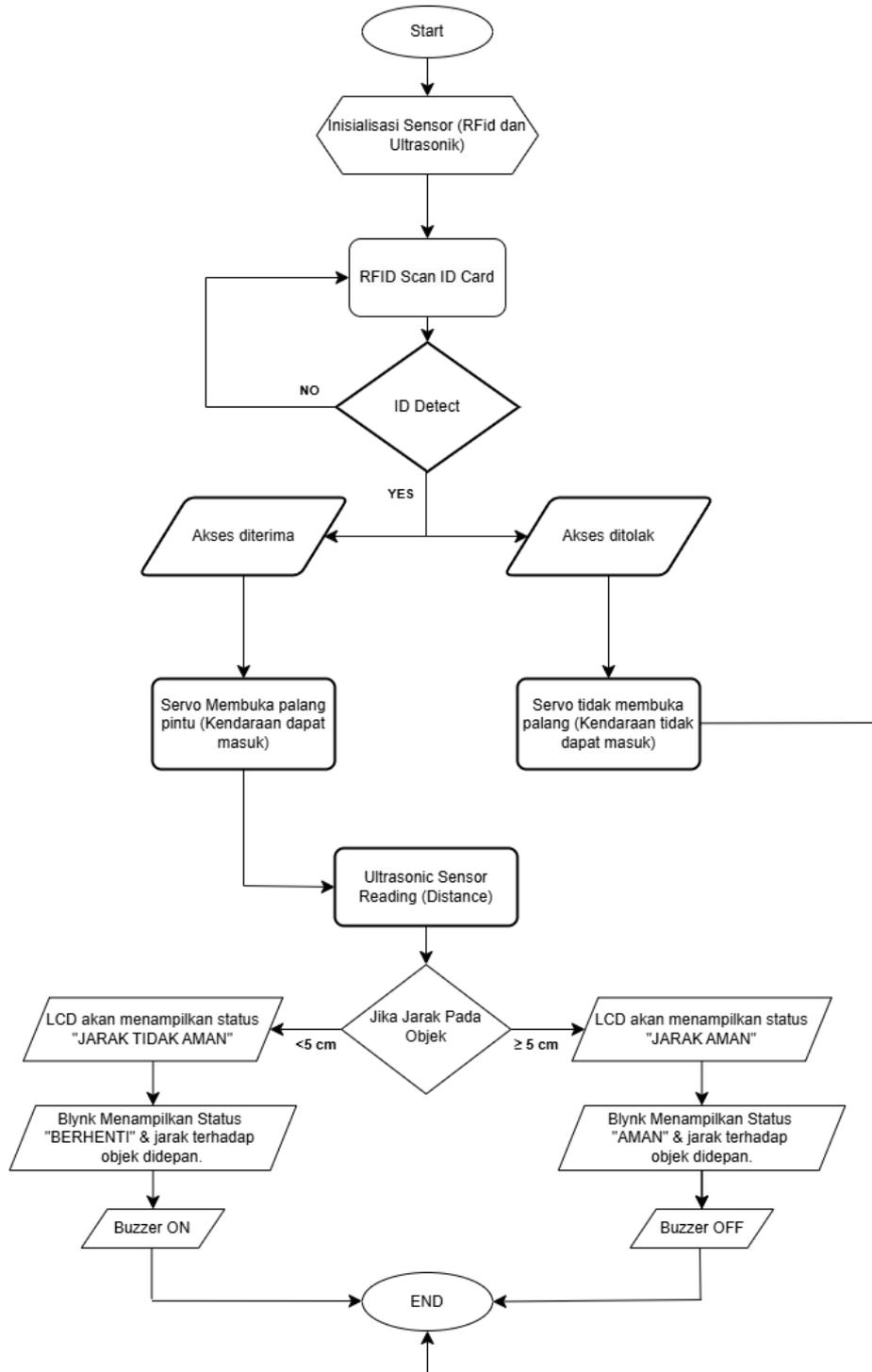
Gambar 3. 1 Alur Penelitian

3.3 PERANCANGAN SISTEM

Pada bagian ini dijelaskan bagaimana perancangan sistem dari alat perancangan sensor parkir berbasis IoT dengan sensor ultrasonik dan Esp8266 yang akan dibuat.

3.3.1 Flowchart Alur Sistem

Merancang sistem pada penelitian ini merupakan Langkah awal dalam pembuatan *prototype* sistem yang akan diimplementasikan. Penulis menyusun diagram alur dari sistem sensor parkir dengan menggunakan sensor ultrasonik HC-SR04. *Flowchart* yang disusun dapat dilihat pada gambar 3.2.

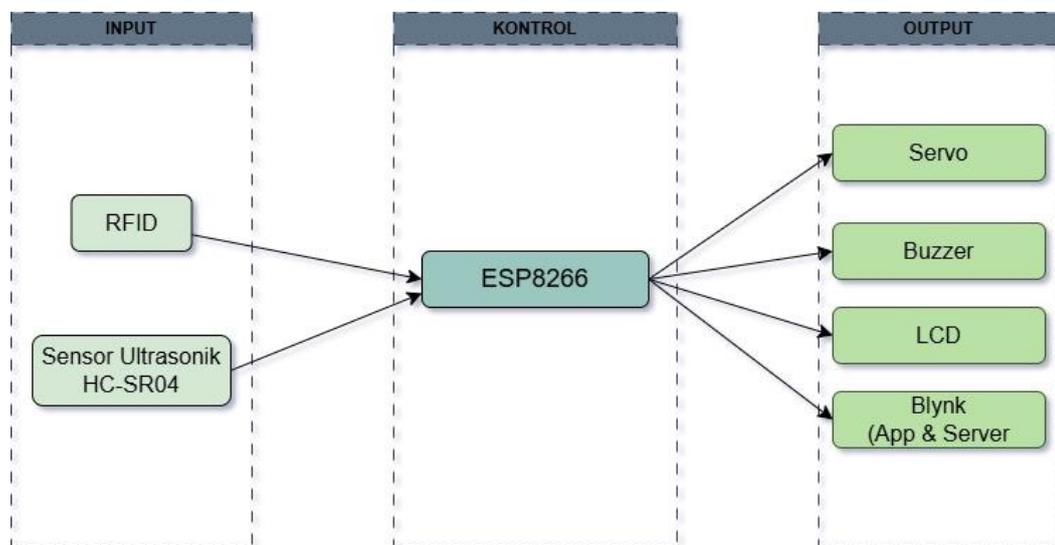


Gambar 3. 2 *Flowchart* Perancangan Sistem

Pada gambar 3.2 yang disajikan oleh RFID sebagai pembaca Kartu Tag Uid pengendara, jika id tag telah terdaftar, RFID akan mengirimkan data ke motor servo untuk membuka pintu parkir. Setelah itu, sensor *ultrasonic* HC-SR04 akan digunakan untuk mendeteksi kendaraan yang akan masuk ke zona parkir, data deteksi kemudian diproses oleh NodeMcu ESP8266. Jika proses pengukuran jarak kendaraan menunjukkan jarak kurang dari 5 cm, LCD dan *Blynk* akan menampilkan status "BERHENTI" sebagai peringatan kepada pengendara, sementara *buzzer* akan mengeluarkan suara beep pendek secara berulang sebagai tanda peringatan bahwa kendaraan telah melampaui batas jarak minimum. Namun, jika jarak lebih dari sama dengan 5 cm, LCD akan menampilkan status "JARAK AMAN" sebagai indikasi jarak yang aman, dengan *Blynk* menampilkan notifikasi "Aman" dan *buzzer* akan nonaktif. Selanjutnya, LCD akan menampilkan hasil pengukuran jarak dan status peringatan yang disebutkan sebelumnya.

3.3.2 Diagram Blok Sistem

Diagram blok sistem dapat digunakan sebagai perancangan alat, sistem dan aplikasi *Blynk* yang dapat mengontrol sensor jarak kendaraan dengan objek yang ada disekitarnya. Berikut merupakan gambar blok diagram sistem sensor jarak pada gambar 3.3.

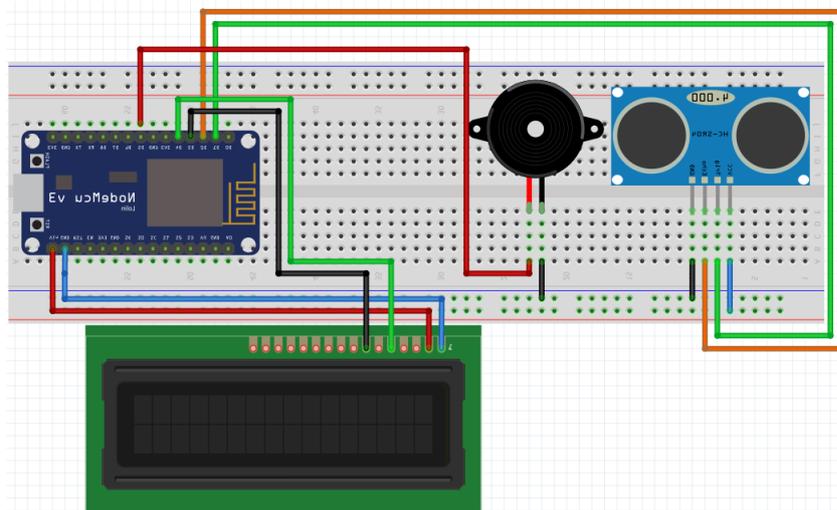


Gambar 3. 3 Blok Diagram Sistem Penelitian

Pada gambar 3.3 Sensor *ultrasonic* HC-SR04 dan Rfid terletak pada komponen input yang bertugas mendeteksi jarak antara kendaraan dan objek di sekitarnya. Di sisi sistem kontrol, NodeMcu ESP8266 bertindak sebagai modul

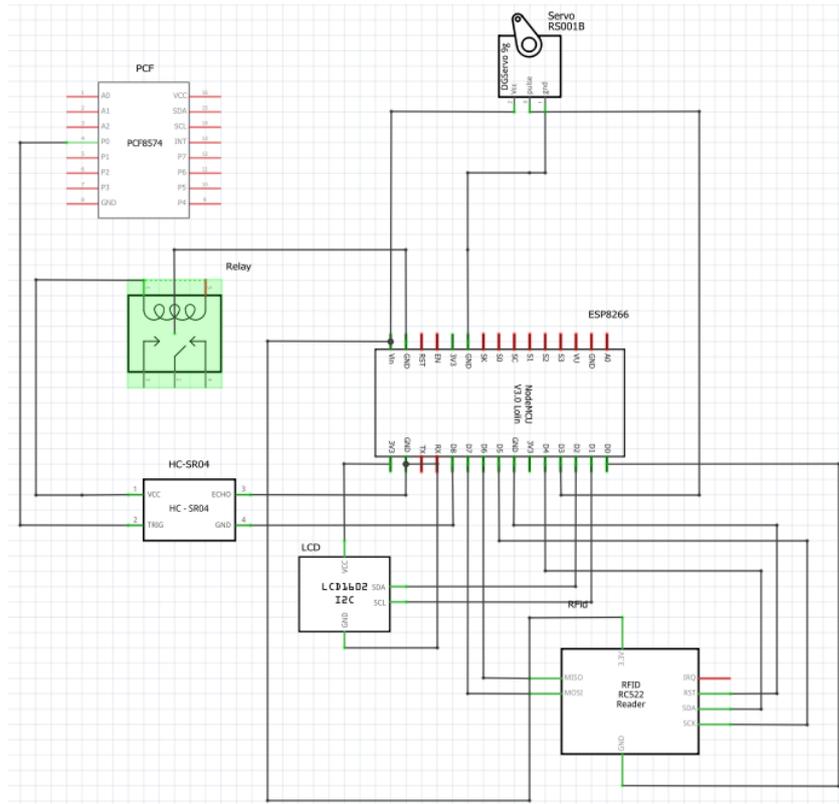
WiFi yang memungkinkan pemantauan dan pengiriman data status parkir dari jarak jauh. Data mengenai ketersediaan tempat parkir dan deteksi jarak aman dapat diakses melalui *platform* aplikasi *online*. Pada sensor ultrasonik HC-SR04 dapat dikontrol melalui NodeMcu ESP8266. Kemudian *input* yang telah dikontrol dapat menghasilkan *output* yaitu, Motor Servo, *Blynk*, *Buzzer* dan LCD. Pada Motor servo digunakan untuk membuka atau menutup pintu parkir. Servo juga dapat disetting melalui *Blynk*. *Buzzer* berfungsi sebagai alarm sistem. *Buzzer* akan menyala jika jarak kendaraan dengan objek <100 cm dan *buzzer* akan *off* jika jarak kendaraan dengan objek >200 cm. Selanjutnya, terdapat LCD sebagai *output* dari NodeMcu ESP8266. LCD akan memberi informasi visual kepada pengemudi tentang seberapa dekat kendaraan dengan objek di sekitar. *Blynk* menghasilkan *output* pada server *Blynk* yang berfungsi sebagai penampilan *widget* atau elemen yang dapat digunakan pada *Blynk* label sebagai jarak kendaraan dan *button* sebagai *on* atau *off* sistem, lalu *Blynk* juga menghasilkan *output* pada *buzzer*.

3.3.3 *Wiring Schematic*



Gambar 3. 4 *Wiring Schematic*

Pada gambar 3.4 merupakan konteks perancangan sebuah *Wiring Schematic* adalah representasi visual dari susunan komponen-komponen dalam suatu rangkaian, yang dirancang untuk memperlihatkan struktur alat yang akan dirancang.



Gambar 3. 5 Skematik Perancangan *Hardware*

Gambar 3.5 menunjukkan komponen-komponen rangkaian yang disusun untuk menghasilkan bentuk alat yang akan dirancang. Berikut konfigurasi pin yang digunakan untuk mengetahui seluruh komponen.

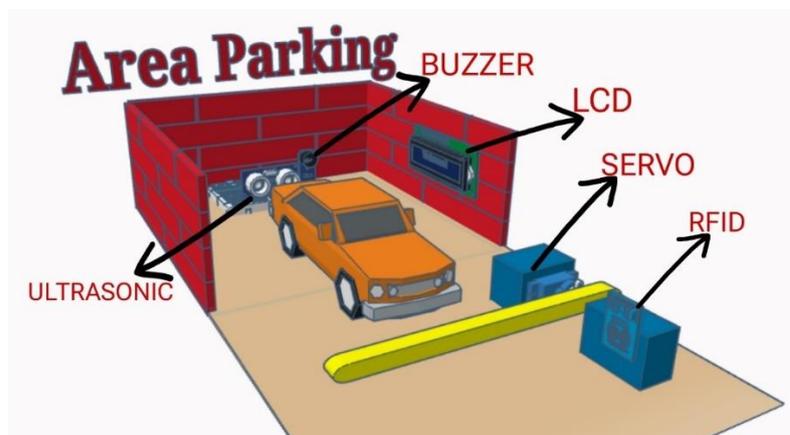
Tabel 3. 2 Konfigurasi Pin Seluruh Komponen

Modul	Pin Modul	Wire
Sensor Ultrasonik HC-SR04	VCC	Relay Pin NO
	Trig	PCF8574 Pin P0
	Echo	ESP8266 Pin D8
	GND	GND
RFid	SDA	ESP8266 Pin D4
	SCK	ESP8266 Pin D5
	MOSI	ESP8266 Pin D7
	MISO	ESP8266 Pin D6
	GND	GND
	RST	ESP8266 Pin D0
	VCC	VCC

Modul	Pin Modul	Wire
Servo	Signal	ESP8266 Pin D3
	VCC	VCC
	GND	GND
Relay	IN	PCF8574 Pin P2
	GND	GND
	VCC	VCC
Buzzer	Input	VCC
	GND	PCF8574 Pin P1
LCD	GND	GND
	VCC	VCC
	SDA	ESP8266 Pin D2
	SCL	ESP8266 Pin D1

3.3.4 Ilustrasi Perancangan Model

Ilustrasi perancangan model ini menggambarkan proses penyusunan sistem yang melibatkan berbagai komponen elektronik dan perangkat lunak. Dalam desain ini, NodeMCU (ESP8266) digunakan sebagai mikrokontroler utama yang mengendalikan sensor RFid, sensor ultrasonik HC-SR04, *buzzer* dan LCD. Komponen-komponen ini dihubungkan pada *board shield* untuk memudahkan pengujian dan modifikasi. NodeMCU berfungsi untuk menerima data dari sensor, dan kemudian menampilkan informasi tersebut pada LCD serta memberikan umpan balik melalui *buzzer*.

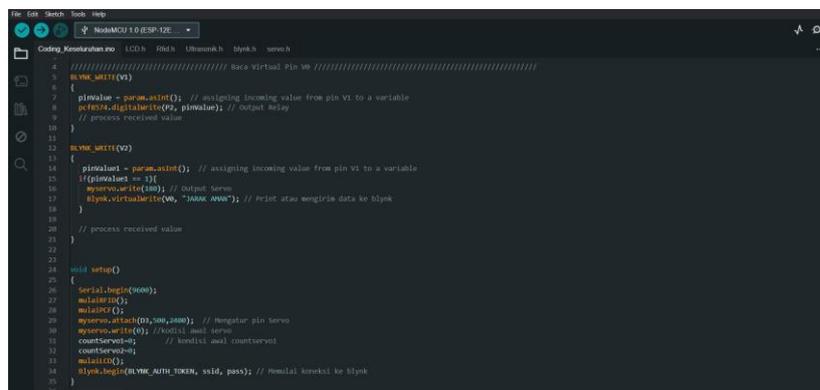


Gambar 3. 6 Perancangan Model 3D

3.4 PERANCANGAN *SOFTWARE*

Perancangan perangkat lunak berfungsi sebagai panduan dalam pengembangan program untuk mengoperasikan seluruh sistem. Dalam perancangan ini bahasa pemrograman yang digunakan adalah C++. Proses perancangan memiliki beberapa tahapan, yaitu:

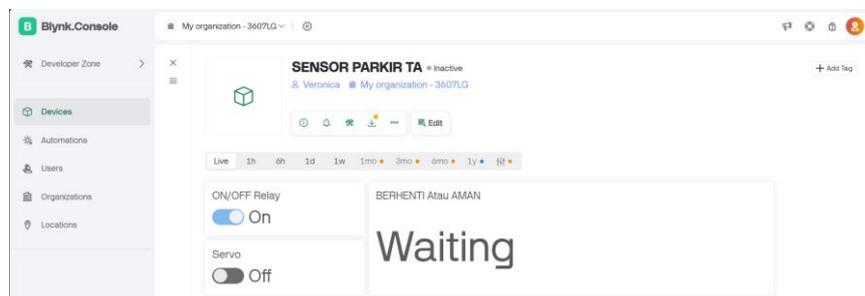
- Instalasi Pustaka.
- Konfigurasi sistem.
- Pembacaan data dari Mikrokontroler.
- Penampilan data pada aplikasi *Blynk*.



```
1 /////////////////////////////////////////////////// Baca Virtual Pin via Blynk ///////////////////////////////////
2
3 #include <Blynk/Blynk.h>
4 #include <Servo.h>
5 #include <Wire.h>
6
7 Blynk_WRITE(V1)
8 {
9   pinMode = paraw.asInt(); // assigning incoming value from pin V1 to a variable
10  digitalWrite(P2, pinMode); // output Relay
11  // process received value
12 }
13
14 Blynk_WRITE(V2)
15 {
16   pinValue1 = param.asInt(); // assigning incoming value from pin V1 to a variable
17   if (pinValue1 == 1)
18     myServo.write(180); // output servo
19   Blynk.virtualWrite(V3, "AMAN AMAN"); // print atau mengirim data ke blynk
20 }
21 // process received value
22
23
24 void setup()
25 {
26   Serial.begin(9600);
27   pinMode(P2);
28   myServo.attach(P3, 90, 210); // Mengatur pin servo
29   myServo.write(0); // posisi awal servo
30   countServo=0; // kendali awal countservo
31   countServo=0;
32   myServo.write(0);
33   Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass); // Memulai koneksi ke blynk
34 }
35
```

Gambar 3. 7 Program Perancangan Sistem

Program utama pada perangkat dalam mengatur keseluruhan operasi program, mencakup subprogram yang menjalankan fungsi-fungsi spesifikasi yang diperlukan oleh sistem. Dalam penelitian ini, aplikasi *Blynk* merupakan sistem pengirim notifikasi jarak mobil dengan sensor dan dapat mengatur nyala atau tidaknya relay dan servo.



Gambar 3. 8 Tampilan Aplikasi *Blynk*

3.5 UJI COBA ALAT DAN SISTEM

Selanjutnya dapat melakukan uji coba alat sensor parkir. Tujuan dari pengujian ini akan memberikan gambaran bagaimana setiap komponen berfungsi dan berkontribusi terhadap keseluruhan sistem parkir serta bagaimana data dari sensor dikirimkan melalui jaringan *Internet Of Things* untuk memberikan informasi parkir yang akurat dan *real-time*. Berikut ada beberapa perancangan sistem yang akan diuji oleh penulis:

1. Pengujian Sistem Pembacaan RFID *Card*.
2. Pengujian Sistem Keakuratan Sudut Motor Servo SG90.
3. Pengujian Sistem *Ultrasonic* HC-SR04.
4. Pengujian Sistem Konektivitas *Blynk* sistem.

3.5.1 Pengujian Sistem Pembacaan RFid *Card*

Pengujian ini dilakukan pada sistem keamanan parkir RFID *card* yang akan didaftarkan berjumlah 4 buah kartu dan 2 kartu tidak terdaftar. Pengujian RFID ini bertujuan untuk mengukur jarak kartu menggunakan penggaris lalu membaca id tag yang telah terdaftar dan yang belum terdaftar. Jika sudah terdaftar maka kartu memiliki akses dan jika kartu tidak terdaftar maka akses akan. Pengujian dilakukan dengan mendekatkan RFID *card* ke RFID *reader* yang telah tersedia. Lalu dilanjutkan dengan pengujian kartu yang akan dibaca oleh sensor RFID untuk mengetahui berapa jarak dari pengujian sensor RFID beserta kode program yang digunakan.

Alat Uji:

1. Laptop
2. Penggaris
3. *Software* Arduino IDE (Serial Monitor)
4. RFID
5. Kartu

```
jarak :25 cm UID tag : BA D8 F1 77  
Pesan : Akses diterima : Veronika
```

Gambar 3. 9 Pengujian *Output* RFID Akses Diterima Pada Serial Monitor

```
jarak :75 cm UID tag : 94 F8 79 71  
Pesan : Akses ditolak :
```

Gambar 3. 10 Pengujian *Output* RFID Akses Ditolak Pada Serial Monitor

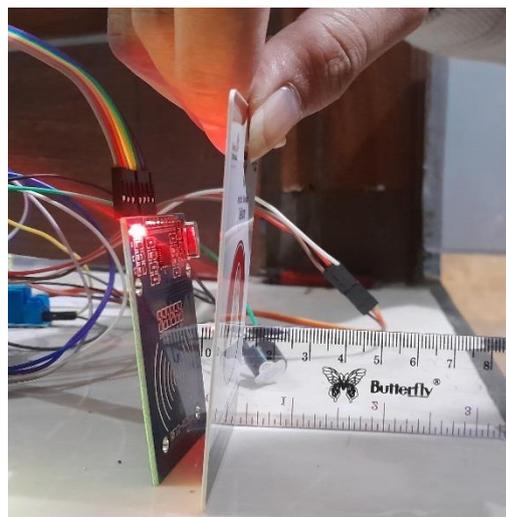
Pada pengujian dilakukan dengan 2 jenis akses pada kartu yaitu, akses diterima seperti pada gambar 3.9 dan akses ditolak seperti pada gambar 3.10.



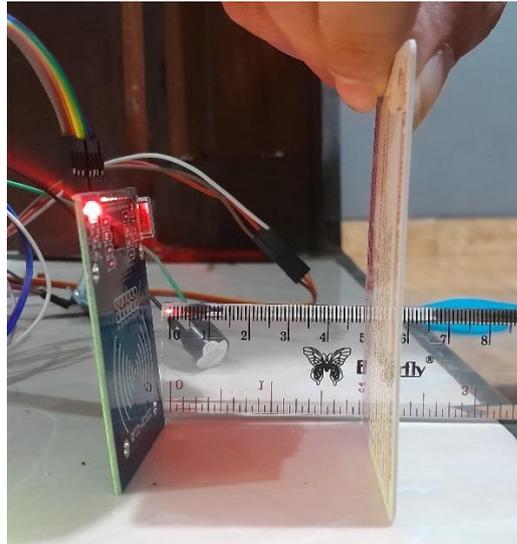
Gambar 3. 11 Pengujian Sistem RFID Card

Pada gambar 3.11 merupakan contoh gambar pengujian sistem yang akan dilakukan pada RFID. Pengujian ini hanya digunakan oleh 1 kartu dengan *code card c7 76 DF A6* yang akan di-tap melalui RFID card ke RFID reader yang telah tersedia.

Pengujian sistem selanjutnya akan dilakukan pengujian pada pembacaan RFID card dengan jarak 1 cm sampai 5 cm. Pengujian ini dilakukan agar mengetahui seberapa jarak pembacaan yang masih dapat di akses pada RFID reader.



Gambar 3. 12 Pengujian Jarak 1 cm Pada Sensor RFID dengan Kartu



Gambar 3. 13 Pengujian Jarak 5 cm Pada Sensor RFID dengan Kartu

Pengujian sistem jarak merupakan objek pengukuran jarak RFID *card* ke RFID *reader* yang dapat diketahui bahwa pembacaan kartu dimulai dari jarak 0 cm sampai 3 cm. Pada gambar 3.12 merupakan objek pengukuran RFID *Card* dengan jarak 1 cm dan pada gambar 3.13 merupakan objek pengukuran RFID *Card* dengan jarak 5 cm.



Gambar 3. 14 *Output* RFID Akses Diterima Pada LCD



Gambar 3. 15 *Output* RFID Akses Ditolak Pada LCD

Pada RFid *Card* dapat menghasilkan *output* dari LCD bahwa akses kartu diterima seperti pada gambar 3.14 kemudian terdapat tampilan akses kartu ditolak seperti pada gambar 3.15.

3.5.2 Pengujian Sistem Keakuratan Sudut Motor Servo

Pengujian sistem pada motor servo dilakukan untuk memastikan bahwa motor servo berfungsi respon terhadap sinyal kontrol dari mikrokontroler ESP8266, serta pengujian akurasi dan keandalan pergerakan servo dalam menyesuaikan posisi berdasarkan data yang diterima. Pengujian sistem keakuratan motor servo dapat diuji menggunakan alat ukur busur derajat yang mampu mengukur sudut dari 0° hingga 90°. Dalam pengujian ini, motor servo diprogram untuk bergerak ke posisi sudut tertentu, misalnya 0°, 30°, 45°, 60° dan 90°. Setiap kali servo bergerak ke sudut yang diinginkan, alat ukur busur derajat digunakan untuk mengukur dan memverifikasi posisi sudut yang dicapai oleh poros servo.

Alat Uji:

1. Laptop
2. Busur 180°
3. Aplikasi Arduino IDE (*Source Code* dan Serial Monitor)
4. Servo

```
18:48:31.073 -> Sudut Servo :19
18:48:31.107 -> Sudut Servo :18
18:48:31.140 -> Sudut Servo :17
18:48:31.140 -> Sudut Servo :16
18:48:31.174 -> Sudut Servo :15
18:48:31.174 -> Sudut Servo :14
18:48:31.207 -> Sudut Servo :13
18:48:31.240 -> Sudut Servo :12
18:48:31.240 -> Sudut Servo :11
18:48:31.273 -> Sudut Servo :10
18:48:31.273 -> Sudut Servo :9
18:48:31.342 -> Sudut Servo :8
18:48:31.342 -> Sudut Servo :7
18:48:31.373 -> Sudut Servo :6
18:48:31.373 -> Sudut Servo :5
18:48:31.373 -> Sudut Servo :4
18:48:31.407 -> Sudut Servo :3
18:48:31.407 -> Sudut Servo :2
18:48:31.440 -> Sudut Servo :1
18:48:31.440 -> Sudut Servo :0
```

Gambar 3. 16 Sudut Servo Pada Serial Monitor

Pada gambar 3.16 merupakan tampilan serial monitor keluaran sudut servo. Serial monitor hanya berguna untuk menampilkan perubahan sudut pada servo.

Pengujian berikutnya akan dilakukan pengukuran kemudian dibandingkan dengan sudut yang diprogram untuk menentukan seberapa akurat servo dalam mencapai posisi yang diinginkan. Keakuratan ini penting untuk aplikasi yang membutuhkan kontrol posisi presisi, seperti dalam robotika dan mekanisme kontrol

otomatis, memastikan bahwa motor servo dapat menyesuaikan posisi dengan tepat sesuai perintah yang diberikan.



Gambar 3. 17 Pengujian 30° Pada Sudut Servo Menggunakan Busur

Pengujian servo menggunakan busur digunakan sebagai metode untuk mengukur amplitudo perpindahan pada motor servo. Dalam program Arduino IDE, motor servo diprogram untuk melakukan uji pada posisi sudut tertentu, yaitu 0°, 30°, 45°, 60° dan 90° dengan interval waktu penundaan sebesar 10000 ms (10 detik), hanya saja akan dicantumkan satu contoh gambar pengujian, yaitu pengujian pada sudut 30° seperti pada gambar 3.17.

3.5.3 Pengujian Sistem Sensor *Ultrasonic* HC-SR04

Sensor ultrasonik HC-SR04 memiliki rentang pengukuran jarak atau ketinggian yang bervariasi antara 2 cm hingga 400 cm. Rentang tegangan masukan untuk sensor ultrasonik HC-SR04 berkisar dari 1 *Volt* hingga 5 *Volt*. Data analog yang dihasilkan oleh sensor ultrasonik HC-SR04 digunakan sebagai *input* oleh mikrokontroler, yang kemudian diproses menjadi nilai aktual jarak atau ketinggian oleh mikrokontroler tersebut. Pengujian Ultrasonik HC-SR04 dilakukan dengan menempatkan objek didepan sensor Ultrasonik HC-SR04 secara tegak lurus untuk memastikan keakuratan dan konsisten pembacaan jarak.

Alat Uji:

1. Laptop
2. Sensor Ultrasonik
3. *Software* Arduino IDE (*Source Code* dan Serial Monitor)
4. Aplikasi *Blynk*

```

18:49:04.218 -> jarak :56 cm
18:49:05.399 -> jarak :56 cm
18:49:06.526 -> jarak :56 cm
18:49:07.685 -> jarak :56 cm
18:49:08.864 -> jarak :56 cm
18:49:10.023 -> jarak :73 cm
18:49:11.143 -> jarak :56 cm
18:49:12.315 -> jarak :56 cm
18:49:13.464 -> jarak :56 cm
18:49:14.650 -> jarak :219 cm
18:49:15.789 -> jarak :219 cm
18:49:16.984 -> jarak :219 cm
18:49:18.136 -> jarak :219 cm
18:49:19.295 -> jarak :72 cm

```

Gambar 3. 18 Jarak Ultrasonik Pada Serial Monitor

Pada gambar 3.18 merupakan tampilan serial monitor pada jarak pengukuran sensor *ultrasonic* HC-SR04.

BERHENTI Atau AMAN

BERHENTI

Gambar 3. 19 *Dashboard Blynk Ultrasonic <5 cm*

BERHENTI Atau AMAN

JARAK AMAN

Gambar 3. 20 *Dashboard Blynk Ultrasonic >5 cm*

Pada *dashboard Blynk* kurang dari jarak 5 cm maka akan menampilkan *output* “BERHENTI” dan mendapatkan notifikasi peringatan dari aplikasi *Blynk* pada android seperti pada gambar 3.19. Notifikasi ini hanya didapatkan ketika jarak objek dengan sensor *ultrasonic* HC-SR04 sudah mendekati nilai jarak dibawah 5 cm. Pada jarak diatas 5 cm maka tampilan *dashboard Blynk ultrasonic* lebih dari 5 cm akan menampilkan *output* “JARAK AMAN” seperti pada gambar 3.20.

$$ERROR = \frac{\text{Selisih Pengukuran}}{\text{Pengukuran Manual}} \times 100$$

Dengan menggunakan rumus diatas maka dapat menghasilkan perhitungan berapa *error* sensor *ultrasonic* yang didapatkan.

$$\text{Rata-rata kesalahan} = \frac{\text{Total Persentasi Kesalahan}}{15}$$

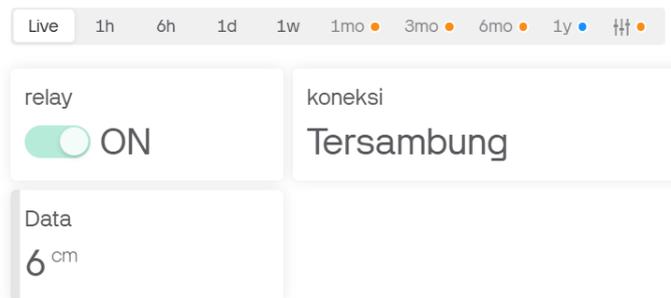
Dengan menggunakan rumus diatas maka akan dapat menghasilkan rata-rata kesalahan yang akan diuji pada sensor ultrasonik.

3.5.4 Pengujian Sistem Konektivitas *Blynk*

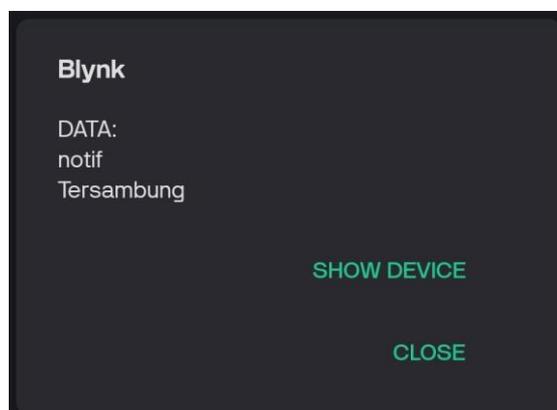
Pengujian sistem dari aplikasi *Blynk* bertujuan untuk mengevaluasi fungsi dan kinerja aplikasi dalam mengontrol dan memantau sensor parkir secara *real-time*. Selanjutnya, dilakukan uji koneksi antara aplikasi *Blynk* dan NodeMCU ESP8266 untuk memastikan transmisi data yang lancar dan *real-time*. Pengujian juga mencakup pemantauan tampilan data jarak parkir yang diterima dari sensor ultrasonik melalui aplikasi *Blynk*. *Output* akan menentukan responsivitas aplikasi dalam memberikan notifikasi jarak parkir kepada pengendara.

Alat Uji:

1. Laptop
2. *Handphone* (Sebagai notifikasi)
3. *Software Blynk*



Gambar 3. 21 Pengujian Sistem Pada Dashboard *Blynk*



Gambar 3. 22 Pengujian Sistem Pada Notifikasi Diterima Melalui *Blynk*