

**RANCANG BANGUN SISTEM MONITORING AREA PARKIR
DAN PELACAKAN POSISI KENDARAAN BERBASIS
MIKROKONTROLLER DAN COMPUTER VISION**

SKRIPSI

*Untuk Memenuhi Sebagian Persyaratan
Mencapai Gelar Sarjana Komputer*

Program Studi : Sistem Komputer
Jenjang Pendidikan : Strata 1 (S1)



Diajukan Oleh

QADRI ALFISYAHRI

20101152620032

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PUTRA INDONESIA "YPTK"**

PADANG

2024

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Qadri Alfisyahri
NoBP : 20101152620032
Fakultas : ILMU KOMPUTER
Jurusan : SISTEM KOMPUTER

Menyatakan Bahwa :

1. Sesungguhnya skripsi yang saya susun ini merupakan hasil karya tulis saya sendiri. Adapun bahagian-bahagian tertentu dalam skripsi yang saya peroleh dan hasil karya tulis orang lain, telah saya tuliskan sumbernya dengan jelas, sesuai dengan kaidah penulisan ilmiah.
2. Jika dalam pembuatan skripsi baik pembuatan program maupun skripsi secara keseluruhan terbukti dibuatkan oleh orang lain, maka saya bersedia menerima sanksi yang diberikan akademik, berupa pembatalan skripsi dan mengulang penelitian serta mengajukan judul yang baru.

Demikianlah surat pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari pihak manapun.

Padang, Februari 2024

(Qadri Alfisyahri)
20101152620032

HALAMAN PENGESAHAN PENGUJI SIDANG SKRIPSI

**RANCANG BANGUN SISTEM MONITORING AREA PARKIR
DAN PELACAKAN POSISI KENDARAAN BERBASIS
MIKROKONTROLLER DAN COMPUTER VISION**

OLEH :

QADRI ALFISYAHRI
20101152620055

PROGRAM STUDI SISTEM KOMPUTER

Skripsi ini telah dinyatakan LULUS oleh Penguji Materi Pada Sidang Program
Studi Strata 1 Ilmu Komputer Program Studi Sistem Komputer
Universitas Putra Indonesia “YPTK” Padang
Pada Hari/Tanggal : Kamis / 07 Maret 2024

TIM PENGUJI

1. **Mardhiah Masril, S.Kom., M.Kom**
NIDN : 1012108401
2. **Anip Febriko, S.Kom., M.Kom**
NIDN : 1016028203

Padang, 07 Maret 2024

Mengetahui
Dekan Fakultas Ilmu Komputer
Universitas Putra Indonesia YPTK Padang

(Prof. Dr. Yuhandri, S.Kom., M.Kom)
NIDN : 1015057301

HALAMAN PENGESAHAN LULUS SIDANG SKRIPSI

**RANCANG BANGUN SISTEM MONITORING AREA PARKIR
DAN PELACAkan POSISI KENDARAAN BERBASIS
MIKROKONTROLLER DAN COMPUTER VISION**

Yang dipersiapkan dan disusun oleh

QADRI ALFISYAHRI
20101152620032

Telah dipertahankan di depan dewan penguji
pada tanggal: 7 Maret 2024
dan dinyatakan telah lulus memenuhi syarat

Pembimbing I

Pembimbing II

(Okta Andrica Putra, S.Kom., M.Kom) **(Ondra Eka Putra, S.Kom., M.Kom)**
NIDN : 1005108604 **NIDN : 1006068701**

Padang, 7 Maret 2024
Dekan Fakultas Ilmu Komputer
Universitas Putra Indonesia “YPTK” Padang

(Prof. Dr. Yuhandri, S.Kom, M.Kom)
NIDN : 1015057301

HALAMAN PERSEMBAHAN

Dengan segala puji syukur kepada Tuhan yang Maha Esa dan atas dukungan dan doa dari orang-orang tercinta, akhirnya skripsi ini dapat dirampungkan dengan baik dan tepat pada waktunya. Oleh karena itu, dengan rasa bangga saya ucapkan rasa syukur dan terimakasih saya kepada :

1. Teristimewa kepada kedua orang tua mama dan papa yang dengan penuh kesabaran selalu memberi dukungan moril dan materil serta tidak pernah lelah selalu mengiringi dengan do'a selama perkuliahan serta proses penyusunan skripsi ini.
2. Terima kasih juga buat Abang Fris dan Kak Eby yang telah menolong dalam proses pengantaran alat skripsi ke Padang.
3. Teman-temanku, Zizah, Imam, Juan, Ilham, Aya, Kia, Ulfa, Ferdi, Jefri, Habib terima kasih atas bantuannya selama yang diberikan kepada penulis baik moril maupun materil.
4. Terima kasih kepada Rani yang telah berkontribusi banyak dalam penulisan skripsi ini, baik tenaga , waktu maupun materi kepada saya. Telah meberikan dukungan, mendengar keluh kesah, dan memberikan semangat untuk pantang menyerah.
5. Terakhir, terima kasih untuk diri sendiri karena telah mampu berjuang menjalani sejauh ini. Mampu mengendalikan diri dari berbagai tekanan diluar keadaan dan tak pernah memutuskan menyerah sesulit apapun proses penyusunan skripsi ini dengan menyelesaikan sebaik dan semaksimal mungkin.

ABSTRACT

Thesis Title : **Design and Development of a Parking Area Monitoring System and Vehicle Position Tracking Based on Microcontrollers and Computer Vision**
Student Name : **Qadri Alfisyahri**
Student Number : **20101152620032**
Study Program : **Computer Engineering**
Study Program : **Strata 1 (S1)**
Advisors : **1. Okta Andrica Putra, S.Kom., M.Kom**
2. Ondra Eka Putra, S.Kom., M.Kom

Currently, parking management still relies on manual methods, where an attendant must individually check each available parking slot. Additionally, large multi-level parking lots require vehicle users to circle around in search of empty parking spots. Once a vehicle is parked, users may forget its location or have difficulty finding it again. A prototype parking slot monitoring system has been developed using HC-SR04 ultrasonic sensors and vehicle license plate recognition with ESP32-CAM. The system is designed to facilitate drivers in obtaining parking slot information in multi-level buildings and tracking the position of parked vehicles in connected parking areas using the ESP32-CAM and Arduino Mega 2560 modules. Each slot is equipped with an ultrasonic sensor that detects distance to determine if a vehicle is present in the slot. The system includes license plate character recognition using Optical Character Recognition (OCR) methods, and the vehicle plate data is then stored in a database.

Keywords : Arduino Mega 2560, ESP32-Cam, Computer Vision, OCR. Monitoring

ABSTRAK

Judul Skripsi : **Rancang Bangun Sistem Monitoring Area Parkir Dan Pelacakan Posisi Kendaraan Berbasis Mikrokontroller Dan Computer Vision**
Nama : **Qadri Alfisyahri**
No BP : **20101152620032**
Program Studi : **Sistem Komputer**
Jenjang Pendidikan : **Strata 1 (S1)**
Pembimbing : **1. Okta Andrica Putra, S.Kom., M.Kom**
2. Ondra Eka Putra, S.Kom., M.Kom

Saat ini, manajemen parkir masih mengandalkan metode manual, di mana seorang petugas harus memeriksa secara individual setiap slot parkir yang tersisa. Selain itu, lahan parkir bertingkat yang besar membuat pengguna kendaraan harus berputar-putar mencari tempat parkir yang kosong. Setelah kendaraan diparkir, pengguna mungkin lupa lokasi parkirnya atau mengalami kesulitan dalam menemukan kembali kendaraannya. Tujuan penelitian ini menghasilkan purwarupa sistem pemantauan slot parkir menggunakan sensor ultrasonik HC-SR04 dan pengenalan plat nomor kendaraan dengan ESP32-CAM. Sistem dibuat agar dapat memudahkan pengendara dalam mengetahui informasi slot parkir pada gedung bertingkat dan melacak posisi kendaraan yang parkir pada area parkir yang terhubung dengan modul ESP32-CAM dan Arduino Mega 2560. Pada tiap slot terdapat sensor ultrasonik yang mendeteksi jarak sehingga dapat mengetahui apakah terdapat kendaraan pada slot. Sistem dilengkapi pengenalan karakter plat nomor kendaraan yang menggunakan metode Optical Character Recognition (OCR) selanjutnya data plat kendaraan disimpan pada database.

Kata kunci : Arduino Mega 2560, ESP32-Cam, Computer Vision, OCR. Monitoring

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah S.W.T yang telah memberikan rahmat dan karunianya, sehingga bisa menyelesaikan penulisan skripsi penulis dengan judul **“RANCANG BANGUN SISTEM MONITORING AREA PARKIR DAN PELACAKAN POSISI KENDARAAN BERBASIS MIKROKONTROLLER DAN COMPUTER VISION”**. Penulisan laporan ini bertujuan dalam rangka untuk memperoleh gelar Sarjana Komputer pada Fakultas Ilmu Komputer Universita Putra Indonesia “YPTK” Padang.

Penulisan laporan ini diwujudkan dalam bentuk suatu perancangan alat, yang tidak lepas dari kata kekurangan, keterbatasan, dan jauh dari kata sempurna, baik dari segi penulisan, bahasa, dan alatnya. Hal ini dikarenakan oleh ilmu, pengetahuan dan pengalaman penulis miliki masih kurang serta keterbatasan fasilitas yang disediakan. Oleh karena itu penulis berharap adanya kritik saran yang mendukung agar dapat diperbaiki kembali dan sempurna hendaknya.

Selanjutnya didalam penulisan laporan ini penulis sadar bahwa keberhasilan penulis dalam menyelesaikan skripsi ini berkat dari dorongan dan bimbingan dari pembimbing, orang tua, teman, dan berbagai pihak, oleh karena itu penulis ingin mengucapkan rasa terima kasih, khususnya kepada:

1. Ibu **Dr. Hj. Zerni Melmusi, MM., Ak., CA** selaku Ketua Yayasan Perguruan Tinggi Komputer “YPTK” Padang.
2. Bapak **Prof. Dr. Sarjon Defit, S.Kom., M.Sc** selaku Rektor Universitas Putra Indonesia “YPTK” Padang.

3. Bapak **Prof. Dr. Yuhandri, S.Kom., M.Kom** selaku Dekan Fakultas Ilmu Komputer Universitas Putra Indonesia “YPTK” Padang.
4. Ibu **Retno Devita, S.Kom., M.Kom** selaku Ketua Jurusan Sistem Komputer Universitas Putra Indonesia “YPTK” Padang.
5. Bapak **Okta Andrica Putra S.Kom, M.Kom** selaku Pembimbing I yang telah membimbing dan sangat banyak memberi masukan kepada penulis.
6. Bapak **Ondra Eka Putra S.Kom, M.Kom** selaku Pembimbing II yang telah membimbing dan sangat banyak memberi masukan kepada penulis.
7. Bapak dan Ibu Dosen Universitas Putra Indonesia “YPTK” Padang.

Akhir kata, penulis berharap semoga kebaikan dan amal yang mereka berikan kepada penulis mendapatkan balasan yang berlipat ganda hendaknya dari Allah S.W.T, *Aamiiin ya rabbal ‘Alamiin.*

Padang, Februari 2024

Qadri Alfisyahri

DAFTAR ISI

LEMBAR PERNYATAAN	ii
HALAMAN PENGESAHAN SKRIPSI.....	iii
HALAMAN PENGESAHAN PENGUJI SIDANG SKRIPSI.....	iv
HALAMAN PENGESAHAN LULUS SIDANG SKRIPSI.....	v
ABSTRACT	vi
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Hipotesa.....	4
1.5 Tujuan Penelitian.....	5
1.6 Manfaat Penelitian.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Konsep Dasar Sistem.....	7
2.1.1 Karakteristik Sistem	8
2.1.2 Alur Pengembangan Sistem	9
2.2 Alat-Alat Bantu Dalam Perancangan Sistem Dan Logika Program.....	11
2.2.1 Contex Diagram	11
2.2.2 Diagram Flow Data	13
2.2.3 Flowchart	16
2.3 Sistem Kontrol.....	19
2.3.1 Sistem Loop Terbuka (Open-Loop Control System)	19
2.3.2 Sistem Loop Tetutup (Closed-Loop Control System).....	20
2.4 Deteksi Plat Kendaraan	20
2.4.1 Python	21
2.4.2 Tesseract OCR.....	21

2.4.3	Visual Studio Code (VS Code)	23
2.5	Komponen Utama.....	24
2.5.1	Arduino Mega 2560	24
2.5.1	Esp32-Cam.....	26
2.5.2	Servo SG90	27
2.5.3	Sensor ultrasonic HC-SR04	28
2.5.4	Sensor Infrared	29
2.5.5	LCD.....	31
2.5.6	Push Button	32
2.5.7	DFPlayer Mini.....	33
2.5.8	Speaker	34
2.6	Komponen pendukung	35
2.6.1	Resistor.....	35
2.6.2	IC Regulator	37
2.6.3	Power supply	37
2.6.4	Kabel Jumper	38
2.7	Arduino IDE	39
2.7.1	Pengenalan Software Arduino IDE	40
2.8	Bahasa Pemograman Arduino	46
2.8.1	Struktur.....	46
2.8.2	Variabel	47
2.8.3	Tipe data	47
2.8.4	Operator.....	48
2.7.1	Operator Aritmatika.....	48
2.7.2	Operator Perbandingan.....	49
2.7.3	Operator Logika	49
2.8.5	Kondisi	50
2.7.5.1	Kondisi If.....	50
2.7.5.2	Kondisi <i>If-Else</i>	51
2.7.5.3	Kondisi If Else If Else.....	51
2.7.5.4	Kondisi Switch Case.....	52
2.8.6	Perulangan (<i>Loop</i>)	53

2.7.6.1	Perulangan For	53
2.7.6.1	Perulangan <i>While</i>	53
2.7.6.1	Perulangan <i>Do-While</i>	54
BAB III	METODOLOGI PENELITIAN	55
3.1	Kerangka Kerja Penelitian	55
3.2	Uraian Kerangka Kerja Penelitian	55
3.2.1	Identifikasi Masalah	56
3.2.2	Pengumpulan Data	56
3.2.2.1	Waktu Penelitian	57
3.2.2.2	Metode Penelitian	57
3.2.3	Analisis sistem	59
3.2.4	Perancangan Sistem	59
3.2.5	Pembuatan Alat	62
3.2.6	Pengujian alat	62
3.2.7	Implementasi sistem	63
BAB IV	ANALISA DAN HASIL	64
4.1	Desain Sistem Secara Umum	64
4.1.1	Context Diagram	64
4.1.2	Data Flow Diagram (DFD)	68
4.1.3	Block diagram	72
4.2	Prinsip Kerja Sistem	74
4.3	Rancangan Fisik Sistem	75
4.4	Desain Sistem Terperinci	81
4.4.1	Desain Rangkaian Arduino Mega 2560	81
4.4.2	Desain Rangkaian Ultrasonic	82
4.4.3	Desain Rangkaian RTC	85
4.4.4	Desain Rangkaian Push Button	86
4.4.5	Desain rangkaian motor servo	87
4.4.6	Desain Rangkaian LCD	88
4.4.7	Desain Rangkaian Sensor Infrared	89
4.4.8	Desain Rangkaian ESP32-Cam	90
4.4.9	Desain Rangkaian DFPlayer	91

4.5	Rancangan Modul Program	92
4.5.1	Flowchart	92
4.5.2	Modul Program	101
4.5.2.1	Program Arduino Mega 2560	101
4.5.2.2	Program Ultrasonic 9 Buah.....	105
4.5.2.3	Program RTC DS3231	106
4.5.2.4	Program Push Button	107
4.5.2.5	Program Servo SG90	108
4.5.2.6	Program LCD 16x2 i2c.....	108
4.5.2.7	Program Infrared Obstacle.....	109
4.5.2.8	Program ESP32-Cam.....	109
4.5.2.9	Program DFPlayer Mini Mp3	113
4.5.2.10	Program LCD 2 dan 3	114
BAB V PENGUJIAN SISTEM		116
5.1	Pengujian Sistem Permodul.....	116
5.1.1	Pengujian Mikrokontroler Dan Software Arduino IDE	116
5.1.2	Pengujian Sensor Ultrasonik	118
5.1.3	Pengujian Push Button	120
5.1.4	Pengujian Infrared Obstacle	120
5.1.5	Pengujian ESP32-Cam	121
5.2	Pengujian Sistem Keseluruhan	122
BAB VI PENUTUP		129
6.1	Kesimpulan.....	129
6.2	Saran.....	130
DAFTAR PUSTAKA.....		147
LAMPIRAN.....		153

DAFTAR TABEL

Tabel 2.1 Simbol Contex Diagram	12
Tabel 2.2 Simbol DFD	15
Tabel 2.3 Simbol Flowchart.....	17
Tabel 2.4 Sperifikasi Arduino Mega 2560	25
Tabel 2.5 Kategori Dan Nama Pin Arduino Mega 2560	25
Tabel 2.6 Warna Gelang Resistor	36
Tabel 2.7 Jenis Tipe Data.....	48
Tabel 2.8 Operator Aritmatika	49
Tabel 2.9 Operator Pemanding.....	49
Tabel 2.10 Operator Logika	50
Tabel 3.1 Jadwal Penelitian	57
Tabel 3.2 Hardware dan Software	59
Tabel 5.1 Pengujian Jarak Ultrasonik.....	118
Tabel 5.2 Pengujian Kesesuaian Parkir	119
Tabel 5.3 Pengujian Push Button.....	120
Tabel 5.4 Pengujian IR Obstacle.....	120
Tabel 5.5 Pengujian IR Obstacle Jarak Terdeteksi.....	121
Tabel 5.6 Pengujian ESP32-Cam	121

DAFTAR GAMBAR

Gambar 2.1 Tahapan Model Waterfall	11
Gambar 2.2 Sistem Loop Terbuka.....	20
Gambar 2.3 Sistem Loop Tertutup	20
Gambar 2.4 Arduino Mega 2560.....	24
Gambar 2.5 ESP32-Cam	27
Gambar 2.6 Motor Servo SG90	28
Gambar 2.7 Sensor Ultrasonik HC-SR04	29
Gambar 2.8 Sensor Infrared	30
Gambar 2.9 LCD 16x2	32
Gambar 2.10 Push Button	33
Gambar 2.11 DFPlayer Mini	34
Gambar 2.12 Speaker	35
Gambar 2.13 Resistor	36
Gambar 2.14 IC Regulator	37
Gambar 2.15 Power Supply.....	38
Gambar 2.16 Kabel Jumper	39
Gambar 2.17 Arduino IDE	40
Gambar 2.18 Struktur dalam Bahasa Arduino	47
Gambar 3.1 Kerangka Kera Penelitian	55
Gambar 4.1 Context Diagram Sistem Parkir	65
Gambar 4.2 DFD Sistem Parkir	69
Gambar 4.3 Blok Diagram Sistem Parkir.....	73
Gambar 4.4 Desain Rancangan Fisik.....	80
Gambar 4.5 Desain Rangkaian Arduino Mega 2560	81
Gambar 4.6 Desain Rangkaian Sensor Ultrasonik HC-SR04.....	82
Gambar 4.7 Desain Rangkaian RTC DS3231.....	85
Gambar 4.8 Desain Rangkaian Push Button	86
Gambar 4.9 Desain Rangkaian Motor Servo Sg90	87
Gambar 4.10 Desain Rangkaian 3 LCD 16x2 I2C	88

Gambar 4.11 Desain Rangkaian Sensor Infrared Obstacle	89
Gambar 4.12 Desain Rangkaian ESP32-Cam	90
Gambar 4.13 Desain Rangkaian DFPlayer Mini Mp3.....	91
Gambar 4.14 Flowchat Arduino Mega 2560.....	98
Gambar 4.15 Flowchat ESP32-Cam.....	99
Gambar 4.16 Flowchat Arduino Uno R3	100
Gambar 5.1 Tampilan Awal Arduino IDE	116
Gambar 5.2 Pemilihan Tipe Board Arduino	117
Gambar 5.3 Tampilan program berhasil <i>upload</i>	117
Gambar 5.4 Tampilan LCD 1 informasi ketersediaan slot	123
Gambar 5.5 Tampilan portal terbuka masuk.....	123
Gambar 5.6 Tampilan database	124
Gambar 5.7 Tampilan Data Plat	124
Gambar 5.8 Tampilan portal tertutup masuk	125
Gambar 5.10 Tampilan Slot Diisi.....	125
Gambar 5.10 Tampilan portal terbuka keluar	126
Gambar 5.11 Tampilan Portal Tertutup.....	127
Gambar 5.12 Website Menampilkan Data kendaraan	127
Gambar 5.13 Website Pencarian Data Kendaraan di Gedung Parkir.....	128

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan pesat teknologi telah menciptakan transformasi signifikan dalam berbagai aspek kehidupan sehari-hari termasuk sistem parkir di kota besar. Penerapan teknologi dalam pemantauan area parkir menjadi suatu kebutuhan yang mendesak mengingat tingginya mobilitas penduduk perkotaan. Teknologi monitoring area parkir modern dapat memberikan solusi efektif dalam manajemen parkir yang efisien, memberikan informasi *real-time* tentang ketersediaan tempat parkir, serta meningkatkan pengalaman pengguna secara keseluruhan (Wisoso et al., 2022).

Fasilitas parkir adalah sebuah area atau tempat yang disediakan khusus untuk mengatur dan menyimpan kendaraan roda dua atau roda empat sementara saat tidak digunakan (Hidayat et al., 2022). Fasilitas parkir ini dirancang untuk memudahkan pemilik kendaraan dalam mencari tempat yang aman dan teratur untuk memarkirkan kendaraannya baik dalam jangka waktu yang singkat maupun panjang. Fasilitas parkir kendaraan umumnya dapat ditemukan di berbagai lokasi seperti gedung perkantoran, pusat pembelajaran, bandara, stasiun, rumah sakit, hotel, pusat hiburan, dan area publik lainnya.

Pertumbuhan jumlah kendaraan di perkotaan telah menyebabkan sebuah masalah serius dalam manajemen tempat parkir sementara lahan parkir yang tersedia sedikit. Saat ini manajemen tempat parkir masih menggunakan cara yang

manual, dimana seorang petugas harus memeriksa secara satu persatu slot parkir yang tersisa. Hal ini akan membutuhkan waktu karena harus memeriksa seluruh slot yang ada dan juga petugas yang lebih banyak. Selain itu lahan parkir yang besar dan bertingkat mengharuskan pengguna kendaraan untuk berputar-putar mencari tempat parkir yang kosong (Widodo & Almasri, 2021). Kemudian, setelah kendaraan terparkir pun pengguna bisa saja lupa dengan posisi parkir atau kesulitan untuk menemukan posisi kendaraan mereka yang telah terparkir sebelumnya.

Salah satu penelitian yang relevan dilakukan oleh Auliya, Khairina et al. (2023) yang berjudul Sistem Pemantauan Slot Parkir Menggunakan Sensor Ultrasonik JSN-SR04T dan Pengenalan Plat Nomor Kendaraan dengan ESP32-CAM. Penelitian ini mengembangkan suatu sistem inovatif dengan memanfaatkan teknologi sensor ultrasonik JSN-SR04T guna mendeteksi keberadaan kendaraan di setiap slot parkir. Lebih lanjut, penelitian sebelumnya juga mencoba mengintegrasikan pengenalan plat nomor kendaraan, namun belum optimal dalam memberikan informasi yang akurat dan cepat.

Bedasarkan masalah diatas dan penelitian tersebut, maka diperlukan sebuah teknologi yang bisa menghitung dan memeriksa slot parkir yang kosong tanpa harus diperiksa secara manual oleh petugas parkir. Sistem ini nantinya dapat memberitahu secara langsung kepada petugas di gerbang masuk dan pemilik kendaraan dimana lokasi parkir yang tersedia dan memudahkan pengemudi untuk mengisi slot parkir yang tersedia tersebut tanpa harus berkeliling mengitari lahan parkir untuk mencari yang kosong. Sistem ini nantinya dilengkapi dengan sebuah teknologi *computer vision* yang akan berfungsi mendata plat kendaraan yang masuk

pada area parkir dimana kemudian data itu digunakan untuk mencari posisi kendaraan yang telah terparkir, pemilik kendaraan tidak harus berkeliling dan kesulitan untuk mengingat dimana mereka memarkirkan kendaraan mereka jika mereka lupa dengan posisi kendaraan mereka. Maka peneliti membuat sebuah alat **“RANCANG BANGUN SISTEM MONITORING AREA PARKIR DAN PELACAKAN POSISI KENDARAAN BERBASIS MIKROKONTROLLER DAN COMPUTER VISION”**.

1.2 Rumusan Masalah

Berdasarkan uraian diatas, rumusan masalah dalam penelitian yang akan dilakukan adalah sebagai berikut:

1. Bagaimana sistem monitoring area parkir dapat mengatasi pengguna kendaraan untuk mengetahui ketersediaan ruang parkir pada gedung bertingkat ?
2. Bagaimana sistem navigasi area parkir dapat mengatasi kesulitan pengguna kendaraan dalam mencari slot parkir yang kosong di area parkir gedung bertingkat ?
3. Bagaimana sistem pelacakan posisi parkir kendaraan pada area parkir dapat mengatasi masalah pengguna kendaraan yang lupa posisi parkir di area parkir gedung bertingkat ?

1.3 Batasan Masalah

Batasan masalah digunakan agar pembahasan dalam laporan kerja praktik ini tidak terlalu meluas, oleh karena itu perlu nya ada pembatasan masalah. Adapun batasan masalah yang dikemukakan adalah sebagai berikut :

1. Pada bagian sistem pencarian kendaraannya menggunakan nomoar plat kendaraan yang terdaftar awal masuk area parkir.
2. Pada *prototype* ini hanya menggunakan satu komputer untuk menampung data kendaraan dan untuk mencari posisi kendaraan.
3. Sistem parkir ini dirancang untuk pemakaian dalam ruang atau gedung bukan untuk lapangan.
4. Alat yang dirancang ini hanya menggunakan sistem untuk menghitung slot parkir dan mencari posisi letak kendaraan yang terparkir, tidak terhubung dengan administrasi keuangan.

1.4 Hipotesa

Berdasarkan pada perumusan masalah diatas, maka dapat diambil beberapa hipotesa yaitu :

1. Diharapkan sistem monitoring area parkir dapat mengatasi pengguna kendaraan untuk mengetahui ketersediaan ruang parkir pada gedung bertingkat.
2. Diharapkan sistem navigasi area parkir dengan memanfaatkan teknologi sensor pengguna kendaraan dapat dengan mudah menemukan slot parkir yang

tersedia melalui panduan waktu nyata. Hal ini dapat meningkatkan efisiensi pencarian parkir dan mengurangi frustrasi pengguna.

3. Diharapkan dengan mengimplementasikan sistem pelacakan posisi parkir kendaraan yang terintegrasi dengan sebuah website, pengguna dapat dengan mudah menemukan kembali posisi parkir mereka, termasuk informasi tentang lantai dan slot parkir yang digunakan.

1.5 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam pembuatan alat ini adalah sebagai berikut :

1. Membuat sebuah sistem monitoring area parkir yang dapat menghitung jumlah sisa parkir yang ada.
2. Membuat sebuah sistem navigasi pada area parkir yang dapat memberikan informasi mengenai keberadaan slot parkir yang masih kosong.
3. Membuat sebuah sistem pelacakan posisi kendaraan yang dapat mencari posisi parkir kendaraan yang tersedia dengan mudah.

1.6 Manfaat Penelitian

Adapun manfaat yang diharapkan penelitian dalam perancangan sistem ini adalah sebagai berikut :

A. Bagi Penulis

1. Menambah pengetahuan penulis di bidang elektronika, *computer vision* dan *system control*.

2. Menjadi referensi yang dapat dimanfaatkan untuk penelitian lebih lanjut dalam bidang computer vision.
3. Membantu penulis dalam mengalisis dan berpikir kritis terhadap sebuah permasalahan melalui penelitian yang dilakukan.

B. Bagi Program Studi

1. Menambah referensi dalam literature bagi mahasiswa yang berhubungan dengan *webcam*, *computer vision*, sensor IR obsctale, sensor ultrasonic, DFPlayer, dan Arduino Mega 2560.
2. Menjadikan referensi untuk lebih berkembangnya pemanfaatan ilmu dan teknologi yang ada serta dapat menambah bahan kepustakaan ilmu dan teknologi.
3. Menambah aplikasi pada labor sistem computer.

C. Bagi Masyarakat

1. Pengguna tempat parkir menjadi lebih nyaman dan tidak pusing untuk mencari tempat parkir yang masih kosong.
2. Pengguna parkir yang lupa posisi kendaraannya mereka dapat mencari posisi kendaraan mereka nantinya dengan mudah.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Menurut Kamus Besar Bahasa Indonesia (KBBI) sistem adalah suatu perangkat unsur yang saling berkaitan dengan teratur hingga menciptakan suatu totalitas. Sistem juga didefinisikan sebagai gabungan atas elemen yang saling berkaitan dan bekerja sama agar dapat mengoperasikan *input* (masukan) yang diarahkan kepada *system* tersebut untuk diolah sehingga menghasilkan *output* (keluaran) yang diinginkan. Sistem harus terdiri atas beberapa komponen atau elemen yang saling berkaitan sehingga menghasilkan satu kesatuan yang utuh (Kustina et al., 2022).

Secara sederhana, sistem dapat dipahami sebagai sekumpulan suatu unsur, komponen dan variable-variabel yang teratur, saling berinteraksi dan saling berkaitan satu dengan yang lainnya, pada dasarnya sistem merupakan sekumpulan unsur yang berkaitan erat satu dengan yang lainnya untuk mencapai tujuan tertentu, bentuk umum yang terdapat dalam suatu sistem adalah masukan, proses, dan keluaran (Mualim, 2021).

Berdasarkan definisi di atas dapat disimpulkan bahwa suatu sistem memiliki unsur-unsur yang mewakilinya yaitu *input* (masukan), *processing* (pengolahan) dan *output* (keluaran). Fungsi dari sistem adalah sebagai penyatuan atau pengintegrasian seluruh unsur yang ada dalam suatu ruang lingkup, dimana komponen-komponennya tidak dapat berdiri sendiri.

2.1.1 Karakteristik Sistem

Sebuah sistem pasti mempunyai sebuah karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut biasanya dikatakan sebagai sebuah sistem. (Mualim, 2021) menjelaskan bahwa karakteristik dari sebuah sistem :

1. Komponen sistem (*components*)

Suatu sistem terdiri atas beberapa komponen dimana mereka melakukan interaksi dan bekerja sama kemudian menghasilkan satu kesatuan untuk mencapai suatu tujuan.

2. Batasan sistem (*boundary*)

Batasan sistem adalah wilayah yang membatasi atau menghalangi antara suatu sistem dengan sistem yang lain atau membatasi dengan daerah luarnya.

Batasan sistem menyiratkan bagaimana ruang lingkup dari sistem tersebut.

3. Lingkungan luar sistem (*enviromtent*)

Adalah segala hal yang berada di luar lingkup atau batasan yang memberi dampak pada operasi sistem itu sendiri.

4. Penghubung sistem (*interface*)

Yaitu sebuah media yang mengaitkan sistem dengan subsistem lain agar bisa berinteraksi dan dapat menghasilkan suatu kesatuan.

5. Masukan sistem (*input*)

Merupakan suatu tenaga yang dimasukkan kedalam sistem, tenaga ini dapat berbentuk terdapat sinyal (*signal input*) dan perawatan (*maintenance input*) yang dimasukkan supaya sistem dapat beroperasi kembali.

6. Keluaran sistem (*output*)

Merupakan olahan dari hasil energi yang kemudian dikelompokkan menjadi *output* yang bermanfaat untuk subsistem lainnya.

7. Pengolahan sistem (*process*)

Pengolah sistem merupakan suatu proses didalam sistem yang dapat mengubah *input* menjadi *output*.

8. Sasaran sistem (*objective*)

Suatu tujuan atau sasaran pasti terdapat didalam suatu sistem, hal ini dikarenakan sasaran dapat menjadi penentu untuk *input* yang diperlukan sistem dan *output* yang akan dihasilkan.

2.1.2 Alur Pengembangan Sistem

Sistem yang mempunyai sebuah alur pengembangan atau siklus hidup merupakan sebuah proses evolusioner yang di ikuti dalam menerapkan sistem. Siklus ini sering kali dikaitkan dengan pendekatan air terjun bagi pengembangan dan penggunaan sistem. Menggunakan sistem air terjun akan memudahkan dalam menganalisis masalah dan menyusun sistem sesuai dengan kebutuhan untuk menyelesaikan permasalahan tersebut (Mallisza et al., 2022). Pada alur air terjun ini memiliki beberapa tahapan sebagai berikut:

1. *Requirements analysis and definition*

Service system, kendala, dan tujuan ditetapkan oleh hasil dari konsultasi dengan user setelah itu didefinisikan secara detail dan berfungsi sebagai spesifikasi dari sebuah sistem.

2. *System and software design*

Tahapan ini membagi kebutuhan-kebutuhan sebuah sistem baik itu di bagian hardware maupun software dengan membentuk sebuah arsitektur sistem secara keseluruhan. Perancangan software melibatkan proses identifikasi dan penggambaran abstraksi sistem dasar software dan hubungannya.

3. *Implementasi and unit testing*

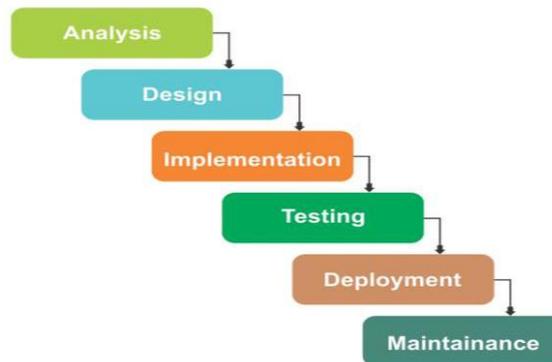
Tahapan ini merancang sebuah software untuk direalisasikan sebagai serangkaian program. dan *testing system* melibatkan verifikasi pada setiap unit pada *system* untuk memenuhi spesifikasi dari software yang dibuat.

4. *Integration and system testing*

Pada tahap ini semua unit yang sudah dibuat dan ditesting akan digabungkan menjadi sebuah *system* lengkap untuk memastikan apakah semuanya sudah sesuai dengan spesifikasi dari software yang dibuat setelah pengujian software akan dikirimkan ke user.

5. *Operation and maintenance*

Tahapan ini merupakan tahapan yang sangat panjang dimana *system* yang telah dibuat di pakai secara nyata dan dilakukan *maintenance* untuk mencari sebuah kesalahan yang tidak terdeteksi pada tahap-tahapan sebelumnya dan untuk meningkatkan implementasi dari setiap unit sistem dan layanan sistem sebagai kebutuhan baru.



Sumber : Mallisza, Danyl et al. Implementasi Model Waterfall Dalam Perancangan Sistem Surat Perintah Perjalanan Dinas Berbasis Website Dengan Metode SDLC. 2022.

Gambar 2.1 Tahapan Model Waterfall

2.2 Alat-Alat Bantu Dalam Perancangan Sistem Dan Logika Program

Proses penganalisaan terhadap suatu sistem dilakukan dengan aturan dasar yang mendefinisikan secara menyeluruh sistem yang akan dirancang. Hal ini mengandung arti bahwa harus ada gambaran yang jelas mengenai ruang lingkup tentang sistem yang dibahas. Media yang digunakan menggambarkan sistem tersebut adalah *Context Diagram*, *Data Flow Diagram (DFD)*, dan *Flowchart*.

2.2.1 Context Diagram

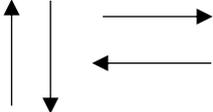
Diagram konteks merupakan diagram yang terdiri atas suatu proses dan menampilkan ruang lingkup dari suatu sistem. Diagram konteks menjadi level tertinggi dari *Data Flow Diagram (DFD)* yang menampilkan seluruh masukan ke dalam sistem atau keluaran dari sistem yang kemudian menampilkan keseluruhan

sistem. Diagram ini berbentuk sederhana dan juga tidak menampilkan data yang tersimpan (Ummah et al., 2019).

Diagram konteks menggunakan tiga buah simbol yaitu : simbol untuk melambangkan *external entity*, simbol untuk melambangkan *data flow* dan simbol untuk melambangkan *process*. Diagram konteks hanya boleh melakukan satu proses tidak boleh lebih dan pada diagram konteks tidak digambarkan *data store*. Proses yang terjadi tidak diberi nomor pada diagram konteks (Sutanti et al., 2020). Simbol yang digunakan dalam pembuatan contex diagram dapat dilihat pada Tabel 2.1.

Tabel 2.1 Simbol Contex Diagram

No	Simbol	Nama	Keterangan
1.		<i>“external entity”</i>	Kesatuan luar system yang dapat berupa sistem lain yang berada diluar lingkungan luarnya yang akan memberikan masukan (<i>input</i>) atau menerima keluaran (<i>output</i>) system.
2.		<i>“process”</i>	Kegiatan yang dilakukan oleh computer dari suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.

No	Simbol	Nama	Keterangan
3.		"data flow"	Arus data mengalir diantara proses, disimpan data dan kesatuan. Arus data menunjukkan arah dari yang masuk ke dalam proses system.

Sumber : Sutanti, Asih et al. *Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur*. 2020.

2.2.2 Diagram Flow Data

Diagram Flow Data (DFD) atau disebut juga dengan Diagram Alir Data (DAD) menurut Sukanto dan Shalahuddin merupakan representasi grafik yang menampakan pergerakan informasi dan perubahannya yang diterapkan sebagai data yang menata dari input hingga output. Diagram Flow Data tidak cocok untuk menampilkan sistem yang memakai pemograman dengan berpusat pada objek (Mardiyati et al., 2022). DFD adalah salah satu jaringan yang menunjukkan sistem manualisasi atau komputersasi dapat juga berupa gabungan kedua sistem tersebut. Dimana pendeskripsiannya menggunakan susunan yang berbentuk kumpulan komponen sistem yang saling berkaitan (Budiman et al., 2021).

Terdapat empat komponen dasar di dalam DAD yang akan menampilkan bagaimana aliran data dalam suatu sistem. Bentuk komponen tersebut yaitu entitas, proses, media penyimpanan dan arus data (Haviluddin, 2021).

1. Entitas (*Entity*)

Entitas adalah suatu tujuan data atau sumber data. Tujuan atau sumber ini terbagi menjadi garis panah atau keterangan yang berhubungan. Makna lain dari entitas adalah terminator, terminator sendiri terbagi atas dua jenis yaitu: terminator tujuan yaitu terminator yang berfungsi sebagai tujuan sebuah data atau suatu informasi sistem dan terminator sumber yaitu terminator yang berfungsi sebagai sumber.

2. Proses (*Process*)

Proses merupakan perubahan terhadap data yang menunjukkan bagian dari sistem yang kemudian merubah masukan menjadi keluaran. Terdapat empat peluang kejadian dalam suatu proses berkaitan dengan masukan dan keluaran yaitu: satu masukan dapat menghasilkan satu output, satu masukan menghasilkan banyak keluaran, banyak masukan menghasilkan satu keluaran dan banyak masukan menghasilkan banyak keluaran.

3. Media Penyimpanan (*Data Store*)

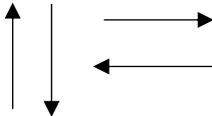
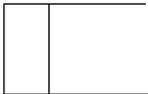
Media penyimpanan digunakan untuk menciptakan model sekumpulan paket data kemudian dinamakan menggunakan kata benda jamak contohnya adalah buku. Media penyimpanan pada umumnya berhubungan dengan penyimpanan - penyimpanan berbentuk file atau database yang berhubungan menggunakan komputerisasi sebagai penyimpanannya contohnya adalah file disket, hariddisk, pita magnetic.

4. Arus data (*Data Flow*)

Arus data adalah gabungan elemen data yang secara logis berkaitan kemudian berpindah dari satu titik ke titik atau dari proses ke proses yang lain. Arus data berbentuk seperti anak panah yang mengarah menuju ke dalam dan keluar dari suatu proses, fungsi dari arus data ialah untuk menjelaskan perpindahan data dari suatu bagian sistem ke bagian sistem lainnya.

Simbol yang digunakan dalam pembuatan Data Flow Diagram dapat dilihat pada Tabel 2.2.

Tabel 2.2 Simbol DFD

No	Simbol	Nama	Keterangan
1.		<i>“external entity”</i>	Kesatuan luar system yang dapat berupa sistem lain yang berada diluar lingkungannya yang akan memberikan masukan (<i>input</i>) atau menerima keluaran (<i>output</i>) system.
2.		<i>“process”</i>	Kegiatan yang dilakukan oleh computer dari suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.
3.		<i>“data flow”</i>	Arus data mengalir diantara proses, disimpan data dan kesatuan. Arus data menunjukkan arah dari yang masuk ke dalam proses system.
4.		<i>“data store”</i>	Tempat penyimpanan file file yang dibutuhkan.

Sumber : Sutanti, Asih et al. Rancang Bangun Aplikasi Perpustakaan Keliling

Menggunakan Pendekatan Terstruktur. 2020

2.2.3 Flowchart

Flowchart adalah penjelasan menggunakan grafik tentang langkah-langkah dan urutan tata cara atau aturan sebuah program pada umumnya memberikan pengaruh untuk menyelesaikan masalah yang paling penting untuk dipelajari dan lebih lanjut untuk dipelajari.

Terdapat 5 jenis pembagian flowchart, diantaranya adalah *system flowchart*, *process flowchart*, *schematic flowchart*, *document flowchart* dan *program flowchart* (Budiman et al., 2021)

a. *System flowchart*

System flowchart adalah bagian yang secara keseluruhan memperlihatkan arus pekerjaan dari sistem. Bagian ini memperlihatkan dengan jelas urutan-urutan dari tata cara yang ada di dalam sistem.

b. *Process flowchart*

Process flowchart adalah bagian alir yang sering digunakan dalam teknik industri. Guna dari bagian alir ini adalah untuk analisis sistem untuk menunjukkan proses dalam suatu prosedur.

c. *Schematic flowchart*

Schematic flowchart atau bagian alir skematik adalah bagian alir yang hampir sama dengan bagian alir sistem yaitu berfungsi untuk menunjukkan aturan di dalam sistem, letak perbedaannya yaitu penggunaan simbol-simbol bagian alir sistem pada *schematic flowchart*, selain itu juga terdapat gambar-gambar komputer beserta penggunaannya tools lainnya. Meskipun mudah untuk

memahami gambar-gambar tersebut akan sulit dan butuh waktu untuk menggambarnya.

d. *Program flowchart*

Program flowchart adalah bagan yang menunjukkan dengan rinci mengenai langkah-langkah dari proses program yang bersumber dari derivisi bagian alur sistem. Terdapat dua jenis program flowchart, yaitu: program logic flowchart dan detailed computer program flowchart. Program logic flowchart yaitu untuk menunjukkan langkah-langkah yang ada di program komputer menggunakan logika, dimana program logic flowchart ini tersedia dari analisis sistem. Sementara detailed computer program flowchart yaitu untuk menunjukkan instruksi atau tata cara program komputer dengan rinci dimana bagan alir ini telah disediakan oleh pemograman.

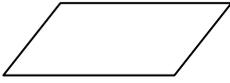
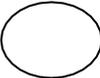
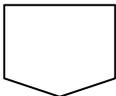
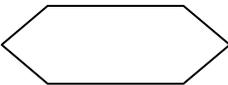
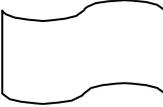
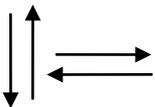
e. Document flowchart

Document flowchart atau form flowchart adalah bagan alir yang mengarahkan arus dari laporan juga formula beserta tembusannya.

Kegunaan flowchart adalah untuk menghadirkan kegiatan manual, kegiatan proses maupun keduanya. Flowchart terdiri atas rangkaian simbol-simbol agar dapat mengkontruksi. Simbol yang digunakan dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol Flowchart

No	Simbol	Nama	Fungsi Simbol
1.		<i>“Terminal”</i>	Awal atau akhir suatu program (Prosedur).

No	Simbol	Nama	Fungsi Simbol
2.		<i>"Output/Input"</i>	Proses input atau output terlepas dari jenis perangkat.
3.		<i>"Process"</i>	Proses operasional computer.
4.		<i>"Decision"</i>	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan, ya/tidak.
5.		<i>"Connector"</i>	Koneksi penghubung proses ke proses lain pada halaman yang sama.
6.		<i>"Offline Connector"</i>	Koneksi Penghubung dari satu proses ke proses lain di halaman lain.
7.		<i>"Predefined Process"</i>	Mewakili ketentuan penyimpanan untuk diproses untuk memberikan awal harga.
8.		<i>"Punched Card"</i>	Input berasal dari kartu atau output ditulis ke kartu.
9.		<i>"Punch Tape"</i>	-
10.		<i>"Document"</i>	Mencetak output dalam format dokumen (melalui printer).
11.		<i>"Flow"</i>	Menyatakan jalannya arus suatu proses.

Sumber : Nurhaliza, Khesya. Mengenal Flowchart dan Pseudocode Dalam

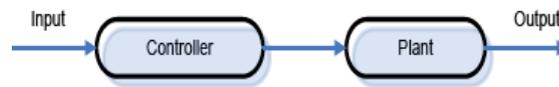
Algoritma dan Pemrograman. 2021.

2.3 Sistem Kontrol

Sistem kontrol merupakan sebuah sistem pengendalian terhadap satu atau beberapa variabel dan parameter sehingga menjadi sebuah satu rangkumanan range tertentu. Dan suatu kumpulan metode yang dipelajari dari sebuah kebiasaan yang dilakukan oleh seorang manusia dalam melakukan suatu aktivitas, dimana seorang manusia membutuhkan suatu penglihatan terhadap suatu kualitas dari yang mereka kerjakan sehingga mempunyai sebuah karakteristik yang diharapkan pada awalnya. Secara umum sistem kontrol adalah kumpulan dari suatu komponen yang saling berhubungan untuk mencapai suatu tujuan tertentu (Friadi & Junadhi, 2019). Sistem kontrol terbagi atas dua jenis yaitu : sistem loop terbuka dan sistem loop tertutup.

2.3.1 Sistem Loop Terbuka (Open-Loop Control System)

Sistem loop terbuka tidak terpengaruh secara langsung dengan sinyal output ketika proses pengendalian sedang terjadi, untuk mengubah sinyal input, sinyal output yang terbentuk tidak bisa dikembalikan, hal ini mengakibatkan terdapat kondisi yang tetap pada in out dan kalibrasi harus dilakukan dahulu. Model dari sistem loop terbuka dapat dilihat pada Gambar 2.2.

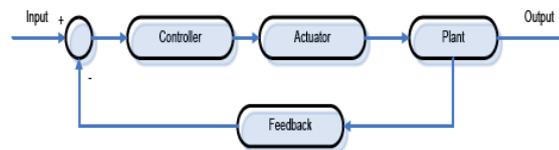


Sumber : Roni & Cekdin. Sistem Kendali Proses Industri. 2020.

Gambar 2.2 Sistem Loop Terbuka

2.3.2 Sistem Loop Tertutup (Closed-Loop Control System)

Sinyal output berpengaruh secara langsung terhadap proses pengendalian di dalam sistem loop tertutup, sistem ini juga dikenal sebagai sistem kendali umpan-balik karena teradinya aksi umpan balik dapat memperkecil kegagalan dari sistem tersebut. Model dari sistem loop tertutup dapat dilihat pada Gambar 2.3.



Sumber : Parulian, Sahat et al. Implementasi Kontrol Lup Tertutup Multi Point Pada Pengatur Temperatur Oven Panggang Roti. 2021.

Gambar 2.3 Sistem Loop Tertutup

2.4 Deteksi Plat Kendaraan

Plat kendaraan adalah salah satu jenis identifikasi sebuah kendaraan. Karakter pada plat di Indonesia terdida dua hal, yaitu nomor polisi dan masa aktif dari plat tersebut. Untuk nomor polisi, plat nomor di Indonesia dimulai dengan sebuah huruf yang digunakan untuk membedakan plat nomor berdasarkan wilayah kendaraan tersebut, kemudian diikuti dengan angka bedasarkan pendaftaran

kendaraannya dan diakhiri dengan huruf lagi berdasarkan asal kota/kabupaten kendaraan tersebut (Tirtana et al., 2021).

2.4.1 Python

Python merupakan bahasa pemrograman yang interperatif dan multiguna dengan filosofi penyusunan yang berpusat pada tingkat keterbacaan kode. Python memiliki fungsi sebagai bahasa yang berfungsi untuk menggabungkan kapabilitas, kemampuan dengan sintaksis kalimat ode yang sangat jelas yang mempunya fungsionalitas Pustaka umum yang besar serta lengkap. Python banyak digunakan *programmer* karena dikenal bahasa yang mudah dipelajari, karena struktur sintak yang rapi dan mudah dipahami.

Bahasa pemrograman python ini didukung multi paradigma pemrograman, utamanya tidak dibatasi pada pemrograman mengarah kepada objek, pemrograman imperaktif, dan pemrograman yang fungsional. Beberapa karakteristik yang terdapat pada python adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan pengolahan memori secara otomatis. Umumnya dimanfaatkan sebagai suatu bahasa skrip walaupun pada pengaplikasiannya lebih luas dan memuat konteks pemanfaatan yang umunya tidak dikerjakan dengan memanfaatkan bahasa skrip (Harani & Hasanah, 2020).

2.4.2 Tesseract OCR

OCR adalah sebuah software untuk mendeteksi teks, angka, maupun sebuah pola karakter pada gambar. OCR sebuah sistem computer yang dapat

membaca sebuah karakter dari sebuah printer atau dari tulisan tangan manusia. Contoh pengaplikasian dari OCR seperti mendeteksi plat kendaraan untuk kebutuhan area parkir, scan barcode untuk penjualan di supermarket, mendeteksi teks bahasa asing, dan lain-lainnya (Nafsin et al., 2020).

Tesseract OCR merupakan sebuah engine Optical Character Recognition yang dikembangkan oleh Google. Pada sebuah software, Tesseract OCR biasanya digunakan sebagai sebuah tambahan library untuk sebuah project dalam pembuatan software yang berfungsi sebagai pendeteksi. Tesseract OCR mendukung berbagai macam bahasa dalam pendeteksian teksnya (Reza Eka Alfarisi, 2020). OCR sebuah teknologi yang *open source* yang dikembangkan pada tahun 1984 hingga 1994. Namun untuk Tesseract baru menjadi sebuah *open source* pada tahun 2005 (Tirtana et al., 2021).

Menurut Kusantoro et al. (2022), ada beberapa langkah dalam sistem kerja metode Tesseract OCR, yaitu:

1. Data capture merupakan proses perubahan suatu objek/dokumen hardcopy menjadi sebuah file citra gambar digital.
2. Preprocessing, pada tahap ini merupakan bagian untuk mengurangi/menghilangkan noise/bagian yang tidak penting pada gambar.
3. Segmentasi, fungsi segmentasi digunakan untuk memisahkan daerah pengamatan pada setiap karakter yang akan dideteksi.
4. Normalisasi, dalam proses normalisasi, dimensi ketebalan pada setiap karakter akan ditambahkan. Algoritma yang digunakan OCR dalam melakukan proses normalisasi adalah algoritma scaling.

5. Pengenalan, tahap ini adalah proses dimana setiap karakter yang diamati akan dilakukan pengenalan karakter berdasarkan ciri-ciri yang sudah ditentukan atau terdapat dalam database.
6. Post-processing merupakan proses terakhir dalam pengenalan karakter, karena akan dilakukan koreksi ejaan sesuai dengan objek/bahasa yang digunakan

2.4.3 Visual Studio Code (VS Code)

VS Code merupakan sebuah editor ringan yang handal dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia untuk Linux, Mac dan Windows. Teks editor ini mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta pemrograman lainnya dengan bantuan dari plugin yang dapat dipasang melalui marketplace Visual Studio Code.

VS Code menyediakan berbagai fitur-fitur yang dapat digunakan diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur extension lainnya. Teks editor VS Code juga bersifat *open source*, kode sumber dari VS code juga dapat dilihat di link Github. Hal ini yang membuat VS Code menjadi favorit oleh para pengembangan aplikasi, karena para *developeper* bisa ikut serta dalam proses pengembangan VS Code kedepannya (Reza Eka Alfarisi, 2020).

2.5 Komponen Utama

2.5.1 Arduino Mega 2560

Arduino adalah sebuah mikrokontroler atmel yang disertai software pendukung untuk melakukan pemrograman, pengembangan sistem digital dan sistem cerdas *Internet of Things* (IoT), dan robotika. Arduino perangkat mikrokontroler yang dapat memproses sebuah input dan output dari sebuah komponen eksternal yang dihubungkan ke Arduino. Arduino adalah sebuah mikrokontroler yang memiliki sifat open source serta mudah untuk diimplementasikan. Hal itu ditujukan agar seseorang dapat membuat sebuah proyek dengan mudah dan menarik. Dan pada proyek kali ini menggunakan Arduino mega 2560 (Wicaksono, 2019). Bentuk fisik dari arduino mega 2560 dapat dilihat pada Gambar 2.4.



Sumber : Budiharto, Widodo. *Menguasai Pemrograman Arduino dan Robot*. 2020.

Gambar 2.4 Arduino Mega 2560

Arduino mega 2560 merupakan sebuah mikrokontroler yang menggunakan IC ATmega2560. Mikrokontroler memiliki pin I/O yang cukup banyak, 54 buah pin digital I/O (15 pin diantaranya bisa digunakan sebagai PWM), 16 buah pin analog, 4 buah pin UART (*Universal Asynchronous receiver /*

transmitter). Arduino mega 2560 memiliki sebuah *oscillator* kritical 16 Mhz, koneksi USB, *power jack* DC, *socket* ICSP (*In-Circuit System Programming*), dan tombol *reset* (Iirsyam & Wiranata, 2020). Spesifikasi arduino mega 2560 dapat dilihat pada Tabel 2.4.

Tabel 2.4 Spesifikasi Arduino Mega 2560

<i>Microcontroller</i>	ATmega2560
<i>Operating Voltage</i>	5 V
<i>Input Voltage (Recommended)</i>	7 - 12 V
<i>Input Voltage (Limit)</i>	6 - 20 V
<i>Digital I/O Pins</i>	54 (<i>of which 15 Provide PWM Output</i>)
<i>Analog Input Pins</i>	16
<i>DC Current Per I/O Pin</i>	20mA
<i>DC Current For 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	256 KB <i>of which 8 KB used by bootloader</i>
SRAM	8 KB
EEPROM	4 KB
<i>Clock Speed</i>	16 MHz
<i>LED BUILTIN</i>	13
<i>Length</i>	101.52 mm
<i>Width</i>	53.3 mm
<i>Weight</i>	37 g

Sumber : Datasheet Arduino Mega 2560

Pin pada Arduino Mega 2560 mempunyai fungsi yang berbeda sehingga tidak semua pin bisa digunakan dengan fungsi yang sama. Kategori dan nama pin dari Arduino Mega 2560 beserta fungsi terdapat pada Tabel 2.5.

Tabel 2.5 Kategori Dan Nama Pin Arduino Mega 2560

Kategori Pin	Nama Pin	Fungsi
<i>Input / Output Digital</i>	0 - 53	Menerima atau mengirim sinyal digital
<i>Input Analog</i>	A0 - A15	Mengubah sinyal analog menjadi sinyal digital
Serial 0	0 (RX) dan 1 (TX)	Pin RX digunakan untuk menerima data serial dan pin
Serial 1	19 (RX) dan 18 (TX)	

Kategori Pin	Nama Pin	Fungsi
Serial 2	17 (RX) dan 16 (TX)	TX untuk mengirim data serial TTL
Serial 3	15 (RX) dan 14 (TX)	
Interupsi Eksternal	2 (<i>Interrupt 0</i>)	Memicu interupsi pada nilai yang rendah, meningkat, menurun atau perubahan nilai
	3 (<i>Interrupt 1</i>)	
	21 (<i>Interrupt 2</i>)	
	20 (<i>Interrupt 3</i>)	
	19 (<i>Interrupt 4</i>)	
18 (<i>Interrupt 5</i>)		
<i>Pulse Width Modulation (PWM)</i>	2 – 13 dan 44 – 46	Menghasilkan sinyal analog dari perangkat digital
SPI	50 (MISO)	Sebagai komunikasi SPI
	51 (MOSI)	
	52 (SCK)	
	53 (SS)	
I2C	20 (SDA)	Sebagai komunikasi I2C atau TWI
	21 (SCL)	
LED	13	Menyalakan LED bawaan papan yang terhubung di pin13
Daya	VIN	Sebagai pin masukan daya Eksternal
	5 V	Sebagai pin keluaran daya 5 Volt
	3,3 V	Sebagai pin keluaran daya 3,3 Volt
	GND	Meniadakan beda potensial
Lainnya	AREF	Tegangan referensi untuk <i>Input</i> analog
	Reset	Menambahkan tombol Reset

Sumber : *Datasheet Arduino Mega 2560*

2.5.1 Esp32-Cam

Esp32-cam merupakan sebuah board pengembangan *WIFI* dan *Bluetooth* dengan mikrokontroller Esp32 dan kamera. Esp 32-Cam adalah sebuah modul yang memiliki open source yang mudah digunakan, salah satunya yaitu dapat capture gambar, *face detection* ,

dan *face recognition*. Esp32-Cam dapat digunakan editor Arduino IDE untuk memprogram modul ini (Isrofi et al., 2021). Tampilan fisik dari esp32-cam dapat dilihat pada Gambar 2.5.



Sumber : Rio Wahyudi & Edidas. Perancang Dan Pembuatan Sistem Keamanan Rumah Berbasis Internet Of Things Menggunakan Esp32-Cam. 2022.

Gambar 2.5 ESP32-Cam

Esp32 cam merupakan sebuah modul yang banyak digunakan dalam berbagai proyek karena modul ini telah dilengkapi dengan mikrokontroler yang terintegrasi, yang bisa bekerja secara mandiri. Selain memiliki konektivitas ke *WIFI* dan *Bluetooth*, esp32-cam juga memiliki kamera video terintegrasi, dan slot microSD sebagai penyimpanan (Arrahma & Mukhaiyar, 2023).

2.5.2 Servo SG90

Motor servo adalah motor DC yang memiliki sistem *feedback* tertutup dimana posisi motornya akan diinformasikan kembali ke rangkaian kontrol didalam servo tersebut. Servo terdiri dari berbagai komponen didalamnya seperti motor DC, serangkaian gear, potensiometer, dan rangkaian kontrol. Potensiometer disini

berfungsi sebagai pengatur batas sudut dari putaran servo. Sedangkan untuk mengatur sudut dari servo berdasarkan lebar pulsa yang dikirim melalui kabel pada servo (Iirsyam & Wiranata, 2020).

motor servo merupakan sebuah jenis actuator elektromekanis yang tidak berputar secara kontinu. Kegunaan dari servo untuk memindahkan ke posisi tertentu dan kemudian berhenti di posisi tersebut. Contoh penggunaan dari motor servo dapat dilihat pada kemudi pesawat terbang, kapal, dan lain-lain.

Pada system ini menggunakan motor servo SG90 adalah sebuah servo kecil dengan keluaran power yang tinggi. Servo ini dapat berotasi sebesar 180 derajat dan dapat bekerja seperti servo standar lainnya hanya saja ukuran lebih kecil. Bentuk fisik dari motor servo SG90 dapat dilihat pada gambar (Wicaksono, 2019). Bentuk fisik dari motor servo SG90 dapat dilihat pada Gambar 2.6.



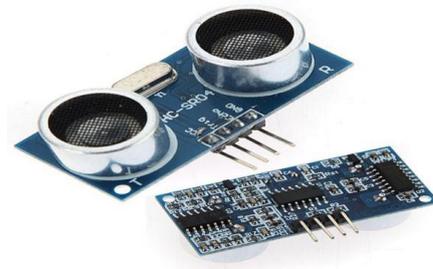
Sumber : Pulungan et al. Rancang Bangun Sistem Parkir dan Ketersediaan Slot Parkir Otomatis Menggunakan Arduino. 2022.

Gambar 2.6 Motor Servo SG90

2.5.3 Sensor ultrasonic HC-SR04

HC-SR04 merupakan sebuah modul sensor ultrasonik yang bisa mengukur jarak dengan *range* dari 2cm sampai 4m dengan ke akurasian mencapai

3mm. Pada modul ini memiliki transmitter, receiver, dan control circuit. Dasar prinsip kerja dari HC-SR04 ini menggunakan IO trigger sedikitnya 10us sinyal high, modul secara otomatis mengirimkan 8 kali 40KHz dan mendeteksi apa terdapat sinyal balik atau tidak, dan jika terdapat sinyal balik, maka durasi waktu dari output high adalah waktu dari pengiriman dan penerimaan ultrasonik (Wicaksono, 2019). Bentuk fisik dari sensor ultrasonik HC-SR04 dapat dilihat pada Gambar 2.7.



Sumber : Malikul. Rancang Bangun Sistem Parkir Secara Otomatis dengan Pendeteksi Tanda Nomor Kendaraan Bermotor Berbasis Pengolahan Citra. 2019.

Gambar 2.7 Sensor Ultrasonik HC-SR04

2.5.4 Sensor Infrared

Sensor berasal dari kata sense merupakan mendefinisikan sensor sebagai piranti yang menerima sebuah stimulus dan merespon dengan sebuah sinyal listrik. Sensor infrared merupakan sebuah modul yang berfungsi sebagai pendeteksi halangan. Cara kerja dari modul ini emitter dari modul akan memancarkan sebuah cahaya infrared dan ketika terkena halangan cahaya tersebut akan terpantulkan dan akan diterima oleh receiver (Pulungan et al., 2022).

Pada penelitian ini menggunakan sensor infrared FC-51 pada sensor ini juga memiliki IR *transmitter* dan IR *receiver*. Transmitter yang akan memancarkan cahaya infra merah, sehingga disebut IR LED. Meskipun bentuk LED nya mirip dengan bentuk LED pada umumnya hanya saja cahayanya tidak bisa dilihat dengan mata manusia dan IR *receiver* yang akan menerima pantulan cahaya yang terkena objek didepannya yang di paparkan oleh IR *transmitter*. IR *transmitter* biasanya berbentuk photodiode dan phototransistor.

Photodiode *infrared* berbeda dari photodiode normal karna photodiode ini hanya dapat menangkap radiasi *infrared* saja. Output dari modul ini bedasarkan intensitas yang diterima oleh photodiode. Pada modul ini terdapat potensiometer yang digunakan untuk mengatur angkauan pendeteksian. Modul ini terdapat komparator IC LM393. Rangkaian komparator menggunakan mode non-inverting dimana ketika tegangan pada pin positif lebih besar dari pin negatif, maka output akan berayun kearah tegangan positif, tetapi ika tegangan pada pin positif lebih kecil dari pin negatif, maka output akan berayun ke tegangan negatif (Wicaksono, 2019). Bentuk fisik dari sensor infrared dapat dilihat pada Gambar 2.8.



Sumber :Ayom Purbo Wiseso. Rancang Bangun Sistem Informasi Ketersediaan Slot Parkir Dalam Mall. 2022.

Gambar 2.8 Sensor Infrared

2.5.5 LCD

LCD (Liquid Crystal Display) adalah sebuah komponen elektronika yang menampilkan sebuah karakter yang dapat disusun sedemikian rupa untuk menampilkan sebuah informasi yang diinginkan. Pencahayaan pada LCD merupakan sebuah efek dari polarizer yang ditempatkan secara vertical dan horizontal, dengan tingkatan pencahayaan berdasarkan besarnya *supply* tegangan yang diterima oleh LCD (Rahardjo, 2022).

Material dari LCD merupakan lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan tegangan, molekul organik panjang dan silindris menyesuaikan diri dengan elektroda pada segmen. Lapisan sandwich memiliki polarizer cahaya vertikal depan dan horizontal belakang mengikuti dengan lapisan reflector.

LCD 16 x 2 merupakan salah satu jenis dari media *display* dari bahan kristal yang pengoperasiannya menggunakan sistem *dotmatrix*. LCD 16 x 2 menampilkan 32 karakter yang terdiri 2 baris dan 16 kolom. LCD berfungsi sebagai *display* yang akan digunakan untuk menampilkan status kerja alat (Irsyam & Wiranata, 2020). Bentuk fisik dari LCD 16x2 dapat dilihat pada Gambar 2.9.



Sumber : Iirsyam & Wiranata. Perancangan Sistem Parkir Mobil Otomatis Menggunakan Nfc Reader Pn532 Berbasis Arduino. 2020.

Gambar 2.9 LCD 16x2

2.5.6 Push Button

Pushbutton merupakan suatu jenis saklar yang berfungsi sama seperti saklar pada umumnya. Prinsip kerja pushbutton sebagai pemutus dan penghubung, yang memiliki dua kondisi diantaranya ada kondisi *ON* dan *OFF*. Istilah *ON* dan *OFF* pada *pushbutton* sangat penting dikarenakan hampir semua listrik yang membutuhkan energi listrik sudah pasti membutuhkan dua kondisi itu pada saklar atau *pushbutton*. Terdapat dua tipe kontak pada *pushbutton* yaitu *Normally Open* (NO) dan *Normally Close* (NC) (Suyono & Hambali, 2020).

Dimana *Normally Open* saklar dalam kondisi terbuka untuk saklar menjadi *ON* berikan sebuah tegangan pada saklar sedangkan *Normally Close* saklar dalam kondisi tertutup untuk memutuskan atau *OFF* saklar berikan sebuah tegangan pada saklar (Santoso & Wijayanto, 2022). Bentuk fisik dari push button dapat dilihat pada Gambar 2.10.



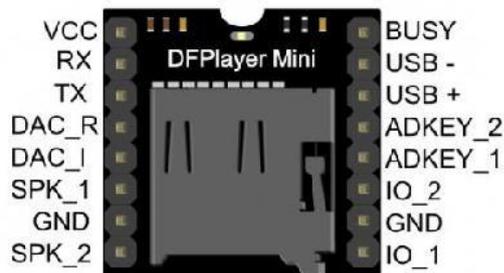
Sumber: Haryono Suyono & Hambali, Perancangan Alat Pengukur Kadar Gula dalam Darah Menggunakan Teknik Non-Invasive Berbasis Mikrokontroler Arduino Uno, 2020.

Gambar 2.10 Push Button

2.5.7 DFPlayer Mini

DFPlayer adalah modul MP3 kecil yang memiliki harga yang murah dengan output secara langsung ke *speaker*. Modul MP3 ini dapat berdiri sendiri tanpa harus menggunakan baterai. DFPlayer mengintegrasikan modul hard decoding secara sempurna mendukung format audio pada umumnya. Selain itu juga mendukung kartu TF dengan FAT16, FAT 32 sistem file. Melalui serial port yang sederhana, user dapat memutar music tanpa memberikan sebuah perintah (Sunarwan, 2020).

Modul DFPlayer membutuhkan *amplifier* atau *speaker* sebagai penguat suara karena modul ini tidak memiliki *output*. Untuk dapat mengakses file audio harus dilakukan dengan cara membuat folder bernama “mp3” pada SD Card terlebih dahulu, setelah itu audio didalam folder “mp3” haruslah diubah menjadi format angka 4 digit seperti 0001, 0002, dst. Untuk mencegah terjadinya *noise* pada audio dapat dilakukan dengan memberikan resistor pada kabel RX (BETA & Astuti, 2019). Bentuk fisik dari DFPlayer Mini dapat dilihat pada Gambar 2.11.



Sumber : Sunarwan. Perancangan Dan Pembuatan Alat Pemutar Narasi Audio Tentang Objek Museum Menggunakan Teknologi Rfid Berbasis Arduino. 2020.

Gambar 2.11 DFPlayer Mini

2.5.8 Speaker

Speaker merupakan sebuah komponen elektronika yang terdiri dari kumparan, membrane dan magnet sebagai bagian yang saling terkait. Tanpa adanya membrane, sebuah *speaker* tidak akan mengeluarkan suara. Fungsi dari *speaker* adalah mengubah gelombang listrik menjadi getaran suara. Proses mengubah sebuah gelombang listrik menjadi sebuah gelombang suara terjadi karena adanya arus DC audio dari penguat audio kedalam kumparan yang menghasilkan gaya magnet sehingga akan menggerakkan membrane, kuat lemahnya arus listrik yang diterima, akan mempengaruhi getaran pada membrane, getaran dari membrane ini yang menghasilkan gembong bunyi yang dapat didengar (Sunarwan, 2020). Bentuk fisik dari speaker dapat dilihat pada Gambar 2.12.



Sumber : Sunarwan. Perancangan Dan Pembuatan Alat Pemutar Narasi Audio Tentang Objek Museum Menggunakan Teknologi Rfid Berbasis Arduino. 2020.

Gambar 2.12 Speaker

2.6 Komponen pendukung

2.6.1 Resistor

Resistor merupakan sebuah komponen elektronika pasif yang berfungsi sebagai pemberi hambatan serta pengatur tegangan dan arus terhadap listrik. Resistor memiliki satuan yang disebut ohm (Ω) (Dwi Cahyono et al., 2023)..

Menurut (Budiharto, 2020) dibutuhkan resistor dengan spesifikasi tertentu dalam rangkaian listrik seperti besarnya hambatan, karakteristik hambatan terhadap suhu dan jumlah arus yang boleh dilewati. Bentuk resistor dapat dilihat pada Gambar 2.13 dibawah:



Sumber : Bagus Dwi Cahyono, dkk, Pengembangan Media Pembelajaran Elektronika Dasar untuk Memahami Nilai Resistor Berdasarkan Kode Warna 3 Gelang dan 4 Gelang Bagi Siswa SMK Kelas X Jurusan Teknik Otomasi Industri, 2023.

Gambar 2.13 Resistor

Resistor memiliki berbagai nilai resistensinya tergantung dari warna yang ada pada gelang resistor. Makna dari warna gelang resistor dapat dilihat pada Tabel 2.6 berikut:

Tabel 2.6 Warna Gelang Resistor

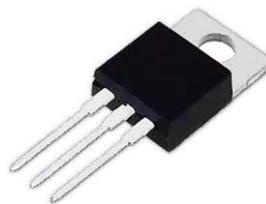
Warna	Digit	Multiplier	Tolerance
Hitam	0	1	
Coklat	1	10	$\pm 1\%$
Merah	2	100	$\pm 2\%$
Orange	3	1000	
Kuning	4	10.000	
Hijau	5	100.000	$\pm 0,5\%$
Biru	6	1.000.000	$\pm 0,25\%$
Violet	7		$\pm 0,1\%$
Abu-abu	8		$\pm 0,05\%$
Putih	9		
Emas		0,1	$\pm 5\%$
Silver		0,01	$\pm 10\%$
None			$\pm 20\%$

Sumber : Bagus Dwi Cahyono, dkk, Pengembangan Media Pembelajaran Elektronika Dasar untuk Memahami Nilai Resistor Berdasarkan Kode Warna 3 Gelang dan 4 Gelang Bagi Siswa SMK Kelas X Jurusan Teknik Otomasi Industri, 2023.

2.6.2 IC Regulator

IC *regulator* atau IC pengatur tegangan merupakan komponen elektronika yang mempunyai fungsi untuk mengatur tegangan agar stabil dan siap digunakan pada saat dibutuhkan. Tidak hanya itu IC *regulator* juga dapat digunakan sebagai pembatas tegangan agar tegangan yang masuk ke rangkaian tidak terlalu tinggi. IC Regulator banyak digunakan pada perangkat elektronik sehari-hari, terutama pada adaptor yang menyuplai berbagai jenis tegangan DC.

Menurut (Silaban et al., 2020) fungsi regulator adalah untuk menstabilkan arus yang beredar pada rangkaian elektronik. Regulator memiliki seri yang berbeda-beda. Seri LM78XX merupakan rangkaian regulator tiga terminal yang menghasilkan tegangan keluaran tetap sebesar XX Volt. Tata letak pin penyesuaian IC yang digunakan oleh pada catu daya. Bentuk IC *Regulator* dapat dilihat pada Gambar 2.14 dibawah:



Sumber: Datasheet IC regulator

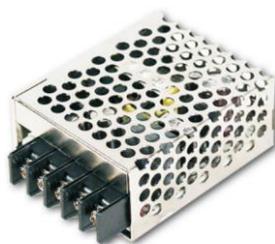
Gambar 2.14 IC Regulator

2.6.3 Power supply

Power supply atau dalam bahasa Indonesia catu daya adalah sebuah alat Listrik yang dapat menyediakan energi Listrik untuk perangkat listrik ataupun elektronika lainnya. Pada dasarnya catu daya ini memerlukan sumber energi listrik

PLN yang besar dan diubah menjadi energi listrik yang dibutuhkan oleh perangkat elektronika lainya (Amalia Yunia Rahmawati, 2020).

Unit catu daya ini ditujukan untuk memberikan daya pada peralatan sistem alarm, yang membutuhkan tegangan pasokan sebesar 12V DC dan arus beban $I=5A$. Desain ini memungkinkan perubahan mudah dari tegangan keluaran, dalam rentang 12V hingga 15V DC, menggunakan potensiometer. Unit catu daya dilindungi terhadap pintshort circuit, beban berlebih, dan tegangan berlebih (Datasheet Power Suply 12V). Bentuk fisik dari power supply dapat dilihat pada Gambar 2.15.



Sumber: Datasheet Power Supply

Gambar 2.15 Power Supply

2.6.4 Kabel Jumper

Kabel jumper ialah kabel penghubung yang umumnya digunakan untuk konstruksi sistem. Kabel ini menjadi bagian penting dalam rangkaian elektronika dan berperan sebagai penghubung antara rangkaian Arduino dan breadboard (Mindasari & Meilantika, 2022). Bentuk fisik dari kabel jumper dapat dilihat pada Gambar 2.16.



Sumber : Mindasari & Meilantika. Sistem Keamanan Kotak Amal di Musala Sabilul Khasanah Berbasis Arduino UNO .2022

Gambar 2.16 Kabel Jumper

2.7 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah software yang khusus digunakan untuk membuat program melalui Arduino, dengan bahasa lain Arduino IDE menjadi sebuah media untuk melakukan program *board* Arduino. Arduino IDE merupakan sebuah software open source yang memudahkan seorang programmer untuk menuliskan program dan meng-uplod code yang telah ditulis ke dalam mikrokontroler. Arduino IDE ini menggunakan bahasa pemrograman C++ dengan versi yang telah disederhanakan, sehingga lebih mudah dalam belajar pemrograman. Software ini dapat bekerja dalam berbagai *platform* seperti Windows, Mac dan Linux (Romadhon & Umam, 2021). Bentuk tampilan awal Arduino IDE dapat dilihat pada Gambar 2.17.



Sumber : Dokumen Pribadi

Gambar 2.17 Arduino IDE

2.7.1 Pengenalan Software Arduino IDE

Arduino Integrated Development Environment - atau Arduino Software (IDE) - berisi editor teks untuk menulis kode, area pesan, konsol teks, toolbar dengan tombol untuk fungsi-fungsi umum dan serangkaian menu. Termasuk menghubungkan ke perangkat keras Arduino untuk meng-upload program dari komputer.

- ✓ *Verify*: Memeriksa kode Anda untuk kesalahan kompilasi.
- ➔ *Upload* : Mengkompilasi kode Anda dan mengunggahnya ke papan yang dikonfigurasi. Lihat pengunggahan di bawah untuk detail.
Catatan: Jika Anda menggunakan pemrogram eksternal dengan papan Anda, Anda dapat menahan tombol “Shift” pada keyboard ketika menggunakan ikon ini. Teks akan berganti menjadi “Upload using Programmer” (Unggah menggunakan Programmer).
- 📄 *New*: Membuat sketsa baru

-  *Open*: Menampilkan sebuah menu dari semua sketsa pada *sketchbook* (buku sketsa) Anda. Mengklik salah satu akan membukanya ke dalam jendela saat ini dan menggantikan isinya.
-  *Save*: Menyimpan sketsa Anda.
-  *Serial Monitor*: Membuka serial monitor.

Perintah tambahan dapat ditemukan dalam lima menu: **File**, **Edit**, **Sketch**, **Tool**, **Help**. Menu-menu ini *context sensitive*, artinya *item-item* yang tersedia hanya yang relevan dengan pekerjaan yang bersangkutan.

1. File
 - a. *New* : membuat sebuah sketsa baru pada editor, termasuk dengan struktur minimal sketsa.
 - b. *Open* : memungkinkan membuka sketsa yang ada pada media penyimpanan komputer.
 - c. *Open Recent* : menyediakan daftar pendek sketsa yang baru-baru yang siap dibuka.
 - d. *Sketchbook* : menampilkan sketsa saat ini dalam struktur folder *sketchbook*; mengklik salah satu nama akan membuka sketsa yang bersangkutan pada editor baru.
 - e. *Examples* : beberapa contoh telah disediakan oleh Arduino Software (IDE) atau pustaka yang ditampilkan dalam menu *item* ini. Semua contoh tersusun dalam struktur pohon yang mudah diakses.
 - f. *Close* : menutup jendela Arduino Software yang bersangkutan.

- g. *Save* : menyimpan sketsa dengan nama yang bersangkutan, jika belum dinamai sebelumnya, bisa dinamai pada jendela “Save as...”.
- h. *Save as* : memungkinkan menyimpan sketsa yang bersangkutan dengan nama yang berbeda.
- i. *Page Setup* : Menampilkan jendela “Page Setup” (pengaturan halaman) untuk pencetakan.
- j. *Print* : mengirim sketsa saat ini ke printer berdasarkan pengaturan yang ditetapkan pada Page Setup.
- k. *Preferences* : membuka jendela Preferences dimana beberapa pengaturan IDE bisa dikustomisasi.
- l. *Quit* : menutup semua jendela IDE. Sketsa yang terbuka ketika Quit dipilih akan otomatis dibuka kembali pada waktu memulai IDE yang selanjutnya.

2. Edit

- a. *Undo/Redo* : kembali satu atau lebih tahap yang Anda lakukan saat editing; ketika Anda memilih Undo, Anda dapat maju satu atau lebih tahap kembali dengan Redo.
- b. *Cut* : memindahkan teks yang diseleksi dari editor ke papan klip (*clipboard*).
- c. *Copy* : menyalin teks yang diseleksi dari editor ke papan klip.
- d. *Copy For Forum* : menyalin kode dari sketsa Anda ke papan klip dalam suatu form yang cocok diposting ke forum, lengkap dengan pewarnaan sintaks.

- e. *Copy as HTML* : menyalin kode dari sketsa Anda ke papan klip sebagai HTML, cocokdi *embed* (tanam) pada laman web.
 - f. *Paste* : meletakkan isi papan klip tepat pada posisi kursor dalam editor.
 - g. *Select All* : menyeleksi dan menyorot seluruh isi editor.
 - h. *Comment/Uncomment* : memberi atau menghapus penanda komen “//” pada awal baris yang diseleksi.
 - i. *Increase/Decrease Indent* : menambah atau mengurangi sebuah spasi pada awal setiap baris yang diseleksi, memindahkan teks satu spasi ke kanan atau menghapus satu spasi di awal.
 - j. *Find* : membuka jendela “Find and Replace” dimana Anda dapat mengetik teks untuk dicari dalam sketsa yang bersangkutan berdasarkan beberapa pilihan.
 - k. *Find Next* : menyorot kejadian selanjutnya (jika ada) dari kata yang diketik pada jendela “Find”, relatif ke posisi kursor.
 - l. *Find Previous* : menyorot kejadian sebelumnya (jika ada) dari kata yang diketik pada jendela “Find”, relatif ke posisi kursor.
3. Sketch
- a. *Verify/Compile* : memeriksa sketsa Anda untuk kesalahan kompilasi, ini akan melaporkan penggunaan memori kode dan variabel pada konsol.
 - b. *Upload* : mengkompilasi dan memuat berkas *binary* ke papan yang dikonfigurasi melalui Port yang dikonfigurasi.
 - c. *Upload Using Programme* : ini akan menulis ulang *bootloader* pada papan, Anda akan perlu menggunakan *Tools > Burn Bootloader* untuk

memperbaikinya dan memungkinkan Mengunggah ke port USB serial kembali. Bagaimanapun, ini memungkinkan Anda untuk menggunakan kapasitas penuh memori Flash untuk sketsa Anda. Tolong catat bahwa perintah ini TIDAK akan membakar sekering. Untuk melakukannya perintah *Tools > Burn Bootloader* harus dieksekusi.

- d. *Export Compiled Binary* : menyimpan sebuah berkas .hex yang bisa diarsipkan atau dikirim ke papan menggunakan alat lain.
- e. *Show Sketch Folder* : membuka folder sketsa yang bersangkutan.
- f. *Include Library* : menambah pustaka ke sketsa Anda dengan memasukkan *statements #include* pada awal kode Anda. Untuk detail lihat pustaka di bawah. Anda juga dapat mengakses *Library Manager* dan mengimpor pustaka baru dari berkas .zip.
- g. *Add File* : menambah sebuah berkas sumber ke sketsa (disalin dari lokasinya). Berkas baru akan muncul dalam tab baru pada jendela sketsa. Berkas dapat dihapus dari sketsa menggunakan menu tab (ikon segitiga kecil dibawah ikon serial monitor pada sisikanan *toolbar*).

4. Tools

- a. *Auto Format* : memformat kode Anda dengan baik. Contoh memasukkannya jadi kurung kurawal membuka dan menutup lurus, dan teks dalam kurung kurawal juga diluruskan.
- b. *Archive Sketch* : mengarsipkan sebuah salinan dari kode yang bersangkutan ke format .zip. Arsip ini ditempatkan pada direktori yang sama dengan sketsa.

- c. *Fix Encoding & Reload* : memperbaiki ketidaksesuaian yang mungkin antara peta karakter *encoding* dan peta karakter sistem operasi lain.
- d. *Serial Monitor* : membukan jendela *serial monitor* dan menginisialisasi perubahan data dengan suatu papan yang terhubung pada port yang bersangkutan. Biasanya mengatur ulang papan, jika papan itu mendukung pengaturan ulang diatas pembukaan port serial.
- e. *Board* : memilih papan yang Anda gunakan. Lihat di bawah untuk deskripsi berbagai macam papan.
- f. *Port* : menu ini mengandung semua perangkat serial (nyata atau virtual) pada mesin Anda. Seharusnya otomatis *me-refresh* setiap saat Anda membuka menu *tools* tingkat atas.
- g. *Programmer* : untuk menyeleksi sebuah perangkat keras pemrogram ketika memrogram sebuah papan atau *chip* dan tidak menggunakan hubungan USB-serial *onboard*. Secara norma Anda tidak membutuhkan ini, tetapi jika Anda membakar *bootloader* ke mikrokontroler baru, Anda akan menggunakan ini.
- h. *Burn Bootloader* : memungkinkan Anda membakar sebuah *bootloader* ke mikrokontroler pada papan Arduino. Ini tidak dibutuhkan untuk penggunaan normal papan Arduino atau Genuino, tetapi sangat berguna jika Anda membeli mikrokontroler Atmega baru (secara normal tanpa *bootloader*). Pastikan Anda telah memilih papan yang benar pada menu **Boards** sebelum membakar *bootloader* ke papan target. Perintah ini juga mengatur sekering yang benar.

2.8 Bahasa Pemrograman Arduino

Perancangan sistem berbasis Arduino sangat erat kaitannya dengan sebuah pemrograman. Dalam hal ini program berguna untuk bagaimana pada akhirnya sistem dapat berjalan sesuai dengan target yang diinginkan. Bahasa C adalah bahasa yang digunakan dalam pemrograman Arduino. Karena Bahasa C merupakan bahasa yang memiliki peranan besar dalam berkembangnya bahasa pemrograman lain dan memudahkan *programer* untuk memprogram sebuah sistem kendali berbasis Arduino. Berikut merupakan penjelasan dari pemrograman Arduino.

2.8.1 Struktur

Setiap menulis sebuah pemrograman dalam Arduino harus memiliki dua fungsi yang disebut dengan void yaitu void *setup* dan void *loop*. Void *setup* akan dijalankan satu kali saat program Arduino dijalankan dan void *loop* akan dijalankan setelah void *setup* selesai dijalankan. Void loop akan dijalankan secara terus menerus sampai dengan daya dimatikan (Wicaksono, 2019). Contoh void setup dan void loop dalam sketch ditunjukkan oleh Gambar 2.18.



```
void setup() {  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
}
```

Sumber : Dokumen Pribadi

Gambar 2.18 Struktur dalam Bahasa Arduino

2.8.2 Variabel

Variabel digunakan untuk menyimpan atau memindahkan angka maupun karakter didalam program. variabel adalah sebuah cara untuk menamai dan menyimpan sebuah nilai yang akan digunakan pada program, seperti data dari sensor atau sebuah nilai sementara pada sebuah perhitungan. Sebelum variabel digunakan , variabel harus dideklarasikan terlebih dahulu. Mendeklarasikan sebuah variabel berarti mendefinisikan tipe variabel dan mungkin saja dapat uga memberikan inisialisasi nilai pada variabel tersebut. Variabel tidak harus diberikan inisialisasi nilai ketika dideklarasikan tapi hal tersebut juga sering berguna jika dilakukan (Wicaksono, 2019).

2.8.3 Tipe data

Pada umumnya data terbagi atas 2 tipe yaitu variabel dan konstanta. Variabel digunakan sebagai tempat penyimpanan suatu nilai yang dapat diubah

selama eksekusi program, sedangkan konstanta nilainya tidak dapat diubah karena bersifat tetap. Terdapat beberapa tipe data menurut (Widharma & Wiranata, 2022) yang akan dijelaskan pada Tabel 2.7 dibawah:

Tabel 2.7 Jenis Tipe Data

Tipe	Ukuran (Bit)	Range
<i>Bit</i>	1	0 – 1
<i>Char</i>	8	-128 – 127
<i>Unsigned char</i>	8	0 – 255
<i>Signed char</i>	8	-128 – 127
<i>Int</i>	16	-32768 - 32767
<i>Short int</i>	16	-32768 – 32767
<i>Unsigned int</i>	16	0 – 65535
<i>Signed int</i>	16	-32768 – 32767
<i>Long int</i>	32	-2147483648 – 21474836467
<i>Unsigned long int</i>	32	0 – 4294967295
<i>Signed ling int</i>	32	-2147483648 – 21474836467
<i>Float</i>	32	1.175e-38 – 3.402e38
<i>Double</i>	32	1.175e-38 – 3.402e38

Sumber: Widharma & Wiranata. *Mikrokontroler Dan Aplikasi* .2022

2.8.4 Operator

Operator merupakan suatu simbol yang terdapat dalam program dalam melakukan suatu operasi. Terdapat dua jenis operator pada Arduino, yaitu operator aritmatika dan operator kondisi (Wicaksono, 2019).

2.7.1 Operator Aritmatika

Operator aritmatika merupakan simbol yang digunakan dalam operasi matematika seperti penjumlahan, pengurangan, perkalian dan pembagian (Wicaksono, 2019). Operator aritmatika Arduino dapat dilihat pada Tabel 2.8.

Tabel 2.8 Operator Aritmatika

Operator Aritmatika	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagi (Modulus)

Sumber: Widharma & Wiranata. *Mikrokontroler Dan Aplikasi* .2022

2.7.2 Operator Perbandingan

Operator perbandingan merupakan operator yang digunakan untuk operasi logika perbandingan antara dua nilai atau variabel (Wicaksono, 2019). Operator perbandingan Arduino dapat dilihat pada Tabel 2.9.

Tabel 2.9 Operator Perbandingan

Operator Perbandingan	Keterangan
<	Lebih kecil
<=	Lebih kecil atau sama dengan
>	Lebih besar
>=	Lebih besar atau sama dengan
==	Sama dengan
!=	Tidak sama dengan

Sumber: Widharma & Wiranata. *Mikrokontroler Dan Aplikasi* .2022

2.7.3 Operator Logika

Operator logika adalah operator yang digunakan untuk menghubungkan dua buah operator lewat relasi. Operator ini sering digunakan untuk kondisi di dalam *statement* percabangan IF atau untuk keluar dari proses perulangan. Operator logika akan mengeluarkan nilai *true* (1) dan *false* (0) (Wicaksono, 2019). Operator logika Arduino dapat dilihat pada Tabel 2.10.

Tabel 2.10 Operator Logika

Operator Logika	Keterangan
!	Logika NOT
&&	Logika AND
	Logika OR

Sumber: Widharma & Wiranata. *Mikrokontroler Dan Aplikasi*. 2022

2.8.5 Kondisi

Struktur kondisi biasa digunakan untuk menentukan kondisi mana yang akan dieksekusi oleh program setelah kondisi terpenuhi. Struktur ini biasanya digunakan untuk memilih program dari berbagai opsi kondisi (Wicaksono, 2019). Beberapa struktur kondisi akan dijabarkan sebagai berikut.

2.7.5.1 Kondisi If

Pada kondisi *If* akan dilakukan pemeriksaan terhadap kondisi pada kode program. Jika kondisi pada struktur ini tidak terpenuhi (*false*), maka program akan melewati kode yang ada dalam struktur tersebut dan malanjutkan proses pemeriksaan. Jika kondisi terpenuhi (*true*), maka program akan mengeksekusi kode program yang terdapat didalam struktur tersebut. Struktur kondisi *if* dapat dilihat sebagai berikut:

```
if (kondisi){
    // statement 1
}
```

2.7.5.2 Kondisi *If-Else*

Kondisi ini merupakan lanjutan dari kondisi *If*. Penggunaan kondisi ini memungkinkan program untuk memilih dua atau lebih jalur eksekusi berdasarkan kondisi yang diberikan. Jika kondisi bernilai *true* maka kode yang terdapat didalam *else* akan dieksekusi, begitu juga dengan *false*. Jika kondisi bernilai *false* terpenuhi, maka kode yang terdapat didalam *else* akan dieksekusi. Struktur kondisi *if-else* dapat dilihat sebagai berikut:

```

if (kondisi){
    // statment 1
}else{
    // statment 2
}

```

2.7.5.3 Kondisi *If Else If Else*

Pada struktur kondisi ini akan dilakuakn pemeriksaan kondisi ini dimulai dari kondisi ke-1 sampai terakhir. Pernyataan ke-1 akan dieksekusi jika kondisi pertama terpenuhi, jika kondisi ke-1 tidak terpenuhi maka kondisi ke-2 akan diperiksa. Jika kondisi ke-2 terpenuhi maka pernyataan ke-2 akan dieksekusi. Jika kondisi ke-1 dan ke-2 tidak dieksekusi maka pernyataan ke-3 akan dieksekusi. Struktur kondisi *if-else if-else* dapat dilihat sebagai berikut:

```

if (kondisi){
    // statement 1
}else if (kondisi){

```

```
        // statement 2
    }else{
        // statement 3
    }
```

2.7.5.4 Kondisi Switch Case

Kondisi ini mengizinkan untuk memilih satu dari beberapa pilihan. Struktur ini berguna ketika menghadapi beberapa nilai atau kondisi yang berbeda - beda namun ingin melakukan tindakan yang berbeda untuk setiap situasinya. Struktur kondisi *switch case* dapat dilihat sebagai berikut:

```
switch (nilai penentu){
    case nilai1: //tindakan yang dilakukan jika nilai penentu sama
                dengan nilai1
    break;
    case nilai2: //tindakan yang dilakukan jika nilai penentu sama
                dengan nilai2
    break;
    default: //tindakan yang dilakukan jika nilai penentu tidak ada
            yang cocok
    break;
}
```

2.8.6 Perulangan (*Loop*)

Perulangan digunakan untuk mengulang suatu pernyataan yang terdapat didalam blok perulangan selama kondisi yang diberikan masih terpenuhi. Struktur perulangan akan dijelaskan sebagai berikut:

2.7.6.1 Perulangan *For*

Perulangan *for* memungkinkan mengulangi sejumlah instruksi atau perintah dalam program selama kondisi tertentu terpenuhi. Struktur perulangan *for* dapat dilihat sebagai berikut:

```
for (inisialisasi nilai; pengujian; penambahan/pengurangan) {  
    // statement  
}
```

2.7.6.1 Perulangan *While*

Perulangan *while* adalah perulangan yang apabila kondisi yang diberikan bernilai *true* maka eksekusi program akan terus berjalan, eksekusi akan keluar dari perulangan ketika kondisi perulangan tersebut telah bernilai *false* yang kemudian program akan beralih ke instruksi selanjutnya. Struktur perulangan *while* dapat dilihat sebagai berikut:

```
while (kondisi) {  
    // statement  
}
```

2.7.6.1 Perulangan *Do-While*

Pada perulangan *do-while*, program akan mengeksekusi pernyataan yang terdapat dalam *do-while* terlebih dahulu, kemudian akan memeriksa *conditional expression*. Jika *conditional expression* bernilai *true* maka program akan mengeksekusi kembali pernyataan tersebut hingga *conditional expression* bernilai *false*. Ketika *conditional expression* bernilai *false* maka program akan keluar dari perulangan *do-while* sepenuhnya. Perulangan ini biasanya dibutuhkan apabila program membutuhkan perulangan yang setidaknya dilakukan sekali. Struktur perulangan *do-while* dapat dilihat sebagai berikut:

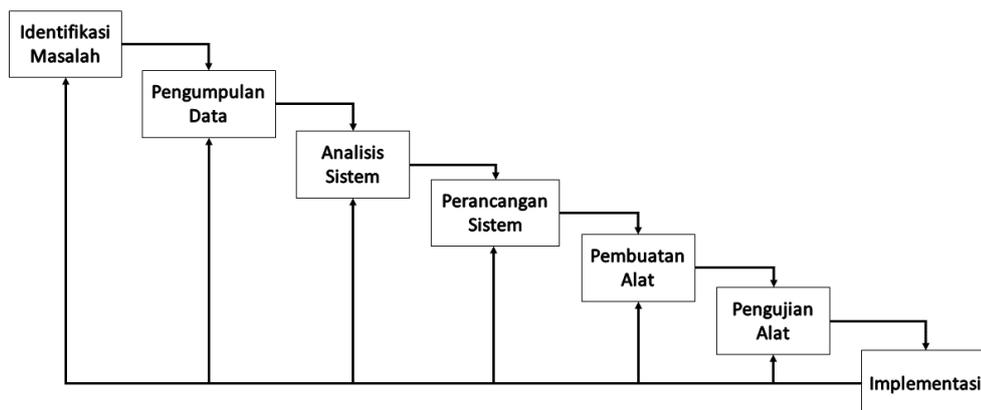
```
do {  
    // statement  
} while(kondisi);
```

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Kerja Penelitian

Kerangka kerja adalah sebuah *step by step* untuk menyelesaikan sebuah persoalan yang akan dibahas. Adapaun kerangka kerja penelitian ini dibuat untuk memberikan susunan pembuatan sistem supaya lebih jelas. Dalam perancangan ini banyak yang harus dilihat terlebih dahulu agar memudahkan peneliti untuk menyelesaikan masalah tersebut. Adapun kerangka kerja dari penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Kerangka Kera Penelitian

3.2 Uraian Kerangka Kerja Penelitian

Bedasarkan kerangka kerja penelitian yang telah dijelaskan diatas, maka dapat diuraikan tahapan-tahapan dalam menyelesaikan peneltitian tersebut sebagai berikut :

3.2.1 Identifikasi Masalah

Masalah yang telah diidentifikasi adalah kurangnya sistem informasi pada lahan parkir mengenai ketersediaan slot parkir terutama pada gedung bertingkat. Tidak adanya sistem yang memandu pengendara ketempat slot parkir yang masih kosong. Dan masalah selanjutnya tidak adanya sistem untuk pelacakan kendaraan pada area parkir untuk pengguna yang lupa posisi kendaraan mereka.

3.2.2 Pengumpulan Data

Penelitian akan dilakukan pengumpulan data yang mendukung perancangan dan implementasi sistem monitoring area parkir untuk mengatasi masalah kurangnya informasi ketersediaan slot parkir, navigasi pengendara, dan pelacakan kendaraan. Data yang relevan melibatkan informasi ketersediaan slot parkir, termasuk jumlah total slot dan status slot pada waktu tertentu. Selain itu, akan dikumpulkan data navigasi, seperti layout lahan parkir dan informasi jalur terpendek ke slot kosong. Sistem pelacakan kendaraan akan mencatat posisi parkir, dan nomor plat kendaraan pengguna. Yang mana pengumpulan data yang didapat membantu menganalisis, pembuatan alat dan menghasilkan laporan yang akurat dengan metode penelitian sesuai dengan yang dijelaskan berikut:

3.2.2.1 Waktu Penelitian

Penelitian dilakukan dengan memproses data yang berhubungan dengan alat yang akan dibuat. Waktu pengumpulan data ini mulai dilakukan sejak bulan September 2023 hingga Februari 2024. Jadwal penelitian tersebut dapat dilihat pada Tabel 3.1.

Tabel 3.1 Jadwal Penelitian

Kegiatan	2023												2024							
	Oktober				November				Desember				Januari				February			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Identifikasi Masalah																				
Pengumpulan Data																				
Analisis Sistem																				
Perancangan Sistem																				
Pembuatan Sistem																				
Pengujian Sistem																				

3.2.2.2 Metode Penelitian

Dalam melakukan penelitian agar mendapatkan hasil yang sesuai dengan tujuan penelitian, maka sekiranya diperlukan suatu metodologi penelitian yang umum dilakukan.

1. Observasi (Wawancara)

Metode observasi, khususnya melalui wawancara akan digunakan untuk mendapatkan data real-time mengenai perilaku pengendara dan pola penggunaan lahan parkir. Dengan mengamati secara langsung aktivitas di lahan parkir, penelitian ini akan mengidentifikasi hambatan yang dihadapi pengendara, preferensi mereka terkait slot parkir, dan harapan terhadap sistem parkir otomatis. Wawancara dengan pengguna parkir dan petugas parkir akan memberikan pemahaman mendalam tentang pengalaman mereka dan kontribusi berharga untuk merancang solusi yang sesuai dengan kebutuhan pengguna.

2. Riset Perpustakaan (*Library Research*)

Mengumpulkan data data yang dibutuhkan untuk mengatasi dari masalah yang telah diidentifikasi dengan membaca buku-buku, jurnal dan referensi-referensi lainnya langkah ini untuk membangun dasar teoritis yang kokoh. Dengan mencari aspek yang terkait dengan informasi slot parkir, sistem navigasi, dan metode pelacakan kendaraan.

3. Penelitian Laboratorium (*Laboratory Research*)

Penelitian laboratorium dimaksudkan untuk menguji sistem yang dibuat. Metode ini dilakukan untuk menguji konsep yang ada dengan menggunakan peralatan yang sesuai. Adapun perangkat *hardware* dan *software* yang penulis gunakan dalam penelitian akan dijelaskan pada Tabel 3.2.

Tabel 3.2 Hardware dan Software

HARDWARE	SOFTWARE
<ul style="list-style-type: none"> • 1 unit laptop ASUS K401LB 8 GB RAM • 1 unit Arduino Mega2560 • 1 unit Webcam • Kabel Downloader • Solder & Timah • Multimeter 	<ul style="list-style-type: none"> • OS Windows 10 Profesional 64 bit • Arduino IDE • Google • Sketchup • Fritzing • Draw.io

3.2.3 Analisis sistem

Setelah melakukan pengumpulan data, maka kerangka penelitian selanjutnya adalah analisis sistem. Analisis sistem ini bertujuan untuk mengetahui *input*, proses, dan *output* yang ada pada sistem yang akan dibuat. Adapun input yang digunakan dalam perancangan sistem ini adalah sensor HC-SR04, Webcam, *push bottom*, dan sensor infrared yang dipasangkan pada *prototype* sistem parkir. Data yang didapat dari inputan tersebut akan diproses oleh mikrokontroler yang nantinya akan menghasilkan beberapa *output* seperti menampilkan hasil bacaan sensor ke LCD 16X2, mengaktifkan motor servo dan *DFPlayer*.

3.2.4 Perancangan Sistem

Panduan dibutuhkan sebelum membuat alat untuk memudahkan penelitian ini. Beberapa panduan tersebut akan dijelaskan sebagai berikut:

1. *Context Diagram*

Komponen *input* dan *output* digambarkan menjadi sebuah bentuk kesatuan dalam membangun sistem yang dirancang. Dimana komponen input yang dimaksud ada webcam, sensor ultrasonik, push button, dan sensor infrared yang nantinya data inputan tersebut masuk dan diinstruksikan ke mikrokontroller dan mengirimkan datanya ke modul program dan mengirimkan instruksi ke output yang digunakan seperti LCD 16x2, motor servo, dan DFPlayer.

2. *Data Flow Diagram*

Data flow diagram adalah suatu diagram yang menggambarkan alir data dalam suatu entitas ke sistem atau sebaliknya. Yang mana nantinya akan menggambarkan alir data dari entity yang digunakan dalam sistem ini seperti *webcam*, sensor ultrasonik, sensor infrared, *push button*, LCD 16x2, motor servo, *DFPlayer*, dan *speaker*.

3. *Blok Diagram*

Blok diagram merupakan sebuah diagram yang dibuat untuk memudahkan seseorang dalam mengenal komponen elektronika pada alat sekaligus memahami bagaimana alur kerja pada alat yang dibuat. Komponen elektronika yang digunakan dalam sistem ini berupa input dan output seperti *webcam*, sensor ultrasonik, sensor infrared, *push button*, LCD 16x2, motor servo, *DFPlayer*, dan *speaker*.

4. Flowchart

Flowchart merupakan proses atau alur kerja yang disajikan dalam bentuk diagram. Diagram ini menggunakan simbol-simbol grafis untuk menggambarkan langkah-langkah dalam suatu proses secara berurutan. Dimana pada sistem ini tahapan dimulai dengan pengaktifan sistem kemudian dengan menekan *push button* akan melakukan perintah ke webcam untuk mencapture gambar setelah gambar tadi di proses dan dibaca data hasil bacaan dikirim ke database dan layer lcd diluar akan menampilkan sisa slot parkir yang tersedia dan setelah itu DFPlayer akan aktif dan memberikan informasi slot parkir yang harus di tepati dan setelah informasi disampaikan, motor servo akan membuka palang, dan setelah melewati sensor infrared palang akan tertutup secara otomatis, dan ketika sensor ultrasonik mendeteksi adanya kendaraan dia akan menampilkan data no kendaraan ke lcd didalam gedung.

5. *Hardware Design*

Hardware desain digunakan untuk menggambarkan gambaran awal dari alat secara menyeluruh. Dengan adanya gambaran awal, prinsip kerja dan komponen dari sistem dapat dirancang dengan jelas. Software yang digunakan untuk membuat desain dari sistem parkir dan bentuk prototype ini menggunakan *software* SketchUP 2023.

3.2.5 Pembuatan Alat

Pembuatan alat didasari oleh beberapa pertimbangan yang telah disebutkan dalam perancangan sistem sebelumnya seperti *context* diagram, data flow diagram, blok diagram, flowchart, dan *Hardware design*. Yang nantinya akan menghasilkan sebuah sistem monitoring area parkir dan pelacakan posisi kendaraan pada area parkir. Yang mana nantinya peneliti akan membuat kerangka dari prototype sistem tersebut sesuai dengan apa yang sudah didesain pada perancangan hardware desain, setelah kerangkanya jadi lanjut ke tahap perakitan komponen komponen yang digunakan dan pemasangan kedalam kerangka yang sudah dibuat. setelah itu melakukan pembuatan alur program dan kerja dari sistem sesuai dengan data flow diagram dan flowchart yang sudah di rancang pada tahap perancangan sistem.

3.2.6 Pengujian alat

Pada tahap ini alat yang dirancang akan diuji dan Analisa agar sesuai dengan rancangan sebelum alat diimplementasikan. Adapun tahap pengujian alat adalah sebagai berikut :

1. Pengujian terhadap push button, webcam, lcd 1 diluar, apakah push button berfungsi untuk mengirimkan perintah capture ke webcam dan apakah webcam berhasil untuk mendeteksi dan pembacaan nomor plat kendaraan pengujian ini menggunakan webcam untuk mecapture gambar dan diproses pembacaan plat setelah plat terbaca dikirim ke database dan lcd akan menampilkan sisa slot yang tersedia pada gedung berlantai 2 tersebut.

2. Pengujian terhadap DFPlayer dan speaker apakah komponen tersebut dapat memberikan informasi sesuai dengan slot yang masih kosong atau tidak. dan pengujian kejernihan dan volume dari speaker yang akan mengeluarkan informasi slot parkir masih kosong.
3. Pengujian terhadap website yang akan menjadi sebuah tempat pencarian posisi dari kendaraan. Apakah website dapat mencari posisi kendaraan berdasarkan nomor plat kendaraan mereka yang terdata ketika awal masuk pada area parkir.

3.2.7 Implementasi sistem

Implementasi sistem adalah sebuah hasil dari sistem yang telah dirancang dan dibangun siap untuk dioperasikan. Tujuan dari implementasi ini untuk melihat seberapa tepat alat yang dirancang apakah sistem dapat menyelesaikan sebuah permasalahan yang telah diidentifikasi sebelumnya. Implementasi sistem ini ditujukan pada sebuah gedung parkir yang bertingkat yang belum memiliki sistem informasi mengenai slot parkir yang tersedia dan sistem pencari posisi kendaraan parkir.

BAB IV

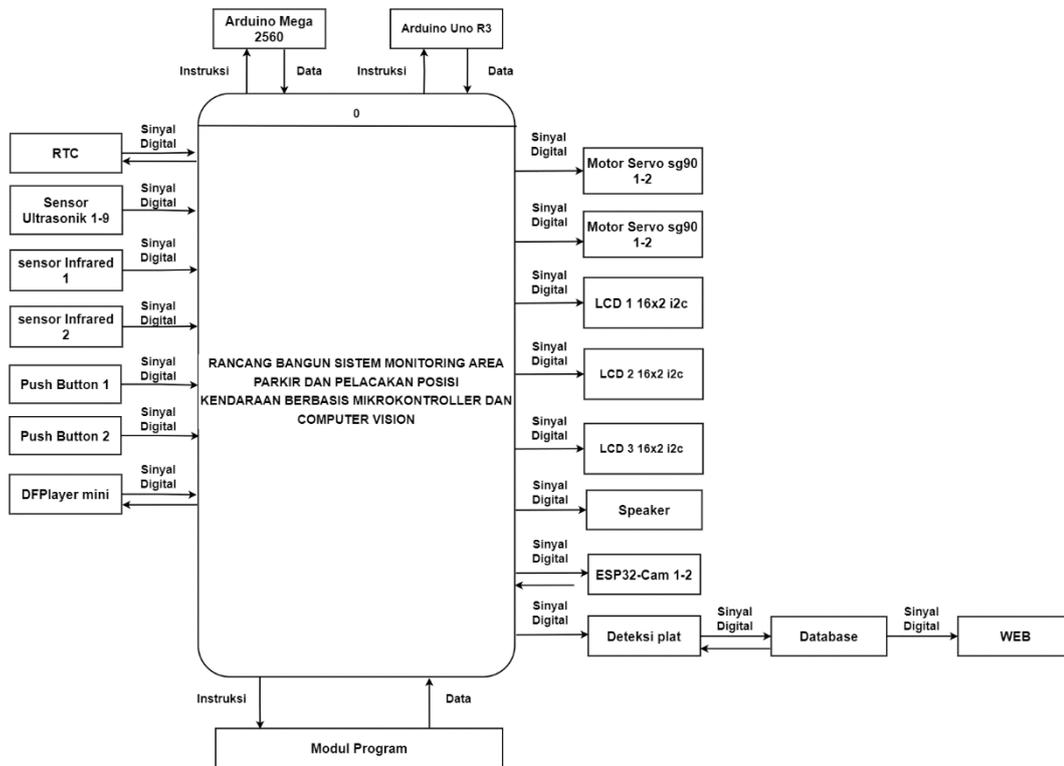
ANALISA DAN HASIL

4.1 Desain Sistem Secara Umum

Aturan dalam proses penganalisan perlu dilakukan pendefinisian terlebih dahulu terhadap sistem yang akan dirancang secara menyeluruh yaitu gambaran secara jelas mengenai ruang lingkup pembahasan masalah dimana medianya terdiri dari *context diagram*, *data flow diagram*, blok diagram, dan *flowchart*. Berikut untuk lebih jelasnya dapat dilihat pada uraian berikut :

4.1.1 Context Diagram

Context diagram merupakan pendefinisian rancangan yang bersifat menyeluruh. *Context diagram* memperlihatkan adanya batasan dan interaksi antara *entity* eksternal dengan sistem. Hal ini digunakan untuk memudahkan dalam proses penganalisaan sistem yang dirancang secara keseluruhan. *Context diagram* tersebut dapat dilihat pada Gambar 4.1.



Gambar 4.1 Context Diagram Sistem Parkir

Sistem ini berkomunikasi dengan beberapa *entity*, yaitu Arduino Mega 2560 , arduino Uno R3, sensor ultrasonic, sensor infrared, push button, motor servo sg90, LCD 16x2 i2c, speaker, DFPlayer mini, RTC, dan ESP32-Cam. Entity tersebut akan dibahas fungsinya sebagai berikut :

1. *Arduino Mega 2560*

Arduino Mega 2560 ini berfungsi sebagai tempat pusat pengolahan seluruh data dan instruksi induk selain penampilan no plat pada lantai 1 dan 2.

2. *Real Time Clock (RTC)*

Berfungsi sebagai penyimpan tanggal dan jam pada alat serta database.

3. *ultrasonik HC-SR04*

Berfungsi sebagai pendeteksi slot parkir berisi atau tidaknya.

4. *Infrared*

Berfungsi sebagai pendeteksi kendaraan melewati portal dan menutup portal otomatis.

5. Push button 1 (masuk)

Berfungsi sebagai pembuka portal masuk, pengaktifan DFPlayer mini, dan mencapture plat kendaraan yang akan masuk.

6. Push button 2 (keluar)

Berfungsi sebagai pembuka portal keluar dan mencapture plat kendaraan yang akan keluar.

7. DFPlayer mini

Berfungsi sebagai komponen yang menghubungkan komunikasi antara Arduino Mega 2560 dengan speaker.

8. Motor servo sg90

Berfungsi sebagai penggerak membuka dan menutup portal masuk dan keluar.

9. LCD 1 16x2 (gerbang masuk)

Berfungsi sebagai menampilkan informasi sisa slot parkir pada gedung parkir.

10. LCD 2 16x2 (lantai 1)

Berfungsi sebagai menampilkan no plat kendaraan yang parkir pada lantai 1.

11. LCD 3 16x2 (lantai 2)

Berfungsi sebagai menampilkan no plat kendaraan yang parkir pada lantai 2.

12. Speaker

Berfungsi sebagai keluaran berupa suara untuk memberikan informasi slot parkir yang kosong untuk ditempati oleh pengendara dan navigasi menuju tempat parkir.

13. ESP32-Cam

Berfungsi sebagai webcam yang nanti yang akan mencapture no plat kendaraan yang masuk dan keluar dari area parkir.

14. Python

Bahasa pemrograman ini nantinya yang akan menangkap atau mendeteksi no plat kendaraan dan sebagai mengirim data ke database dan mengirimkan ke Arduino mega 2560.

15. Database phpMyAdmin

Sebagai tempat menyimpan data waktu kedatangan, no plat, no slot, dan waktu keluar.

16. WEB

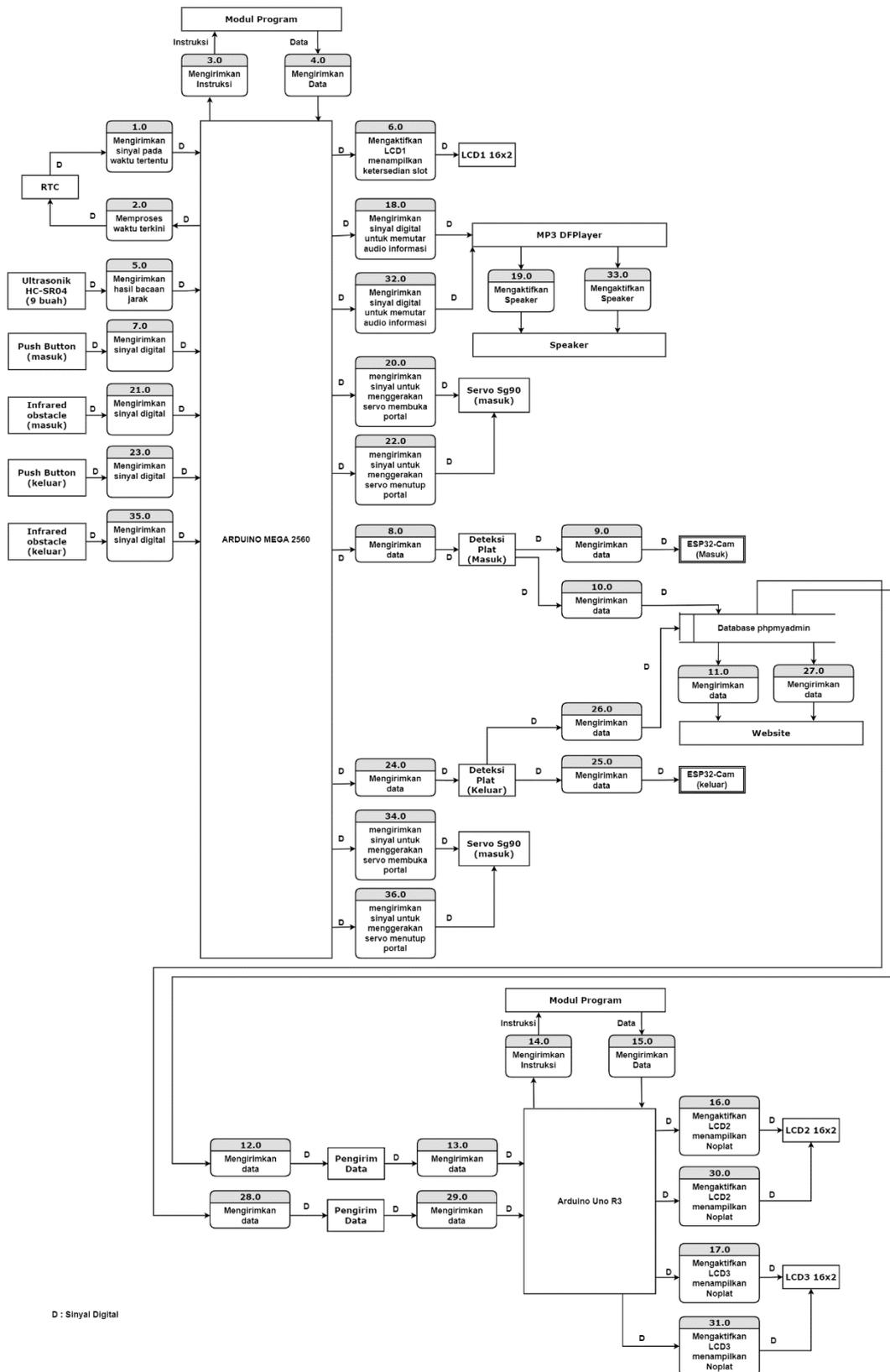
Berfungsi sebagai media untuk mencari posisi kendaraan yang parkir pada area parkir gedung.

17. Modul Program

Modul program menggunakan bahasa pemrograman Arduino untuk mengontrol semua proses yang akan terjadi pada sistem.

4.1.2 Data Flow Diagram (DFD)

Data flow diagram merupakan suatu cara atau metode yang digunakan untuk membuat rancangan sebuah sistem yang berorientasi pada alur data yang bergerak pada sebuah sistem. DFD menggambarkan arus data didalam sistem dengan terstruktur dan jelas. Berikut DFD dapat dilihat pada Gambar 4.2.



Gambar 4.2 DFD Sistem Parkir

Berdasarkan DFD Arduino uno R3 dapat dijelaskan bahwa sistem ini memiliki urutan proses instruksi sebagai berikut :

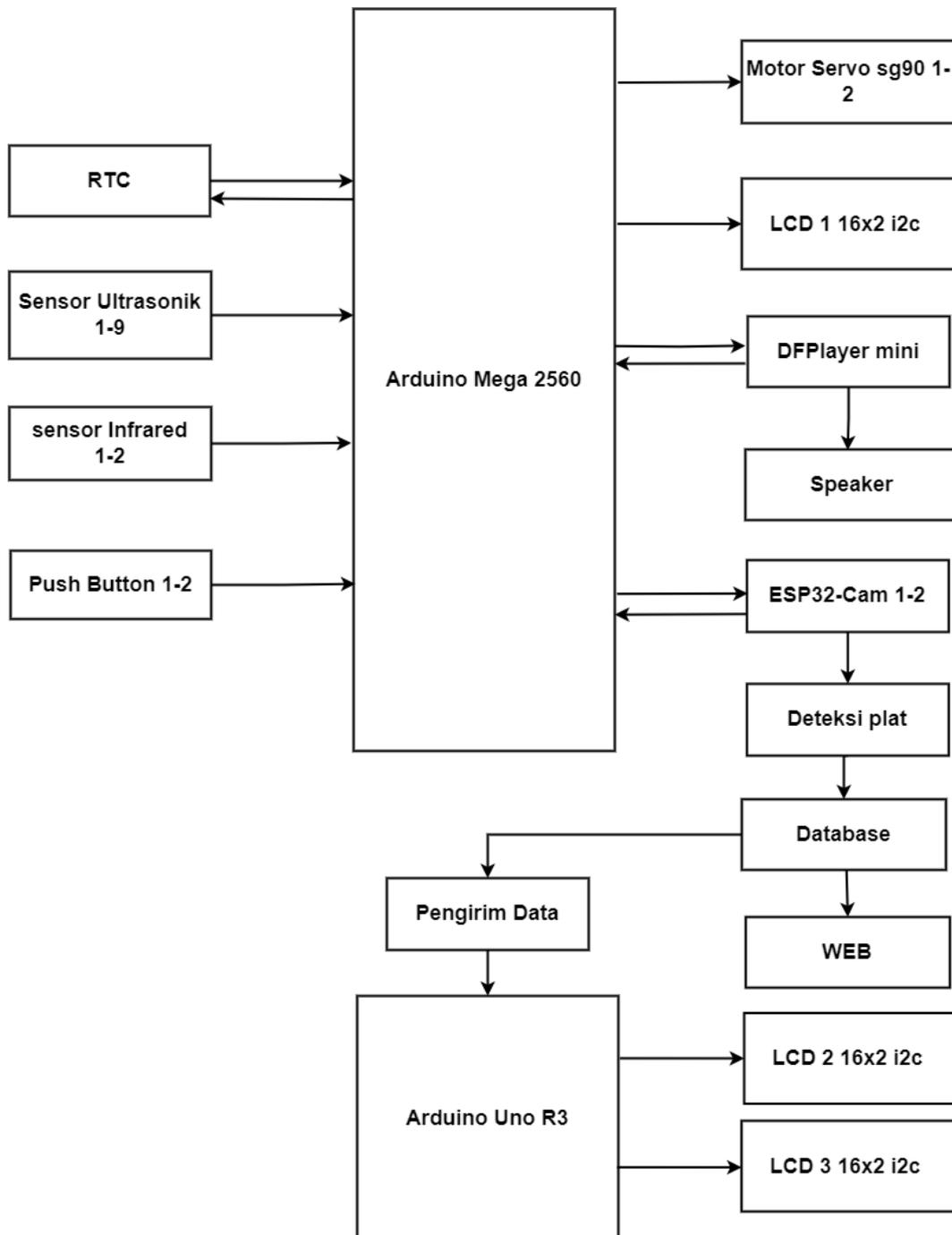
1. RTC mengirimkan sinyal waktu yang disimpan nya ke Arduino mega 2560(1.0), dan Arduino mega 2560 mengirimkan sinyal kembali untuk memproses waktu terkini ke RTC(2.0).
2. Arduino mega 2560 mengirimkan instruksi ke modul program untuk memproses data (3.0), hasil proses dari modul program dikirim kembali ke Arduino mega 2560 (4.0).
3. Ultrasonik mengirimkan hasil bacaan jaraknya ke Arduino mega 2560 (5.0).
4. Arduino mega 2560 mengirimkan kan data hasil bacaan tadi ke LCD 1 dan mengaktifkannya dan menampilkan ketersediaan slot pada gedung parkir (6.0).
5. Push button (masuk) mengirimkan sinyal digital ke Arduino mega 2560 (7.0).
6. Arduino mega 2560 mengirimkan data berupa tanggal, noslot dan perintah untuk mencapture gambar ke deteksi plat (8.0).
7. Deteksi plat akan mengirimkan data ke esp32-cam untuk menangkap gambar (9.0).
8. Deteksi plat akan mengirimkan data yang dia terima tadi ke database (10.0).
9. Database mengirimkan hasil data yang dia terima dari python ke Web(11.0).
10. Database mengirimkan data ke pemograman penerima data yang mana data yang dikirim berupa data no plat yang parkir pada area parkir (12.0).
11. Penerima data akan menerima data tersebut dan diolah untuk dikirimkan data tersebut ke Arduino Uno R3 (13.0).

12. Arduino uno R3 mengirimkan instruksi ke modul program (14.0), hasil dari modul program yang mengolah instruksi yang diterima tadi akan mengirimkan data kembali ke Arduino uno r3 (15.0).
13. Arduino uno r3 akan mengirimkan data ke LCD 2 untuk menampilkan data no plat yang parkir pada lantai 1 (16.0).
14. Arduino uno r3 akan mengirimkan data ke LCD 3 untuk menampilkan data no plat yang parkir pada lantai 2 (17.0).
15. Arduino mega 2560 mengirimkan sinyal untuk menginstruksikan DFPlayer untuk memutar audio informasi slot parkir (18.0).
16. DFPlayer mengaktifkan speaker untuk mengeluarkan suara audio (19.0).
17. Arduino mega 2560 mengirimkan sinyal untuk menggerakkan servo untuk membuka portal masuk (20.0).
18. Infrared obstacle (masuk) mengirimkan sinyal digital ke Arduino mega 2560 (21.0).
19. Arduino meneruskan sinyal dari infrared untuk menggerakkan servo untuk menutup portal masuk (22.0).
20. Push button (keluar) mengirimkan sinyal digital ke Arduino mega 2560 (23.0).
21. Arduino mega 2560 mengirimkan data berupa tanggal dan perintah untuk mencapture gambar ke deteksi plat (24.0).
22. Deteksi plat akan mengirimkan data ke esp32-cam untuk menangkap gambar (25.0).
23. Deteksi plat akan mengirimkan data yang dia terima tadi ke database (26.0).

24. Database mengirimkan hasil data yang dia terima dari python ke Web(27.0).
25. Database mengirimkan data ke pemograman penerima data yang mana data yang dikirim berupa data no plat yang parkir pada area parkir (28.0).
26. Penerima data akan menerima data tersebut dan diolah untuk dikirimkan data tersebut ke Arduino Uno R3 (29.0).
27. Arduino uno r3 akan mengirimkan data ke LCD 2 untuk menampilkan data no plat yang parkir pada lantai 1 (30.0).
28. Arduino uno r3 akan mengirimkan data ke LCD 3 untuk menampilkan data no plat yang parkir pada lantai 2 (31.0).
29. Arduino mega 2560 mengirimkan sinyal untuk menginstruksikan DFPlayer untuk memutar audio informasi slot parkir (32.0).
30. DFPlayer mengaktifkan speaker untuk mengeluarkan suara audio (33.0).
31. Arduino mega 2560 mengirimkan sinyal untuk menggerakkan servo untuk membuka portal keluar (34.0).
32. Infrared obstacle (keluar) mengirimkan sinyal digital ke Arduino mega 2560 (35.0).
33. Arduino meneruskan sinyal dari infrared untuk menggerakkan servo untuk menutup portal keluar (36.0).

4.1.3 Block diagram

Dengan mengacu pada Data Flow Diagram Gambar 4.3 penggunaan komponen-komponen pada alat dapat dilihat dalam blok diagram seperti Gambar 4.3.



Gambar 4.3 Blok Diagram Sistem Parkir

Dari blok diagram diatas dapat dilihat bahwa sistem terdiri dari *input*, *process*, dan *output*. Mikrokontroller Arduino Mega 2560 berperan sebagai pusat

pemrosesan dan pengontrolan. RTC, Sensor Ultrasoni 9 buah, sensor infrared 2 buah, push button 2 buah sebagai *input*. Motor servo sg90 2 buah, LCD 1 16x2 i2c, DFPlayer mini mp3, Speaker, ESP32-Cam 2 buah dan WEB sebagai *output*. Dari blok diagram diatas dapat dilihat bahwa sistem terdiri dari *input*, *process*, dan *output*. Mikrokontroller Arduino Uno R3 berperan sebagai pusat pemrosesan dan pengontrolan. Database sebagai *input* dan python sebagai proses pengirim dan pengambilan data untuk ke Arduino Uno R3. LCD 2 16x2 i2c dan LCD 3 16x2 i2c sebagai *output*.

4.2 Prinsip Kerja Sistem

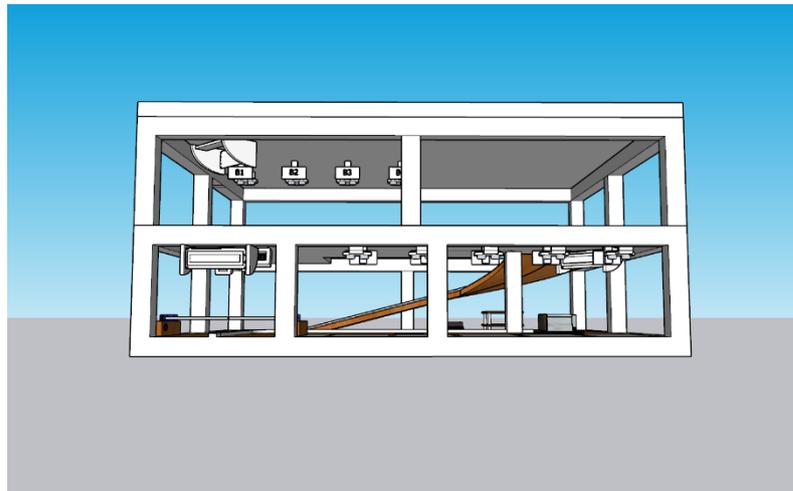
Adapun prinsip dan sistem kerja dari alat sistem monitoring area parkir dan pelacakan posisi kendaraan menggunakan *computer vision* ini adalah sebagai berikut :

1. Di pintu masuk terdapat portal masuk yang digerakan oleh servo, ketika pengandara menekan push button itu akan mengirimkan perintah ke program mendeteksi plat kendaraan, dan akan mengcapture gambarnya dan setelah gambar dicapture, gambar akan diproses dan hasil bacaan gambar akan dikirimkan ke database. Dan diwaktu yang sama DFPlayer mini akan aktif dan memutar audio informasi dimana slot parkir yang akan diisi berdasarkan slot parkir yang kosong terdekat. Dan setelah informasi disampaikan portal akan terbuka.
2. Jika kendaraan sudah melewati sensor infrared portal akan tertutup otomatis.

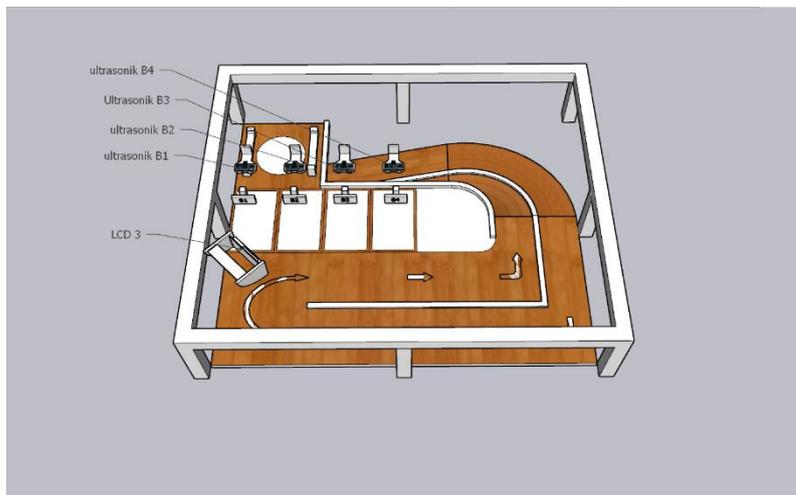
3. Ultrasonik akan mendeteksi apakah sudah ada kendaraan menempati atau belum dengan menghitung jarak antara slot parkir dengan sensor.
4. Setelah menempati slot parkir lcd 1 akan menampilkan sisa slot parkir yang tersedia pada gedung parkir.
5. Lcd kedua dan ketiga yang berada di lantai 1 dan 2 akan menampilkan no plat kendaraan yang telah memasuki area parkir tersebut.
6. Jika pengemudi akan keluar dia akan menekan push button untuk membuka portal dan mengirimkan perintah sama seperti pintu masuk dia akan mencapture gambar dan mendeteksi plat kendaraan dan setelah itu portal terbuka , dan setelah kendaraan melewati infrared portal akan tertutup otomatis.
7. Data-data yang dikirimkan ke database akan dikirimkan ke website dan website akan menampilkan seluruh data kendaraan yang parkir pada gedung parkir.

4.3 Rancangan Fisik Sistem

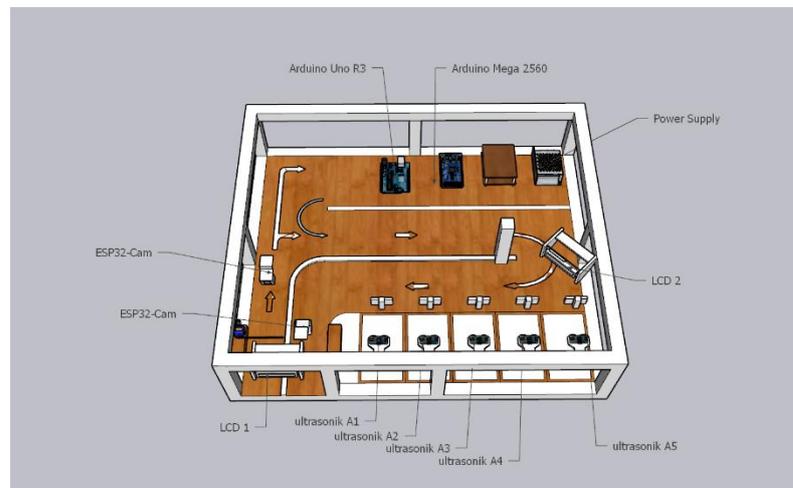
Perancangan sistem ini merupakan tahap awal dari pemasangan dan menganalisa permasalahan yang dihadapi berdasarkan literatur, hal ini dilakukan untuk memberikan gambaran jelas tentang alat yang akan dibuat sehingga tidak akan terjadi kesalahan ketika membuatnya. Rancangan fisik dari sistem ini didesain menggunakan *software* Sketchup 2023. Tampilan lebih jelasnya dapat dilihat pada Gambar 4.4.



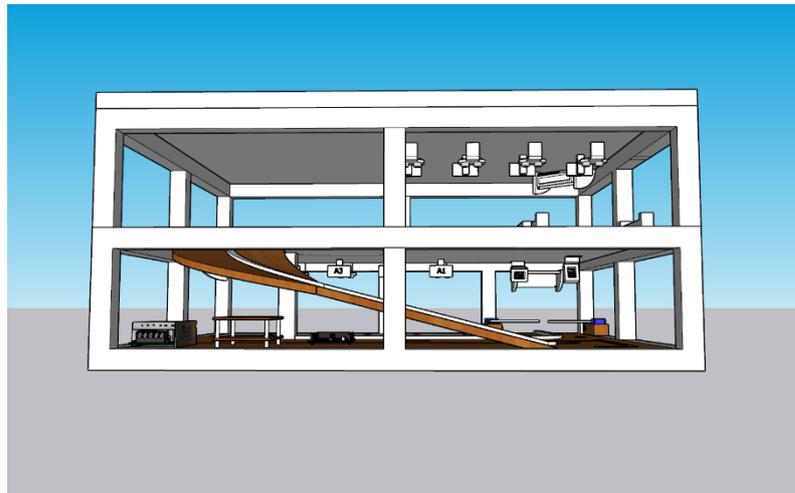
(a) Rancangan Tampilan Depan Full Gedung



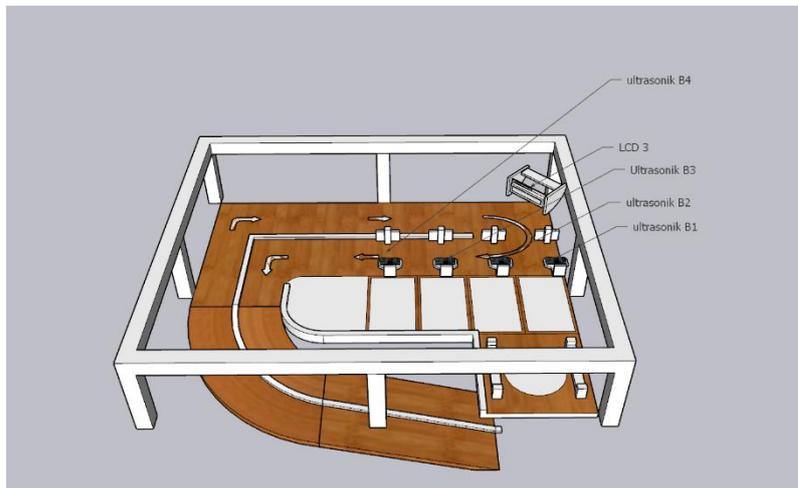
(b) Rancangan Tampilan Depan Lantai 2



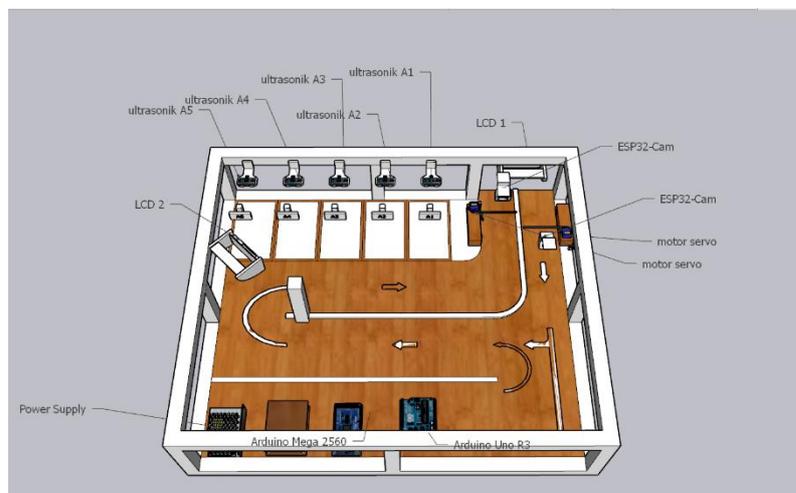
(c) Rancangan Tampilan Depan Lantai 1



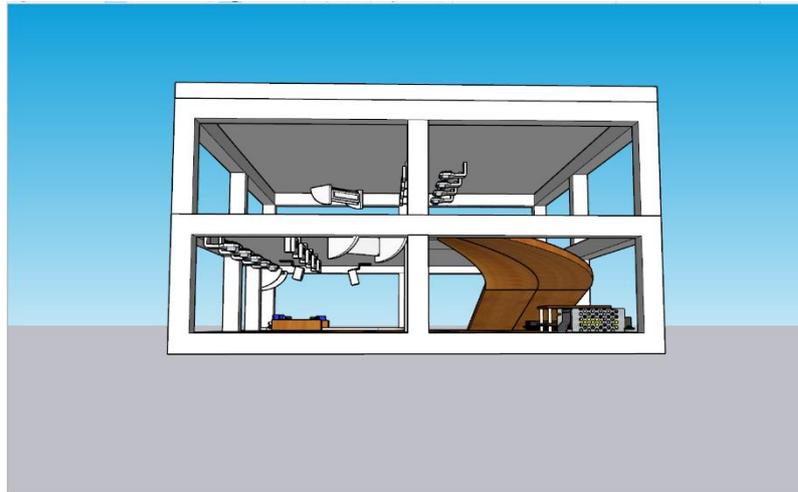
(d) Rancangan Tampilan Belakang Full Gedung



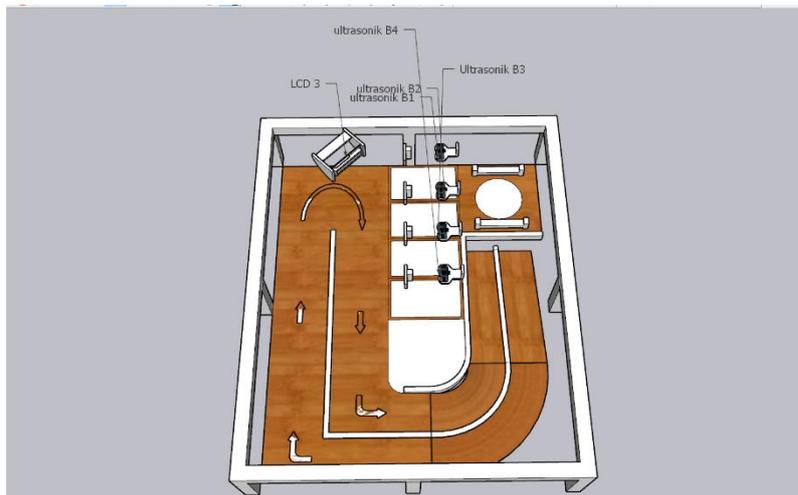
(e) Rancangan Tampilan Belakang Lantai 2



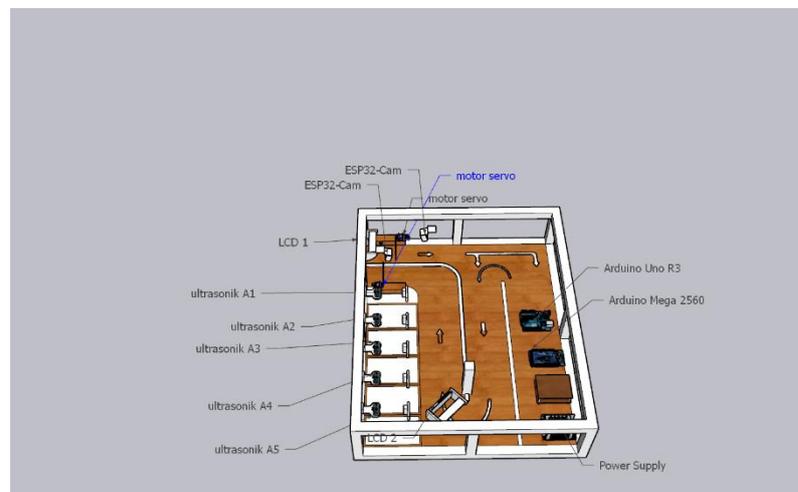
(f) Rancangan Tampilan Belakang Lantai 1



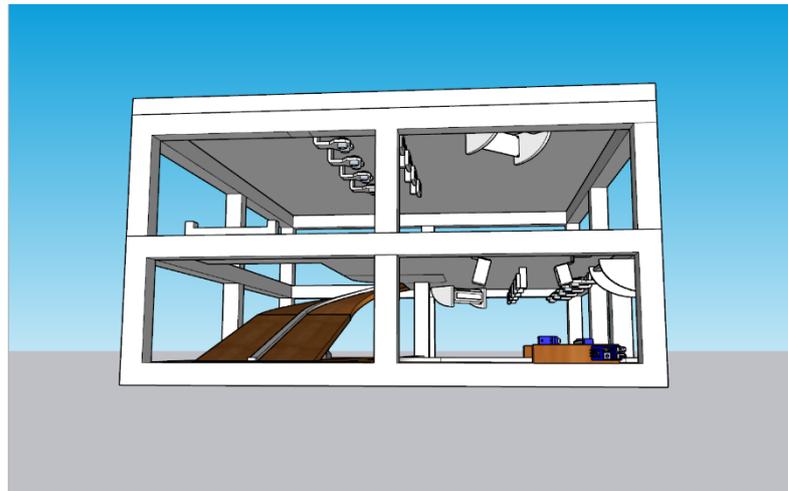
(g) Rancangan Tampilan Kiri Full Gedung



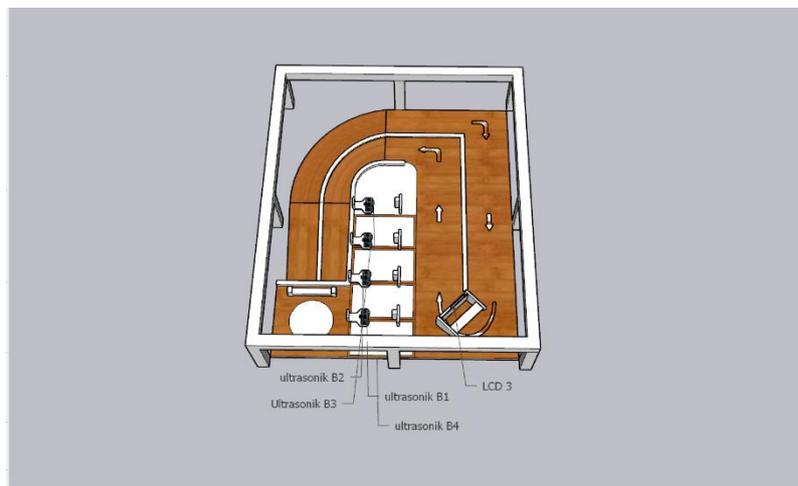
(h) Rancangan Tampilan Kiri Lantai 2



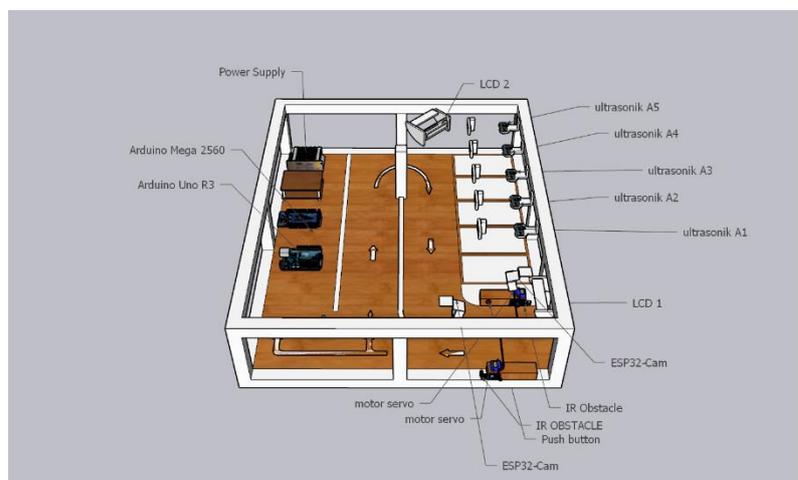
(i) Rancangan Tampilan Kiri Lantai 1



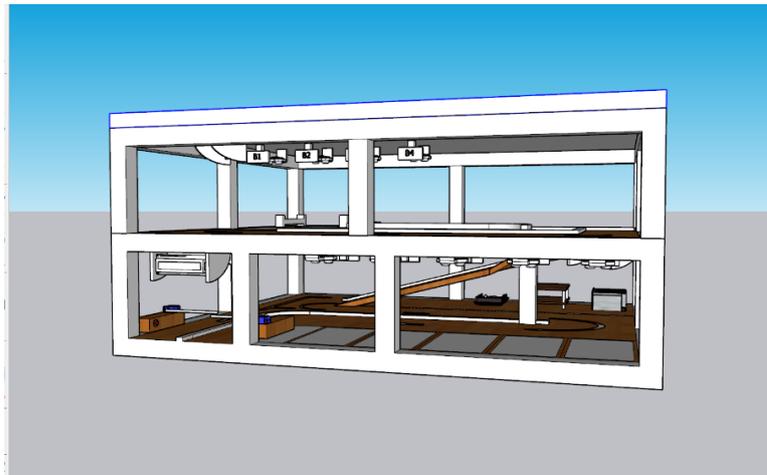
(j) Rancangan Tampilan Kanan Full Gedung



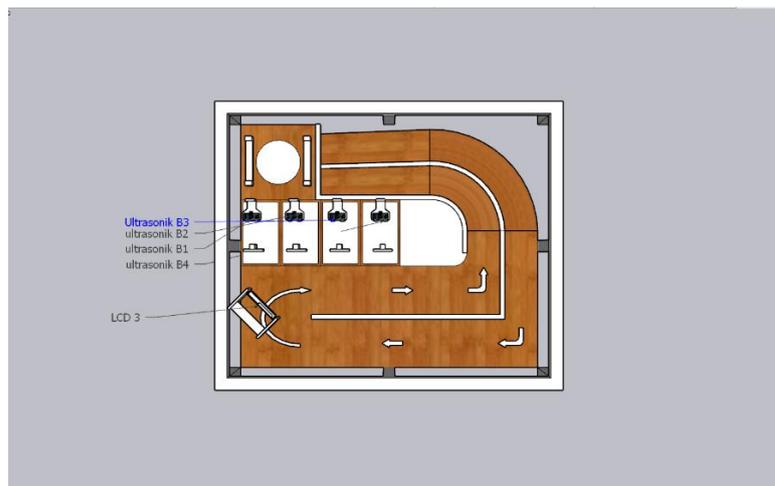
(k) Rancangan Tampilan Kanan Lantai 2



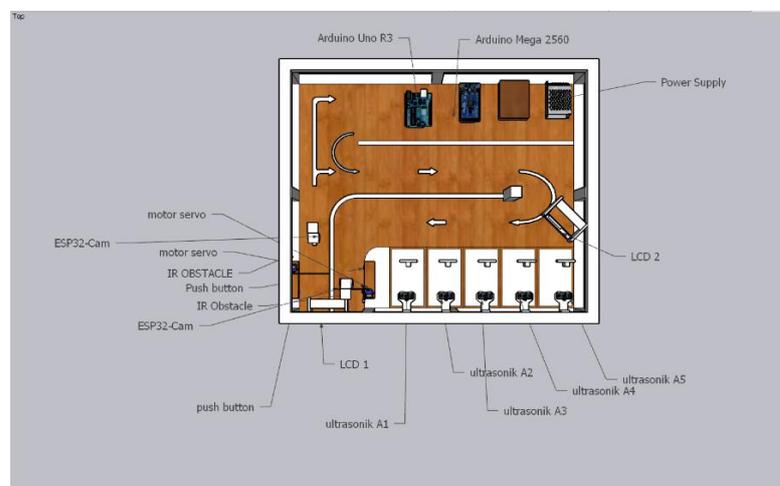
(l) Rancangan Tampilan Kanan Lantai 1



(m) Rancangan Tampilan Iso Full Gedung



(n) Rancangan Tampilan Atas Lantai 2



(o) Rancangan Tampilan Atas Lantai 2

Gambar 4.4 Desain Rancangan Fisik

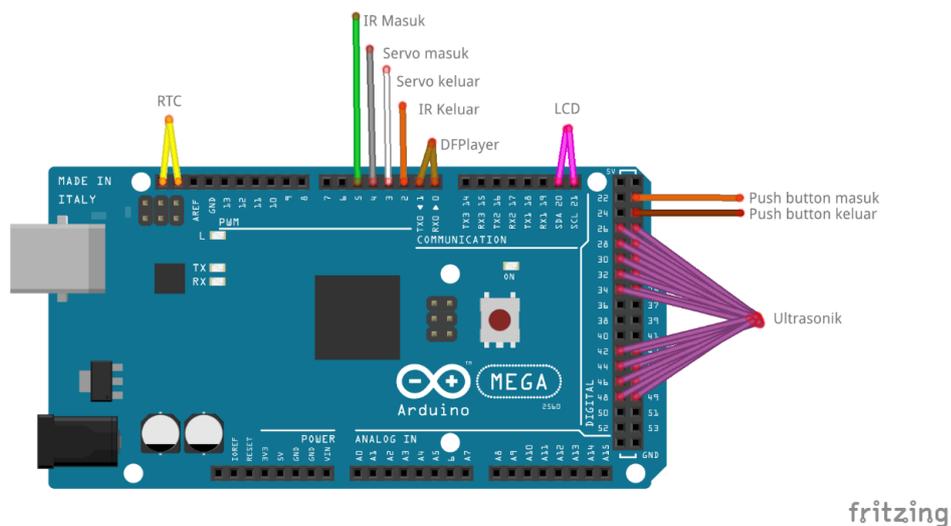
Miniatur ini buat seperti sebuah gedung parkir yang berlantai dua yang didalamnya terdapat pengontrol Arduino mega 2560, Arduino uno R3, sensor *ultrasonic*, LCD 16x2, sensor *infrared*, *push button*, motor servo, RTC, DFPlayer mini, *power supply*, dan ESP32-Cam.

4.4 Desain Sistem Terperinci

Desain pada sistem merupakan gambaran dari sistem yang dirancang dan dibuat. Dengan adanya desain sistem secara terperinci, maka prinsip serta *entity* dari sistem dapat dilihat dengan sangat jelas.

4.4.1 Desain Rangkaian Arduino Mega 2560

Pada sistem ini mikrokontroler Arduino mega 2560 digunakan sebagai pengendali dari sistem. Arduino mega 2560 menggunakan chip ATmega 2560 yang memiliki pin I/O digital 54 buah (15 pin PWM), 16 pin analog *input* dan 4 pin UART. Rangkaian Arduino mega 2560 dapat dilihat pada Gambar 4.5.



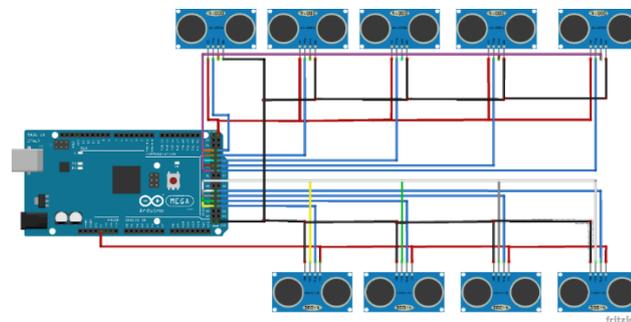
Gambar 4.5 Desain Rangkaian Arduino Mega 2560

Penjelasan pin yang digunakan pada rangkaian Arduino mega 2560 sebagai berikut :

1. Pin 23,25 arduino mega 2560 dihubungkan ke push button.
2. Pin 26-35 dan 42-49 arduino mega 2560 dihubungkan ke ultrasonik hc-sr04.
3. Pin 20,21 arduino mega 2560 dihubungkan ke LCD 16x2 i2c.
4. Pin 0,1 arduino mega 2560 dihubungkan ke DFPlayer mini mp3.
5. Pin 2 arduino mega 2560 dihubungkan ke sensor IR keluar.
6. Pin 3 arduino mega 2560 dihubungkan ke motor servo sg90 keluar.
7. Pin 4 arduino mega 2560 dihubungkan ke motor servo sg90 Masuk.
8. Pin 5 arduino mega 2560 dihubungkan ke sensor IR masuk.
9. Pin SDA SCL Arduino mega 2560 dihubungkan ke RTC DS3231.

4.4.2 Desain Rangkaian Ultrasonic

Pada perancangan alat ini menggunakan sensor ultrasonic HC-SR04 sebagai *input* yang akan nantinya akan mendeteksi jarak pada setiap slot parkir untuk desain rangkaiannya dapat dilihat pada Gambar 4.6.



Gambar 4.6 Desain Rangkaian Sensor Ultrasonik HC-SR04

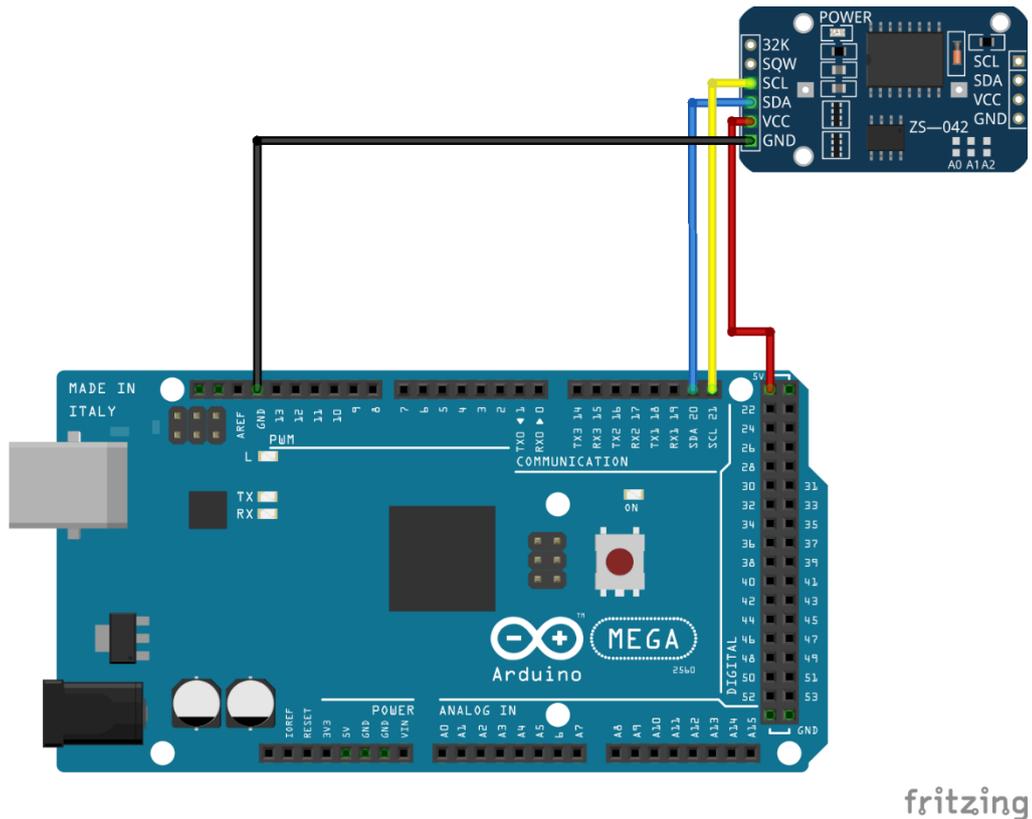
Penjelasan pin yang digunakan setiap sensor ultrasonic yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND semua sensor ultrasonic dihubungkan pada pin GND Arduino mega 2560.
2. Pin VCC semua sensor ultrasonic dihubungkan pada pin 5v Arduino mega 2560.
3. Pin trig sensor ultrasonik 1 dihubungkan pada pin digital 27 pada Arduino mega 2560.
4. Pin echo sensor ultrasonik 1 dihubungkan pada pin digital 26 pada Arduino mega 2560.
5. Pin trig sensor ultrasonik 2 dihubungkan pada pin digital 29 pada Arduino mega 2560.
6. Pin echo sensor ultrasonik 2 dihubungkan pada pin digital 28 pada Arduino mega 2560.
7. Pin trig sensor ultrasonik 3 dihubungkan pada pin digital 31 pada Arduino mega 2560.
8. Pin echo sensor ultrasonik 3 dihubungkan pada pin digital 30 pada Arduino mega 2560.
9. Pin trig sensor ultrasonik 4 dihubungkan pada pin digital 33 pada Arduino mega 2560.
10. Pin echo sensor ultrasonik 4 dihubungkan pada pin digital 32 pada Arduino mega 2560.

11. Pin trig sensor ultrasonik 5 dihubungkan pada pin digital 35 pada Arduino mega 2560.
12. Pin echo sensor ultrasonik 5 dihubungkan pada pin digital 34 pada Arduino mega 2560.
13. Pin trig sensor ultrasonik 6 dihubungkan pada pin digital 43 pada Arduino mega 2560.
14. Pin echo sensor ultrasonik 6 dihubungkan pada pin digital 42 pada Arduino mega 2560.
15. Pin trig sensor ultrasonik 7 dihubungkan pada pin digital 45 pada Arduino mega 2560.
16. Pin echo sensor ultrasonik 7 dihubungkan pada pin digital 44 pada Arduino mega 2560.
17. Pin trig sensor ultrasonik 8 dihubungkan pada pin digital 47 pada Arduino mega 2560.
18. Pin echo sensor ultrasonik 8 dihubungkan pada pin digital 46 pada Arduino mega 2560.
19. Pin trig sensor ultrasonik 9 dihubungkan pada pin digital 49 pada Arduino mega 2560.
20. Pin echo sensor ultrasonik 9 dihubungkan pada pin digital 48 pada Arduino mega 2560.

4.4.3 Desain Rangkaian RTC

Pada perancangan alat ini menggunakan *Real Time Clock (RTC) DS3231* sebagai penyimpanan waktu dan tanggal untuk desain rangkaianannya dapat dilihat pada Gambar 4.7.



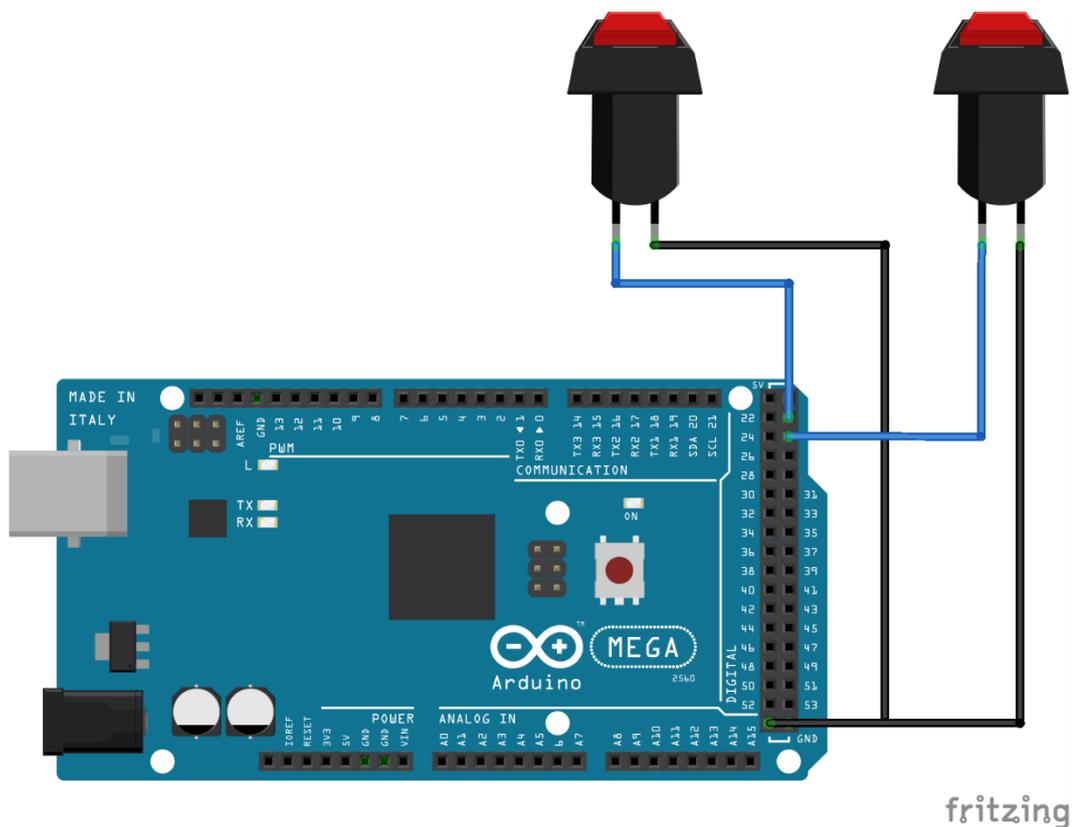
Gambar 4.7 Desain Rangkaian RTC DS3231

Penjelasan pin yang digunakan pada RTC DS3231 yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND RTC DS3231 dihubungkan pada pin GND Arduino mega 2560.
2. Pin VCC RTC DS3231 dihubungkan pada pin 5v Arduino mega 2560.
3. Pin SDA RTC DS3231 dihubungkan pada pin SDA Arduino mega 2560.
4. Pin SCL RTC DS3231 dhubungkan pada pin SCL Arduino mega 2560.

4.4.4 Desain Rangkaian Push Button

Pada perancangan alat ini menggunakan 2 push button sebagai *input* untuk desain rangkaiannya dapat dilihat pada Gambar 4.8.



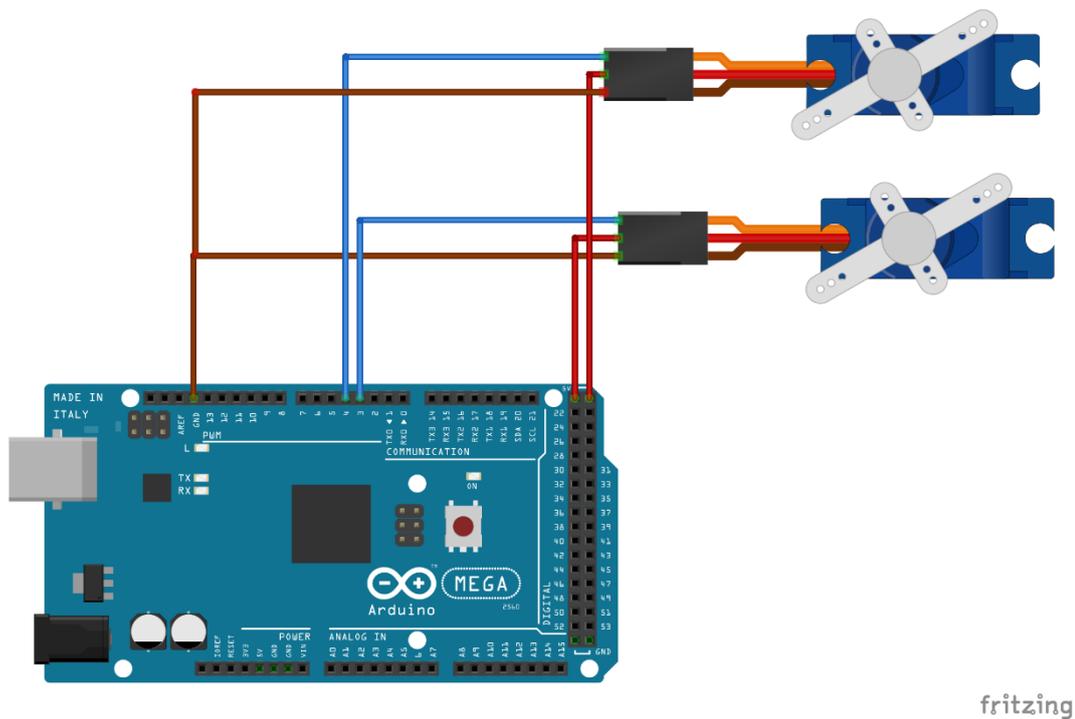
Gambar 4.8 Desain Rangkaian Push Button

Penjelasan pin yang digunakan pada 2 push button yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND kedua push button dihubungkan pada pin GND Arduino mega 2560.
2. Kaki push button 1 dihubungkan pada pin digital 25 Arduino mega 2560.
3. Kaki push button 2 dihubungkan pada pin digital 23 Arduino mega 2560.

4.4.5 Desain rangkaian motor servo

Pada perancangan alat ini menggunakan 2 motor servo sg90 sebagai *output* yang nantinya akan menggerakkan portal pintu masuk dan keluar untuk desain rangkaiannya dapat dilihat pada Gambar 4.9.



Gambar 4.9 Desain Rangkaian Motor Servo Sg90

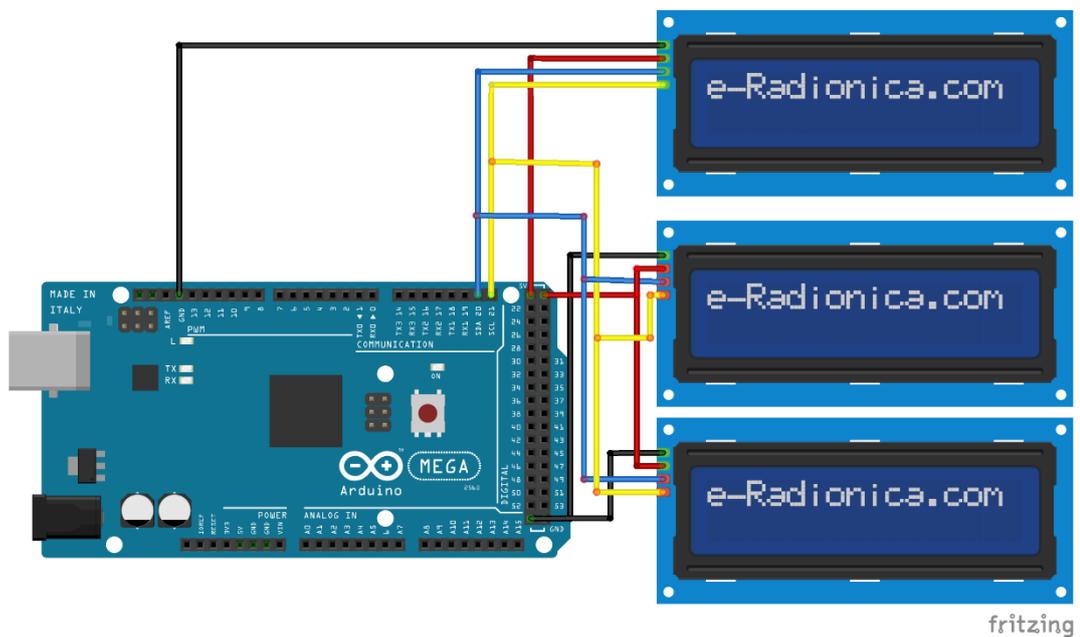
Penjelasan pin yang digunakan pada RTC DS3231 yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND kedua motor servo sg90 dihubungkan pada pin GND Arduino mega 2560.
2. Pin VCC kedua motor servo sg90 dihubungkan pada pin 5v Arduino mega 2560.

3. Pin output motor servo sg90 masuk dihubungkan pada pin digital 4 Arduino mega 2560.
4. Pin output motor servo sg90 keluar dihubungkan pada pin digital 3 Arduino mega 2560.

4.4.6 Desain Rangkaian LCD

Pada perancangan alat ini menggunakan 3 LCD 16x2 I2C sebagai *output* yang nantinya akan menampilkan informasi slot parkir tersedia dan informasi no plat kendaraan yang parkir pada lantai 1 dan 2 dan alamat tiap LCD berbeda yaitu 0x27, 0x26, 0x25 untuk desain rangkaiannya dapat dilihat pada Gambar 4.10.



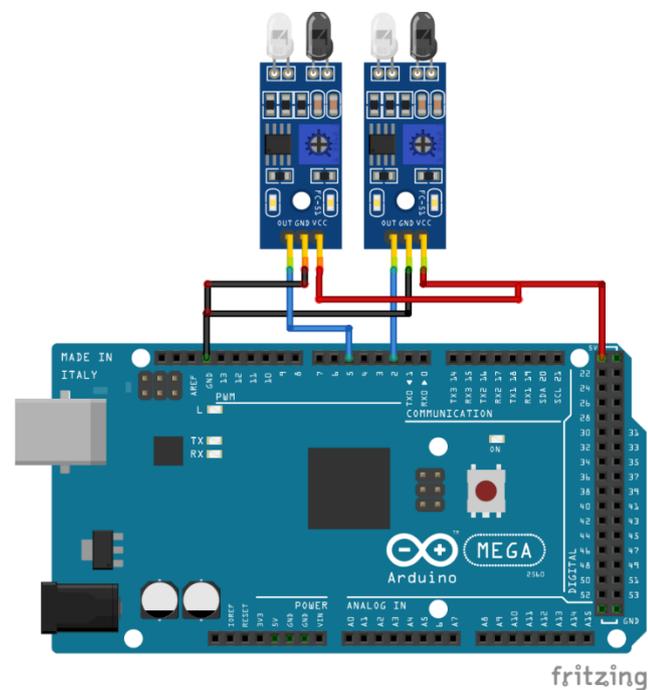
Gambar 4.10 Desain Rangkaian 3 LCD 16x2 I2C

Penjelasan pin yang digunakan pada 3 LCD 16x2 I2C yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND semua LCD 16x2 I2C dihubungkan pada pin GND Arduino mega 2560.
2. Pin VCC semua LCD 16x2 I2C dihubungkan pada pin 5v Arduino mega 2560.
3. Pin SDA semua LCD 16x2 I2C dihubungkan pada pin SDA Arduino mega 2560.
4. Pin SCL semua LCD 16x2 I2C dhubungkan pada pin SCL Arduino mega 2560.

4.4.7 Desain Rangkaian Sensor Infrared

Pada perancangan alat ini menggunakan 2 sensor infrared obstacle sebagai *input* untuk desain rangkaiannya dapat dilihat pada Gambar 4.11.



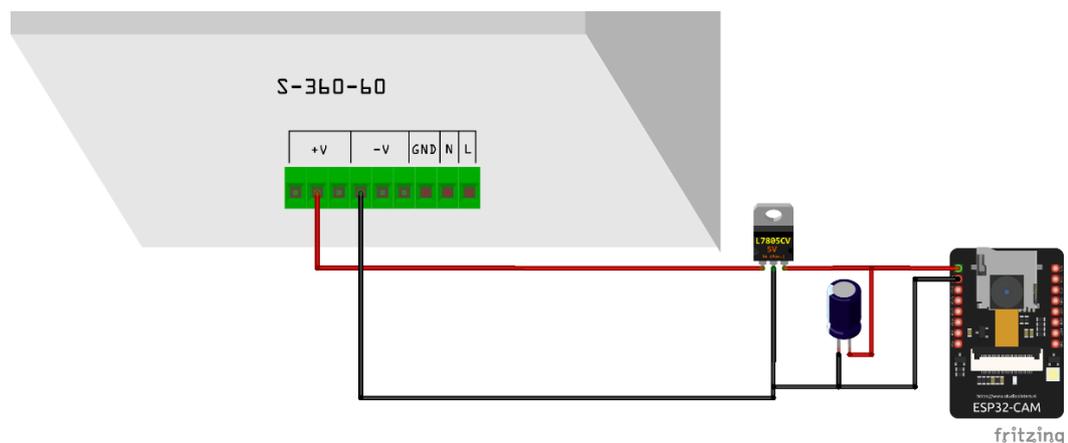
Gambar 4.11 Desain Rangkaian Sensor Infrared Obstacle

Penjelasan pin yang digunakan pada 2 sensor infrared obstacle yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND 2 sensor infrared obstacle dihubungkan pada pin GND Arduino mega 2560.
2. Pin VCC 2 sensor infrared obstacle dihubungkan pada pin 5v Arduino mega 2560.
3. Pin output sensor infrared obstacle masuk dihubungkan pada pin digital 5 Arduino mega 2560.
4. Pin output sensor infrared obstacle keluar dhubungkan pada pin digital 2 Arduino mega 2560.

4.4.8 Desain Rangkaian ESP32-Cam

Pada perancangan alat ini menggunakan 2 ESP32-Cam sebagai kamera untuk mencapture gambar untuk desain rangkaiannya dapat dilihat pada Gambar 4.12.



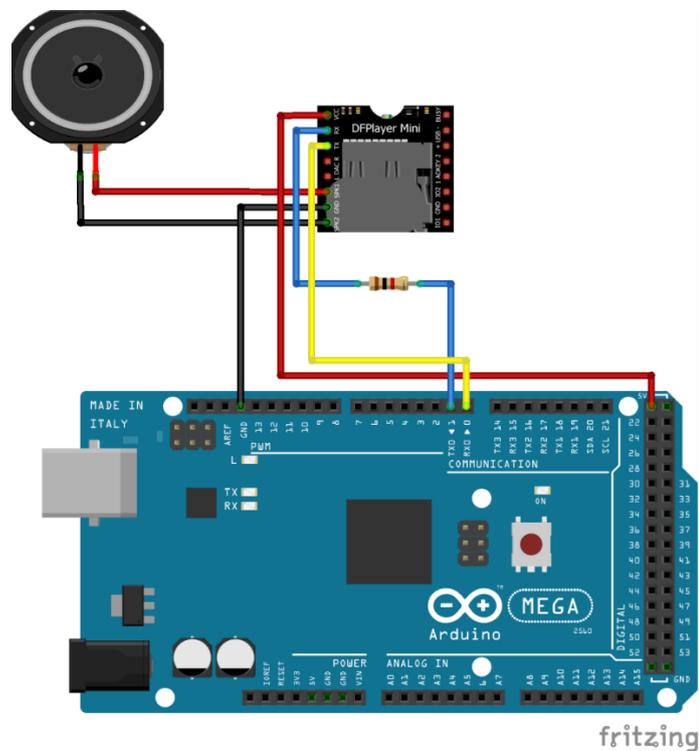
Gambar 4.12 Desain Rangkaian ESP32-Cam

Penjelasan pin yang digunakan pada 2 sensor infrared obstacle yang terhubung ke Arduino mega 2560 sebagai berikut :

1. Pin GND ESP32-Cam dihubungkan pada pin GND ic7805.
2. Pin VCC ESP32-Cam dihubungkan pada pin 5v ic7805.
3. Kaki input ic7805 dihubungkan dengan power supply 12v.
4. Kaki GND ic7805 dihubungkan dengan power supply GND.

4.4.9 Desain Rangkaian DFPlayer

Perancangan alat ini menggunakan speaker sebagai *output* suara pada sistem seperti informasi slot parkir dan navigasi ketempat slot parkir untuk desain rangkainnya dapat dilihat pada gambar 4.13.



Gambar 4.13 Desain Rangkaian DFPlayer Mini Mp3

Penjelasan pin yang digunakan pada DFPlayer mini mp3 yang terhubung ke Arduino mega 2560 sebagai berikut :

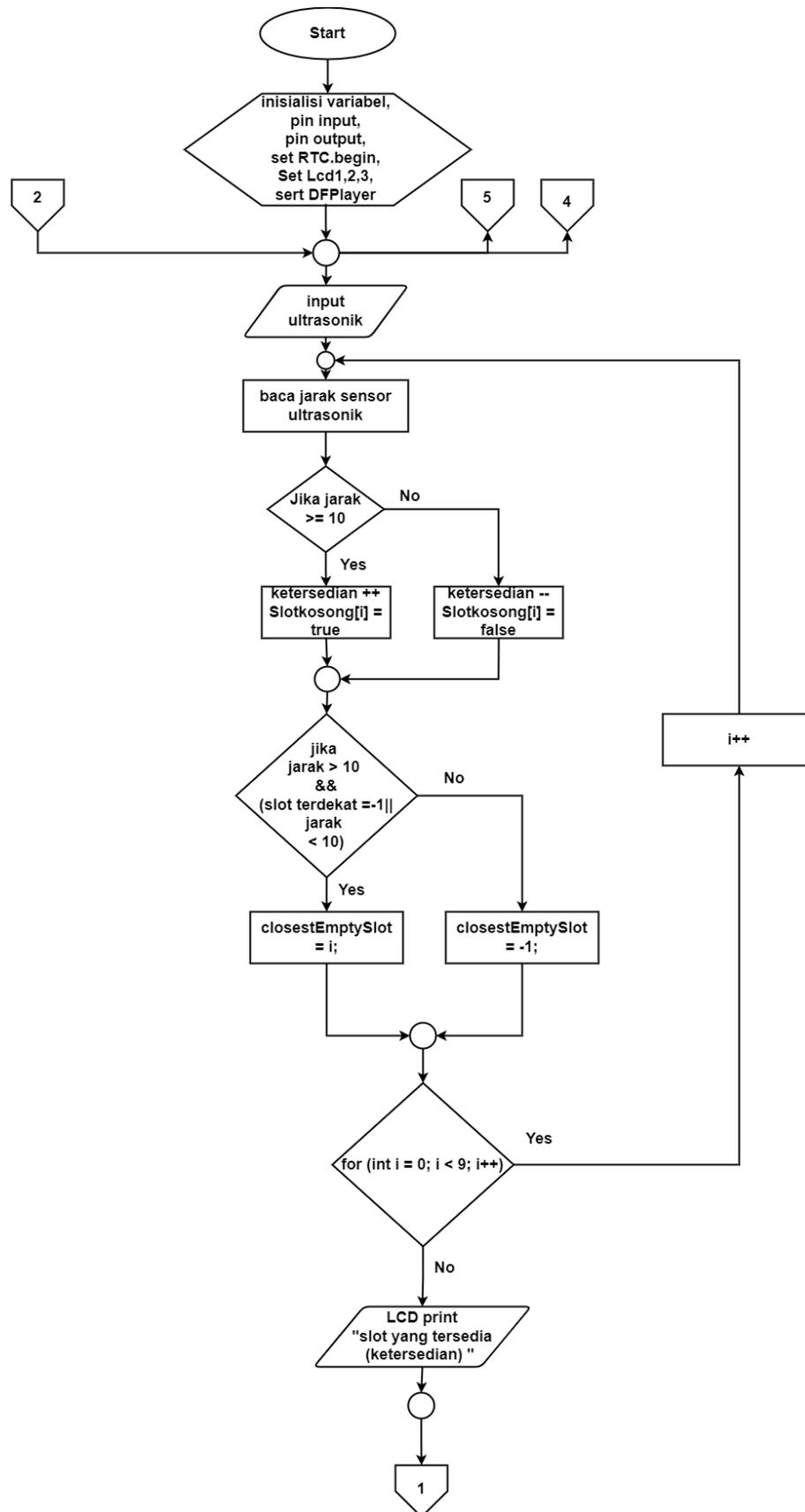
1. Pin GND 2 DFPlayer mini mp3 dihubungkan pada pin GND Arduino mega 2560.
2. Pin VCC 2 DFPlayer mini mp3 dihubungkan pada pin 5v Arduino mega 2560.
3. Pin RX DFPlayer mini mp3 dihubungkan pada pin digital 1 (TX) Arduino mega 2560 ditambah dengan resistor 1k ohm.
4. Pin TX DFPlayer mini mp3 dihubungkan pada pin digital 0 (RX) Arduino mega 2560.
5. Pin SPK 1 DFPlayer mini mp3 dihubungkan pada pin positif speaker.
6. Pin SPK 2 DFPlayer mini mp3 dihubungkan pada pin negatif speaker.

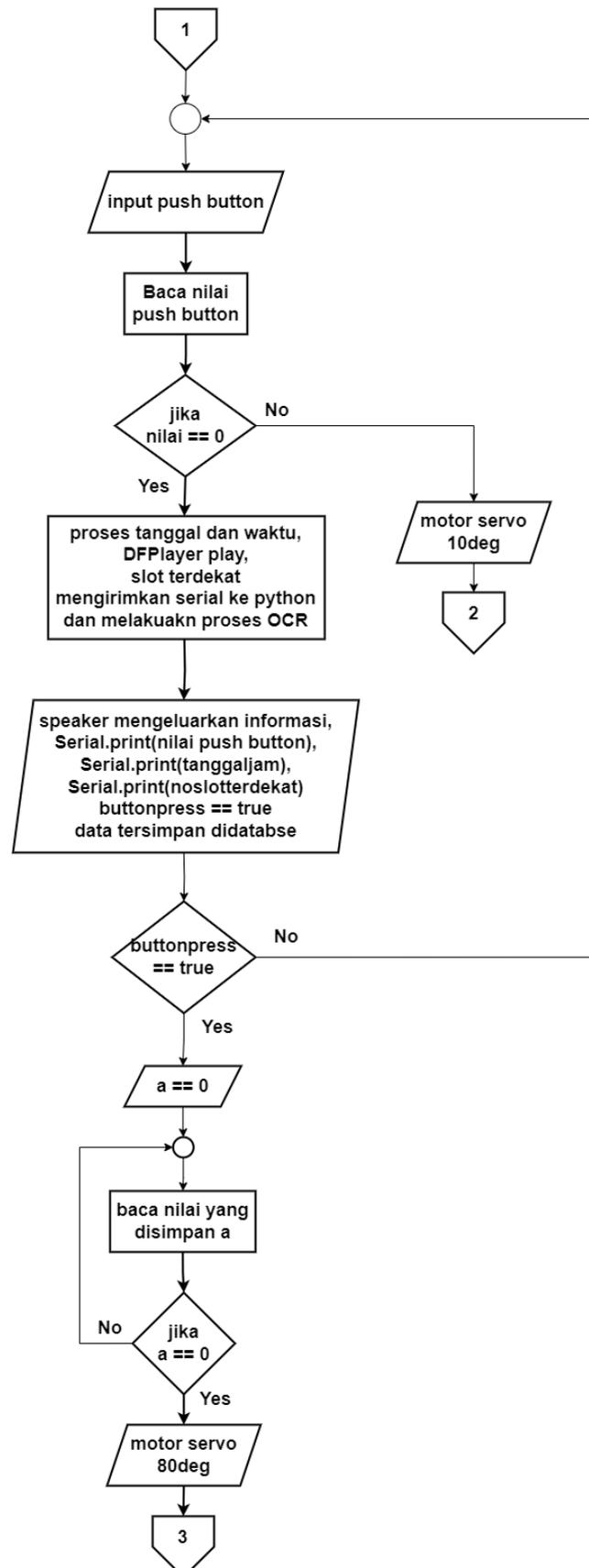
4.5 Rancangan Modul Program

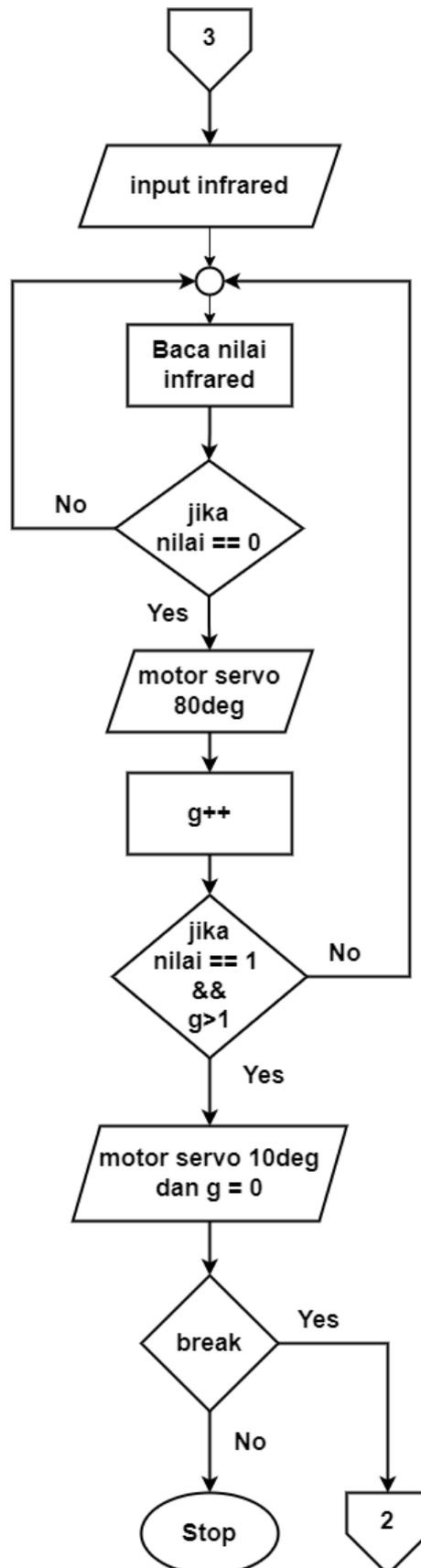
Sub bab ini menjelaskan tentang modul program yang digunakan untuk mengontrol kerja dari sistem yang dirancang. Untuk lebih mudah dimengerti rancangan modul dibagi menjadi 2 bagian yaitu *Flowchart* dan modul program.

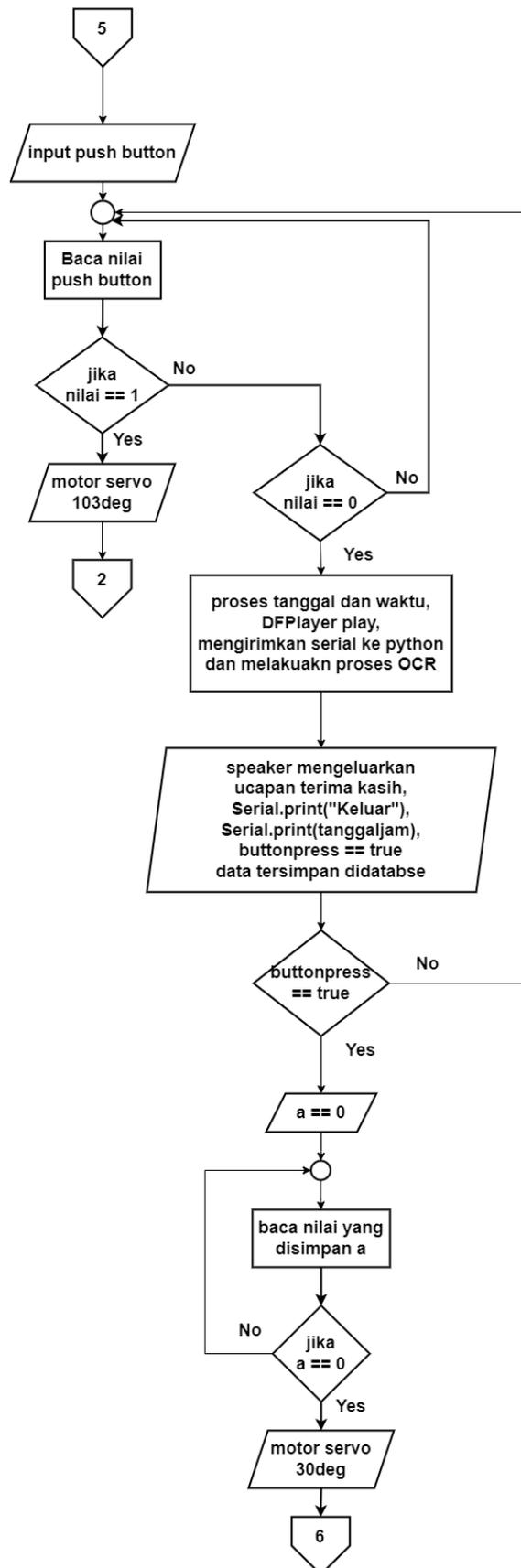
4.5.1 Flowchart

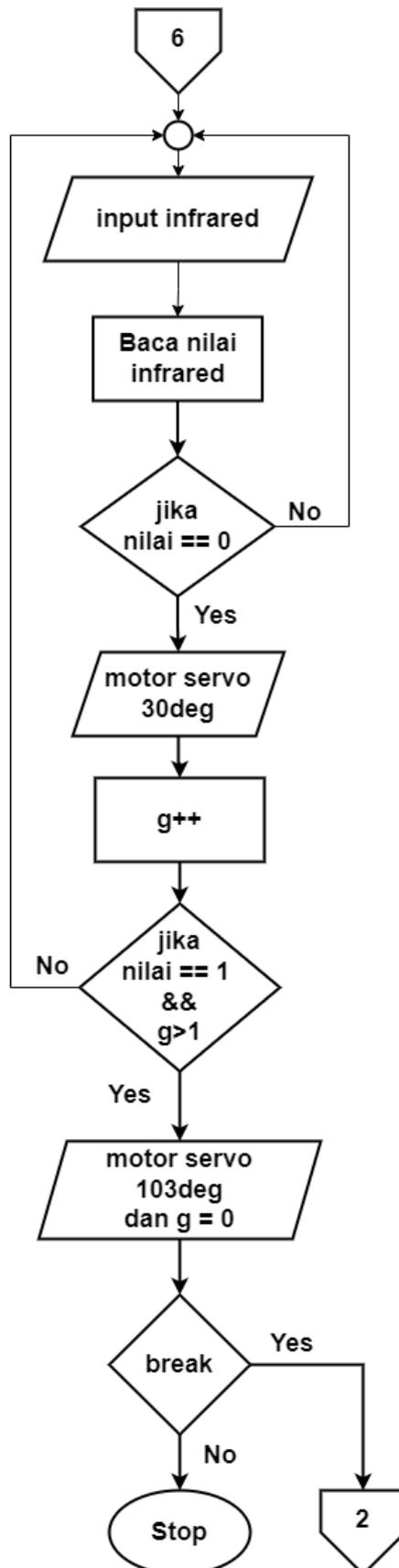
Modul program dirancang menggunakan struktur dengan kualitas yang baik dan mudah dimengerti, maka sebelum pembuatan *listing program* perlu diawali dengan penentuan logika program. Logika dasar gambaran pada penulisan ini adalah dengan menggunakan *flowchart* seperti gambar 4.14 sampai 4.16.

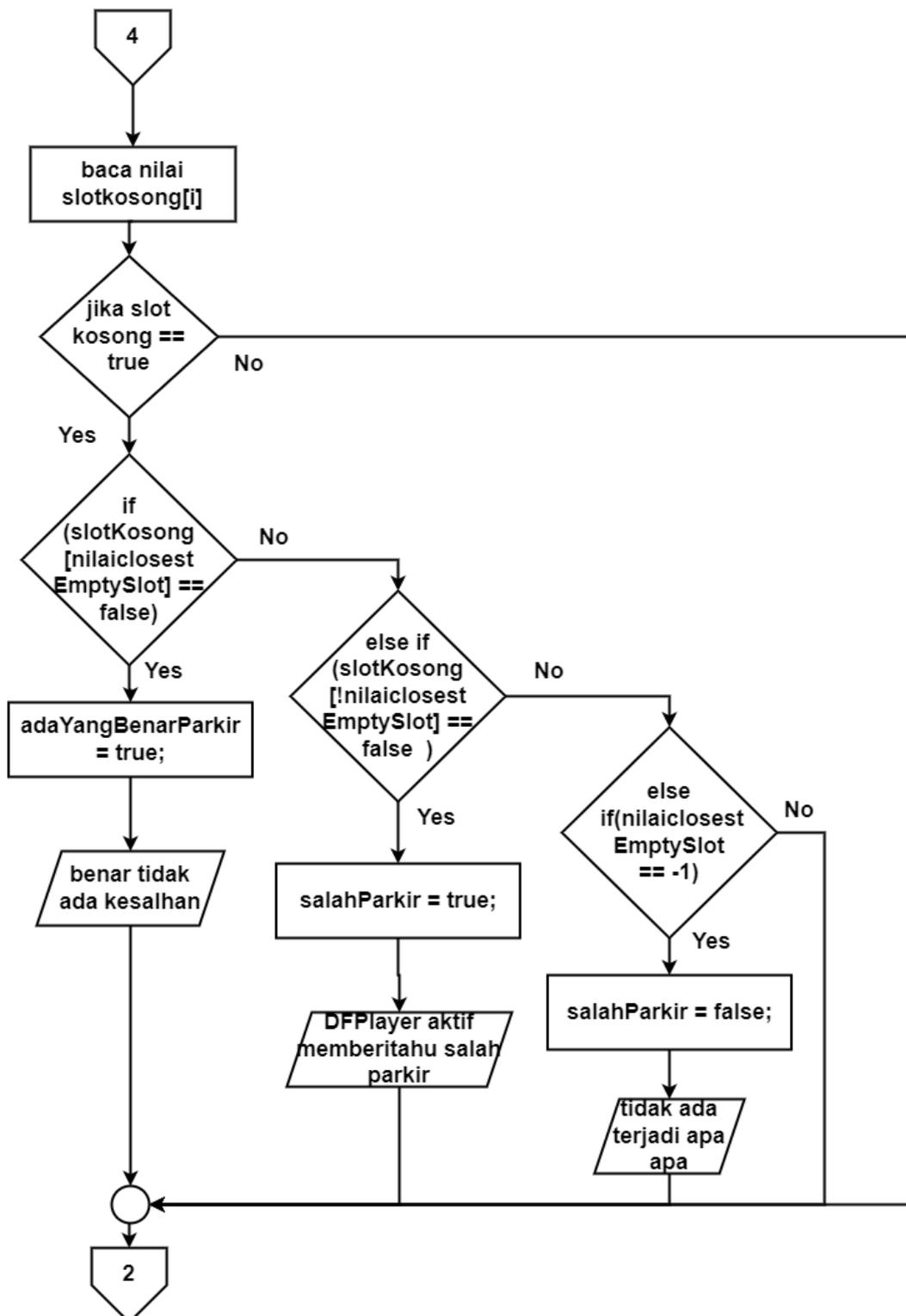




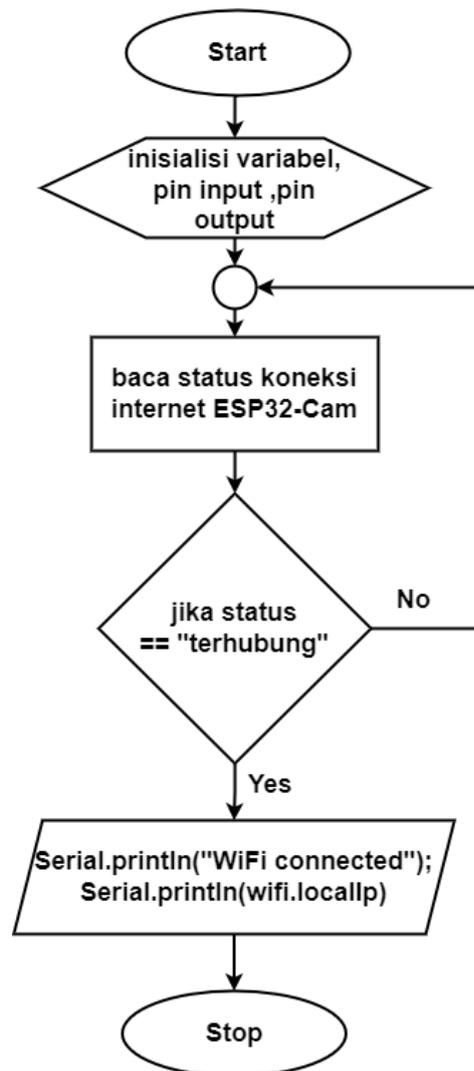




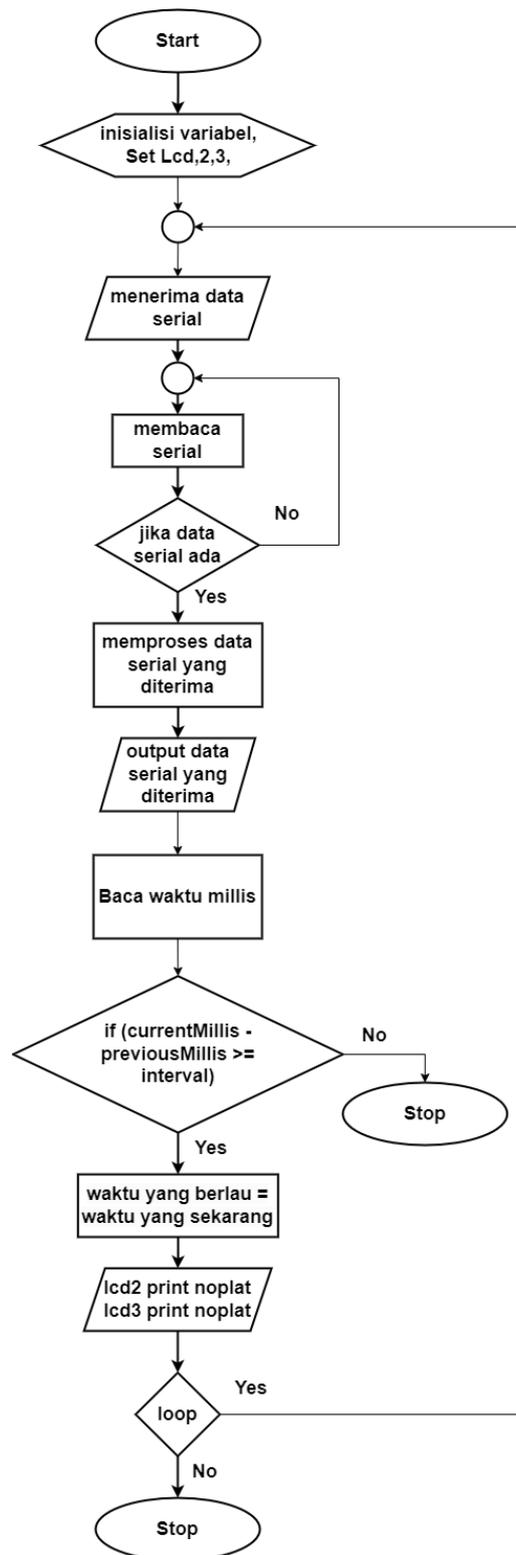




Gambar 4.14 Flowchat Arduino Mega 2560



Gambar 4.15 Flowchat ESP32-Cam



Gambar 4.16 Flowchat Arduino Uno R3

4.5.2 Modul Program

Pada sub bab ini diuraikan mengenai modul program yang digunakan untuk membuat alat sistem monitoring area parkir dan pelacakan posisi kendaraan menggunakan *software* Arduino IDE pada Arduino mega 2560 dan Arduino Uno R3 dan juga menggunakan Bahasa pemograman python. Beberapa modul yang digunakan adalah sebagai berikut:

4.5.2.1 Program Arduino Mega 2560

4.5.2.1.1 *Inisialisasi varibel, input dan Output*

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DFPlayer_Mini_Mp3.h>
#include <Servo.h>
#include <DS3231.h>

int echoPins[] = {26, 28, 30, 32, 34, 42, 44, 46, 48};
int trigPins[] = {27, 29, 31, 33, 35, 43, 45, 47, 49};
int IRSensor[] = {5, 2};
int g;
const int buttonPin[] = {23, 25};
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
Servo masuk, keluar;
```

```
DS3231 rtc(SDA, SCL);
```

```
Time t;
```

```
void setup(){
  Serial.begin(9600);
  rtc.begin();
  lcd.begin();
  mp3_set_serial (Serial);
  delay(5);
  mp3_set_volume (30);
  lcd.backlight();
```

```

lcd.clear();
for (int i = 0; i < 9; i++) {
  pinMode(trigPins[i], OUTPUT);
  pinMode(echoPins[i], INPUT);
}
for (int i = 0; i < 2; i++) {
  pinMode(buttonPin[i], INPUT_PULLUP);
  pinMode(IRSensor[i], INPUT);
}
masuk.attach(4);
keluar.attach(3);
masuk.write(10);
keluar.write(103);

//setting pertama download program
// rtc.setTime(11, 40, 0);
// rtc.setDate(12, 2, 2024);
}

```

4.5.2.1.2 *Looping Program*

```

void loop(){
  bacaslot();
  portalkeluar();
}

```

4.5.2.1.3 *Pembacaan Slot parkir Ultrasonik*

```

void bacaslot(){
  String distances = "";
  int ketersedian = 0;
  int closestEmptySlot = -1;

  for (int i = 0; i < 9; i++) {
    digitalWrite(trigPins[i], LOW);
    delayMicroseconds(2);
    digitalWrite(trigPins[i], HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPins[i], LOW);

    long duration = pulseIn(echoPins[i], HIGH);

```

```

int distance = duration * 0.034 / 2;

distances += "Sensor " + String(i + 1) + ": " + String(distance) + " cm ";

if(distance > 10){
  ketersedian++;
  if (distance > 10 && (closestEmptySlot == -1 || distance <
pulseIn(echoPins[closestEmptySlot], HIGH) * 0.034 / 2)) {
    closestEmptySlot = i;
  }
}
delay(100);
}

```

4.5.2.1.4 Program Menampilkan keLCD 1 Informasi Slot

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Available Slots:");
lcd.setCursor(0, 1);
lcd.print(ketersedian);

```

```
// Serial.println(distances);
```

4.5.2.1.5 Program Push Button Masuk

```

if (digitalRead(buttonPin[0]) == LOW) {
  Serial.write('\n');
  Serial.println("Masuk");
  jamkalender();
  Serial.println(String(closestEmptySlot + 1));
  mp3_play (closestEmptySlot + 1);
  delay (6000);
  bool buttonPressed = true;
  while(buttonPressed){
    int a = 0;
    if(a ==0){
      masuk.write(80);
      int sensorStatus1 = digitalRead(IRSensor[0]);
      if(sensorStatus1==0){
        masuk.write(80);

```

```

    g++;
  }
  if(g > 1 && sensorStatus1==1){
    g=0;
    masuk.write(10);
    break;
  }
}
}
}
}
}
}

```

4.5.2.1.6 Program Push Button Keluar

```

void portalkeluar() {
  int buttonState2 = digitalRead(buttonPin[1]);
  // Serial.println("KeluarButton = " + String(buttonState2) + ", KeluarIR = " +
  String(sensorStatus2));

```

```

  if (buttonState2 == 1) {
    keluar.write(103);
  }
  if (digitalRead(buttonPin[1]) == LOW) {
    Serial.write('\n');
    Serial.println("Keluar");
    jamkalender();
    mp3_play (10);
    delay (3500);
    bool buttonPressed = true;
    while(buttonPressed){
      int a = 0;
      if(a ==0){
        keluar.write(30);
        int sensorStatus2 = digitalRead(IRSensor[1]);
        if(sensorStatus2==0){
          keluar.write(30);
          g++;
        }
      }
      if(g > 1 && sensorStatus2==1){
        g=0;
        keluar.write(103);

```

```

        break;
    }
}
}
}
}
}

```

4.5.2.1.7 Program Jam dan Tanggal

```

void jamkalender() {
    t = rtc.getTime();

    // Send date over serial connection
    Serial.print(t.year, DEC);
    Serial.print("-");
    Serial.print(t.mon);
    Serial.print("-");
    Serial.print(t.date, DEC);
    Serial.print(" ");

    // Send Day-of-Week and time
    Serial.print(t.hour, DEC);
    Serial.print(":");
    Serial.print(t.min, DEC);
    Serial.print(":");
    Serial.println(t.sec, DEC);
    // Wait one second before repeating :)
    delay (1000);
}

```

4.5.2.2 Program Ultrasonic 9 Buah

```

int echoPins[] = {26, 28, 30, 32, 34, 42, 44, 46, 48};
int trigPins[] = {27, 29, 31, 33, 35, 43, 45, 47, 49};

void setup() {
    Serial.begin(9600);
    for (int i = 0; i < 9; i++) {
        pinMode(trigPins[i], OUTPUT);
        pinMode(echoPins[i], INPUT);
    }
}

```

```

void loop(){
  bacaslot();
}

void bacaslot(){
  String distances = "";

  for (int i = 0; i < 9; i++) {
    digitalWrite(trigPins[i], LOW);
    delayMicroseconds(2);
    digitalWrite(trigPins[i], HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPins[i], LOW);

    long duration = pulseIn(echoPins[i], HIGH);
    int distance = duration * 0.034 / 2;

    distances += "Sensor " + String(i + 1) + ": " + String(distance) + " cm ";
    delay(100);
  }

  Serial.println(distances);
  delay(1000);
}

```

4.5.2.3 Program RTC DS3231

```

#include <DS3231.h>
DS3231 rtc(SDA, SCL);
Time t;

void setup(){
  Serial.begin(115200);
  rtc.begin();

  // rtc.setDOW(2); // Set Day-of-Week to SUNDAY
  // rtc.setTime(14, 05, 0); // Set the time to 12:00:00 (24hr format)
  // rtc.setDate(6, 2, 2024); // Set the date to January 1st, 2014
}

```

```

void loop()
{
  t = rtc.getTime();

  // Send date over serial connection
  Serial.print(t.year, DEC);
  Serial.print("-");
  Serial.print(t.mon);
  Serial.print("-");
  Serial.print(t.date, DEC);
  Serial.print(" ");

  // Send Day-of-Week and time
  Serial.print(t.hour, DEC);
  Serial.print(":");
  Serial.print(t.min, DEC);
  Serial.print(":");
  Serial.println(t.sec, DEC);
  delay (1000);
}

```

4.5.2.4 Program Push Button

```

const int buttonPin[] = {23, 25}; // the number of the pushbutton pin
int buttonState = 0; // variable for reading the pushbutton status

```

```

void setup() {
  Serial.begin(9600);
  for (int i = 0; i < 2; i++) {
    pinMode(buttonPin[i], INPUT_PULLUP);
  }
}

void loop() {
  int buttonState1 = digitalRead(buttonPin[0]);
  int buttonState2 = digitalRead(buttonPin[1]);
  Serial.println("masuk = " + String(buttonState1) + ", Keluar = " +
String(buttonState2));
}

```

4.5.2.5 Program Servo SG90

```
#include <Servo.h>
Servo masuk, keluar;

void setup(){
  masuk.attach(4);
  keluar.attach(3);
  masuk.write(90);
  keluar.write(30);
}
```

```
void loop(){
  myservo.write(10);
  delay(3000);
  masuk.write(80);
  delay(3000);
  keluar.write(103);
  delay(3000);
  keluar.write(30);
  delay(3000);
}
```

4.5.2.6 Program LCD 16x2 i2c

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd1(0x27, 16, 2);
LiquidCrystal_I2C lcd2(0x26, 16, 2);
LiquidCrystal_I2C lcd3(0x25, 16, 2);

void setup(){
  lcd1.begin();
  lcd2.begin();
  lcd3.begin();
  lcd1.clear();
  lcd2.clear();
  lcd3.clear();

}
```

```

void loop(){
  lcd1.setCursor(0, 0);
  lcd1.print("Saya");
  lcd2.setCursor(0, 0);
  lcd2.print("Qadri");
  lcd3.setCursor(0, 0);
  lcd3.print("Alfisyahri");
}

```

4.5.2.7 Program Infrared Obstacle

```
int IRSensor[] = {5, 2};
```

```

void setup() {
  Serial.begin(9600);
  for (int i = 0; i < 2; i++) {
    pinMode(IRSensor[i], INPUT);
  }
}

```

```

void loop() {
  int sensorStatus1 = digitalRead(IRSensor[0]);
  int sensorStatus2 = digitalRead(IRSensor[1]);

  Serial.println("masuk = " + String(sensorStatus1) + ", Keluar = " +
String(sensorStatus2));
}

```

4.5.2.8 Program ESP32-Cam

```
#include "esp_camera.h"
```

```
#include <WiFi.h>
```

```

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG
quality
// Ensure ESP32 Wrover Module or other board with PSRAM is selected
// Partial images will be transmitted if image exceeds buffer size
//
// You must select partition scheme from the board menu that has at least
3MB APP space.

```

```

//      Face Recognition is DISABLED for ESP32 and ESP32-S2, because it
//      takes up from 15
//      seconds to process single frame. Face Detection is ENABLED if
//      PSRAM is enabled as well

// =====
// Select camera model
// =====
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B
// Has PSRAM
#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
// ** Espressif Internal Boards **
#define CAMERA_MODEL_ESP32_CAM_BOARD
#define CAMERA_MODEL_ESP32S2_CAM_BOARD
#define CAMERA_MODEL_ESP32S3_CAM_LCD
#define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
#define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
#include "camera_pins.h"

// =====
// Enter your WiFi credentials
// =====
const char* ssid = "*****"; // masukan ssid wifi yang digunakan
const char* password = "*****"; // masukan password wifi yang digunakan

void startCameraServer();
void setupLedFlash(int pin);

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);

```

```
Serial.println();
```

```
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.frame_size = FRAMESIZE_UXGA;
config.pixel_format = PIXFORMAT_JPEG; // for streaming
//config.pixel_format = PIXFORMAT_RGB565; // for face
detection/recognition
config.grab_mode = CAMERA_GRAB_WHEN_EMPTY;
config.fb_location = CAMERA_FB_IN_PSRAM;
config.jpeg_quality = 12;
config.fb_count = 1;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
//           for larger pre-allocated frame buffer.
if(config.pixel_format == PIXFORMAT_JPEG){
  if(psramFound()){
    config.jpeg_quality = 10;
    config.fb_count = 2;
    config.grab_mode = CAMERA_GRAB_LATEST;
  } else {
    // Limit the frame size when PSRAM is not available
```

```

    config.frame_size = FRAMESIZE_SVGA;
    config.fb_location = CAMERA_FB_IN_DRAM;
}
} else {
    // Best option for face detection/recognition
    config.frame_size = FRAMESIZE_240X240;
#ifdef CONFIG_IDF_TARGET_ESP32S3
    config.fb_count = 2;
#endif
}

#ifdef CAMERA_MODEL_ESP_EYE
    pinMode(13, INPUT_PULLUP);
    pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
if(config.pixel_format == PIXFORMAT_JPEG){
    s->set_framesize(s, FRAMESIZE_QVGA);
}

#ifdef CAMERA_MODEL_M5STACK_WIDE ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
#endif

```

```

#if defined(CAMERA_MODEL_ESP32S3_EYE)
  s->set_vflip(s, 1);
#endif

// Setup LED FLash if LED pin is defined in camera_pins.h
#if defined(LED_GPIO_NUM)
  setupLedFlash(LED_GPIO_NUM);
#endif

WiFi.begin(ssid, password);
WiFi.setSleep(false);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

void loop() {
  // Do nothing. Everything is done in another task by the web server
  delay(10000);
}

```

4.5.2.9 Program DFPlayer Mini Mp3

```
#include <DFPlayer_Mini_Mp3.h>
```

```

void setup () {
  Serial.begin (9600);
  mp3_set_serial (Serial);
  delay(5);
  mp3_set_volume (30);
}

```

```

}

void loop () {
  mp3_play (1);
  delay (10000);
}

```

4.5.2.10 Program LCD 2 dan 3

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int num_data = 10;
String data_plat[num_data];
LiquidCrystal_I2C lcd3(0x25, 16, 2);
LiquidCrystal_I2C lcd2(0x26, 16, 2);
unsigned long previousMillis = 0;
const long interval = 1000; // Interval waktu 1 detik

void setup() {
  Serial.begin(115200);
  lcd2.begin();
  lcd2.clear();
  lcd3.begin();
  lcd3.clear();
  // Menunggu data dari Python
  while (!Serial);
}

void loop() {
  unsigned long currentMillis = millis(); // Waktu saat ini

  // Menerima data dari Python
  for (int i = 0; i < num_data; i++) {
    while (!Serial.available()); // Menunggu hingga data tersedia
    data_plat[i] = Serial.readStringUntil('\n'); // Membaca data hingga newline
  }

  // Menampilkan data yang diterima
  if (currentMillis - previousMillis >= interval) {

```

```
previousMillis = currentMillis; // Menyimpan waktu terakhir
static int index = 0; // Indeks data plat yang ditampilkan
lcd2.clear();
lcd2.setCursor(0, 0);
lcd2.print("Plat Nomor A" + String(index + 1));
lcd2.setCursor(0, 1);
lcd2.print(data_plat[index]);
lcd3.clear();
lcd3.setCursor(0, 0);
lcd3.print("Plat Nomor B" + String(index + 1));
lcd3.setCursor(0, 1);
lcd3.print(data_plat[index + 5]); // Menampilkan data dengan indeks 5, 6, 7,
dan 8
    index = (index + 1) % 5; // Menggunakan modulus untuk menggulir melalui
data pertama
}
}
```

BAB V

PENGUJIAN SISTEM

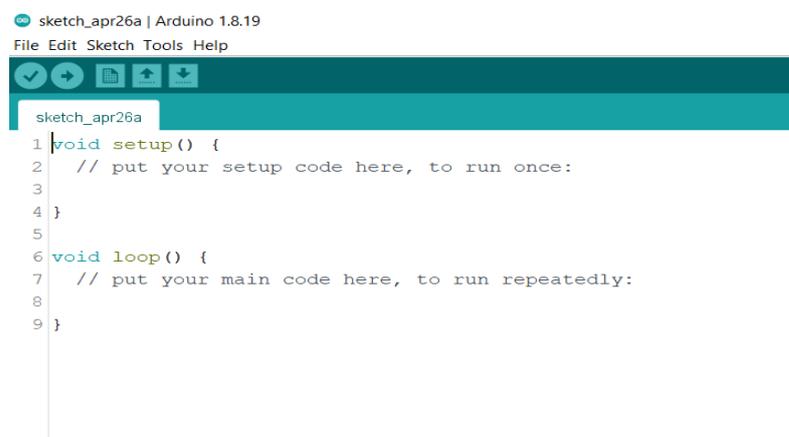
5.1 Pengujian Sistem Permodul

Pengujian berfungsi untuk melihat hasil sistem yang dirancang sudah sesuai dengan yang diinginkan atau belum. Pengujian sistem dilakukan mulai dari pengujian sistem setiap modulnya hingga pengujian sistem secara keseluruhan.

5.1.1 Pengujian Mikrokontroler Dan Software Arduino IDE

Arduino Mega 2560 menggunakan software Arduino IDE untuk pembuatan program dan download program ke Arduino Board. Adapun beberapa langkah yang harus dilakukan dalam pemrograman C pada Arduino Mega 2560 adalah sebagai berikut:

1. Jalankan aplikasi Arduino IDE, sehingga akan muncul tampilan seperti gambar berikut.

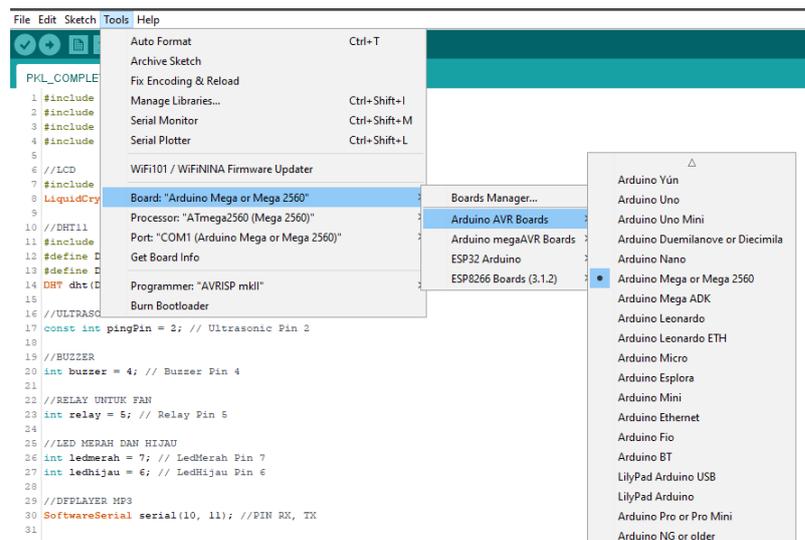


```
sketch_apr26a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_apr26a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Gambar 5.1 Tampilan Awal Arduino IDE

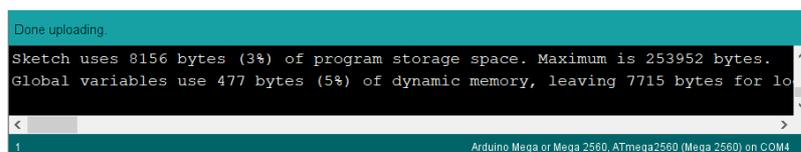
2. Masukkan program meja belajar pada jendela *sketch* yang terbuka.

3. Simpan program dengan mengklik tombol “file”, lalu pilih “save”.
4. Pasangkan kabel penghubung antara Arduino dan PC untuk mengupload program yang akan digunakan.
5. Cek dan pilih tipe *board* dan port yang sesuai dengan Arduino yang digunakan seperti gambar berikut.



Gambar 5.2 Pemilihan Tipe Board Arduino

6. *Upload* program hingga terdapat tampilan jendela seperti berikut.



Gambar 5.3 Tampilan program berhasil *upload*

5.1.2 Pengujian Sensor Ultrasonik

Pengujian sensor ultrasonik dilakukan untuk menentukan jarak antara sensor dengan slot parkir ketika ada kendaraan dan tidak ada kendaraan. Hasil pengujian sensor ultrasonik dapat dilihat pada Tabel 5.1.

Tabel 5.1 Pengujian Jarak Ultrasonik

No	Input	Jenis kendaraan	Ada Kendaraan (cm)	Ket	Tidak Ada Kendaraan (cm)	Ket
1	Sensor ultrasonic A1	Mobil bak	11	Slot Berisi	13	Slot Kosong
2	Sensor ultrasonic A2	Mobil bak	11	Slot Berisi	13	Slot Kosong
3	Sensor ultrasonic A3	Mobil cargo	7	Slot Berisi	13	Slot Kosong
4	Sensor ultrasonic A4	Mobil cargo	7	Slot Berisi	12	Slot Kosong
5	Sensor ultrasonic A5	Mobil avanza	9	Slot Berisi	13	Slot Kosong
6	Sensor ultrasonic B1	Mobil avanza	9	Slot Berisi	13	Slot Kosong
7	Sensor ultrasonic B2	Mobil avanza	9	Slot Berisi	13	Slot Kosong
8	Sensor ultrasonic B3	Mobil avanza	8	Slot Berisi	12	Slot Kosong
9	Sensor ultrasonic B4	Mobil avanza	9	Slot Berisi	13	Slot Kosong

Pengujian sensor ultrasonik untuk mendeteksi apakah pengendara parkir sesuai dengan slot parkir yang di instruksikan. Hasil pengujian sensor ultrasonik dapat dilihat pada Tabel 5.2.

Tabel 5.2 Pengujian Kesesuaian Parkir

No	Kendaraan	Rencana Parkir	Hasil	Keterangan
1	BA 1117 AN	Lantai 1 Slot A1	Lantai 1 Slot A1	Sesuai Dengan Instruksi
2	BA 0912 QA	Lantai 1 Slot A2	Lantai 1 Slot A2	Sesuai Dengan Instruksi
3	BA 1507 EN	Lantai 1 Slot A3	Lantai 1 Slot A3	Sesuai Dengan Instruksi
4	BA 1004 KH	Lantai 1 Slot A4	Lantai 1 Slot A4	Sesuai Dengan Instruksi
5	BA 3103 DN	Lantai 1 Slot A5	Lantai 1 Slot A5	Sesuai Dengan Instruksi
6	B 1207 AK	Lantai 2 Slot B1	Lantai 2 Slot B1	Sesuai Dengan Instruksi
7	BG 2487 AN	Lantai 2 Slot B2	Lantai 2 Slot B2	Sesuai Dengan Instruksi
8	BM 6478 AR	Lantai 2 Slot B3	Lantai 2 Slot B3	Sesuai Dengan Instruksi
9	BH 3567 HY	Lantai 2 Slot B4	Lantai 2 Slot B4	Sesuai Dengan Instruksi
10	B 1207 AK	Lantai 2 Slot B1	Slot penuh	Tidak Bisa Parkir
11	BM 6478 AR	Lantai 1 Slot A3	Lantai 2 Slot B3	Parkir salah tidak sesuai

5.1.3 Pengujian Push Button

Pengujian push button dilakukan untuk menentukan apakah output yang akan diberikan. Hasil pengujian push button dapat dilihat pada tabel 5.3.

Tabel 5.3 Pengujian Push Button

NO	Input	Nilai	Servo	DFPlayer	Python
1	Push Button (Masuk)	1	Menutup portal masuk (10deg)	-	
2	Push Button (Masuk)	0	Membuka portal masuk (80deg)	Mengeluarkan informasi slot parkir yang terdekat	Menerima data nilai push button dan mencapture, tanggaljam, dan no slot
3	Push Button (Keluar)	1	Menutup Portal keluar (103deg)	-	
4	Push Button (Keluar)	0	Membuka Portal Keluar (30deg)	Mengeluarkan ucapan terima kasih.	Menerima data nilai push button, tanggaljam dan mencapture, tanggaljam

5.1.4 Pengujian Infrared Obstacle

Pengujian infrared obstacle dilakukan untuk menentukan apakah output yang akan diberikan ke motor servo . Hasil pengujian IR Obstacle dapat dilihat pada Tabel 5.4.

Tabel 5.4 Pengujian IR Obstacle

NO	Input	Nilai	Servo
1	Infrared obstacle (Masuk)	1	Menutup portal masuk (10deg)
2	Infrared obstacle (Masuk)	0	Membuka portal masuk (80deg)

NO	Input	Nilai	Servo
3	Infrared obstacle (Keluar)	1	Menutup Portal keluar (103deg)
4	Infrared obstacle (Keluar)	0	Membuka Portal Keluar (30deg)

Pengujian infrared obstacle untuk menentukan jarak terdeteksinya halangan. Hasil pengujian IR Obstacle dapat dilihat pada Tabel 5.5.

Tabel 5.5 Pengujian IR Obstacle Jarak Terdeteksi

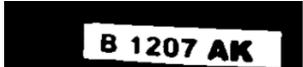
NO	Input	Jarak (cm)		
		3	5	7
1	Infrared obstacle (Masuk)	Bisa	Bisa	Tidak Bisa
2	Infrared obstacle (Keluar)	Bisa	Bisa	Bisa

5.1.5 Pengujian ESP32-Cam

Pengujian ESP32-Cam dilakukan untuk menentukan apakah gambar yang ditangkap berhasil di ubah menjadi biner untuk pembacaan OCR. Hasil pengujian ESP32-Cam dapat dilihat pada Tabel 5.6.

Tabel 5.6 Pengujian ESP32-Cam

NO	Input	Nilai Thresh	Output
1		<pre>Teks dari gambar: Nilai threshold: [[0 0 0 ... 0 0 0]] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]] BA 1004 KH</pre>	
2		<pre>Teks dari gambar: Nilai threshold: [[0 0 0 ... 0 0 0]] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]] BA 1117 AN</pre>	
3		<pre>Teks dari gambar: Nilai threshold: [[0 0 0 ... 0 0 0]] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]] BA 0912 QA</pre>	

NO	Input	Nilai Thresh	Output
4		<pre>Teks dari gambar: Nilai threshold: [[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]] BA 1507 EN</pre>	
5		<pre>Teks dari gambar: Nilai threshold: [[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]] BA 3103 DN</pre>	
6		<pre>Teks dari gambar: Nilai threshold: [[0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] ... [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0] [0 0 0 ... 0 0 0]] B 1207 AK</pre>	

5.2 Pengujian Sistem Keseluruhan

Pengujian secara keseluruhan merupakan pengujian yang dilakukan menggunakan seluruh entity yang ada berdasarkan langkah – langkah yang telah ditentukan . Adapun langkah pengujian sistem keseluruhan dapat dilihat sebagai berikut :

1. Hidupkan alat dengan memasang colokan ke arus AC, dan menyambungkan koneksi arduino mega 2560 dan uno R3 menggunakan kabel serial ke laptop.
2. Ultrasonik akan langsung menghitung jarak dari sensor ke lantai slot parkir, apakah ada kendaraan atau tidaknya, jika jarak yang dideteksi besar dari 10cm maka tidak ada kendaraan, jika kecil dari sama dari 10cm maka ada kendaraan berada pada slot parkir.
3. Ketika jarak yang dideteksi besar dari 10 cm maka LCD akan menampilkan jumlah slot yang tersedia, seperti Gambar 5.4.



Gambar 5.4 Tampilan LCD 1 informasi ketersediaan slot

Gambar diatas merupakan tampilan LCD 1 yang berada di pintu masuk gedung parkir yang menunjukkan informasi jumlah slot parkir yang tersedia pada gedung parkir tersebut. Jumlah ketersediaan parkir yang ditampilkan di LCD 1 ini akan berkurang Ketika ada slot yang terisi.

4. Push button masuk akan mengirimkan perintah ke arduino untuk memutar DFPlayer memberikan informasi mengenai slot parkir yang kosong terdekat dan menunjukkan arah slot tersebut dimana berada. Setelah informasi disampaikan portal akan terbuka. seperti Gambar 5.5.

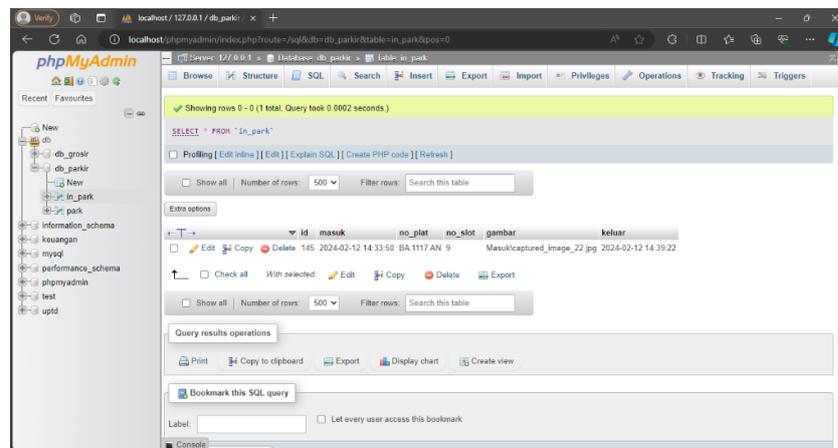


Gambar 5.5 Tampilan portal terbuka masuk

5. Push button masuk ketika ditekan dia akan mengirimkan data ke python melalui sebuah serial yang dikirim berupa kata masuk, data jam dan tanggal, no slot parkir yang tertuju.
6. Ketika data yang diterima tadi ada kata masuk, python akan mengolah data

tersebut untuk melakukan OCR pada gambar yang ditangkap melalui kamera ESP32_Cam.

7. Ketika hasil OCR selesai data hasil OCR tadi akan dikirim ke dalam database bersamaan dengan data yang dikirimkan arduino ke python seperti data tanggaljam, dan no_slot parkir yang dituju. seperti Gambar 5.6.



Gambar 5.6 Tampilan database

8. Ketika data hasil pembacaan OCR tadi sudah masuk kedalam database ,nanti dia akan mengirimkan data no plat yang parkir pada slot ke LCD 2 atau 3 sesuai dimana saja mobil terparkir. seperti Gambar 5.7.

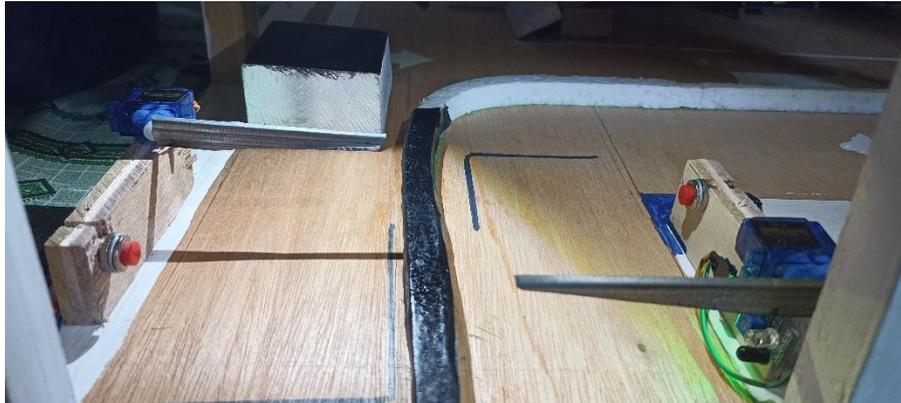


(a) LCD 2

(b) LCD 3

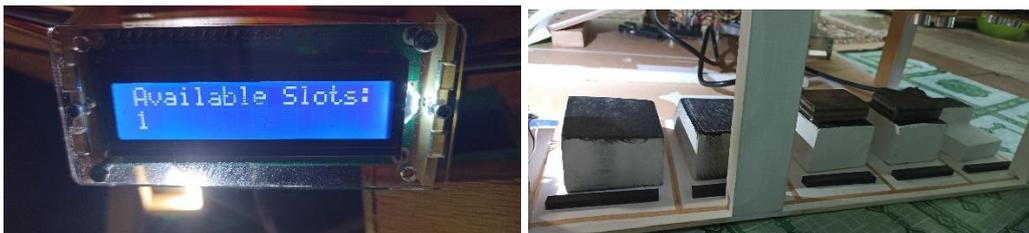
Gambar 5.7 Tampilan Data Plat

9. Ketika kendaraan melewati sensor IR masuk itu akan membuat portal yang terbuka tadi tertutup secara otomatis



Gambar 5.8 Tampilan portal tertutup masuk

10. Kendaraan parkir pada slot yang ditentukan diawal masuk dan layar LCD 1 akan mengurangi jumlah ketersediaan slot, seperti Gambar 5.9.



(a) LCD 1

(b) Kendaran Parkir

Gambar 5.10 Tampilan Slot Diisi

11. Push button keluar akan mengirimkan perintah ke arduino untuk memutar DFPlayer memberikan kata ucapan “ terima kasih sudah menggunakan fasilitas parkir kami” Setelah informasi disampaikan portal akan terbuka. seperti Gambar 5.10.



Gambar 5.10 Tampilan portal terbuka keluar

12. Push button keluar ketika ditekan dia akan mengirimkan data ke python melalui sebuah serial yang dikirim berupa kata keluar, data jam dan tanggal,.
13. Ketika data yang diterima tadi ada kata keluar, python akan mengolah data tersebut untuk melakukan OCR pada gambar yang ditangkap melalui kamera ESP32_Cam.
14. Ketika hasil OCR selesai data hasil OCR tadi akan dikirim ke dalam database dan mengupdate data didalamnya bersamaan dengan data yang dikirimkan arduino ke python seperti data tanggaljam.
15. Ketika data hasil pembacaan OCR tadi sudah masuk kedalam database ,nanti dia akan mengirimkan data no plat yang terbaru yang sudah di update ketika kendaraan telah keluar dari gedung parkir dan tampilan LCD 2 atau 3 akan berubah menjadi kosong pada slot yang tidak terisi lagi.
16. Ketika kendaraan melewati sensor IR keluar itu akan membuat portal yang terbuka tadi tertutup secara otomatis, seperti Gambar 5.11.



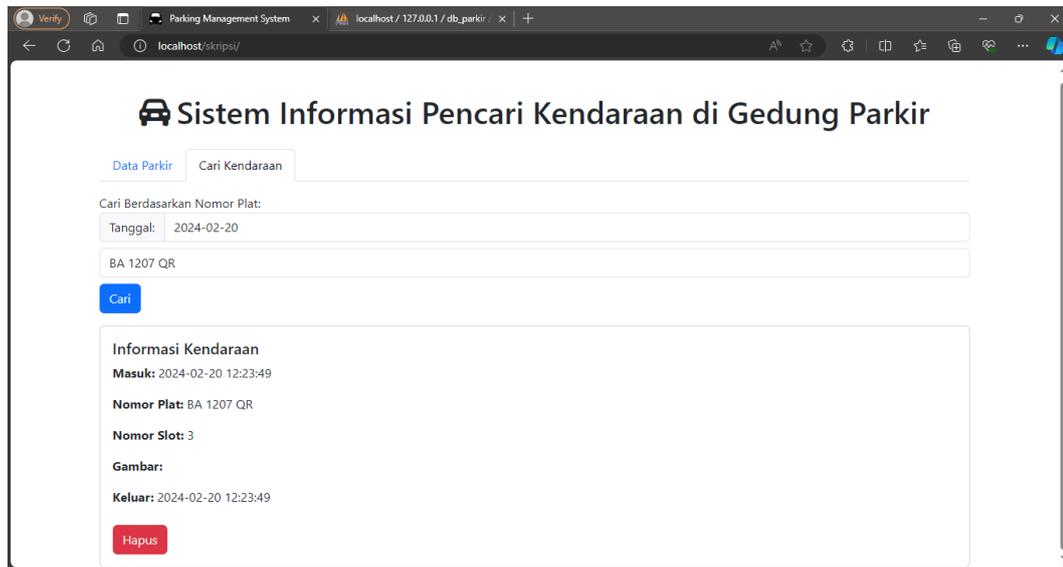
Gambar 5.11 Tampilan Portal Tertutup

17. Data parkir dari kendaraan yang parkir di gedung parkir bisa di cek di halaman websitenya. seperti Gambar 5.12.

Tanggal Masuk	Nomor Plat	Nomor Slot	Gambar Plat	Tanggal Keluar
2024-02-12 14:33:50	BA 1117 AN	9		2024-02-12 14:39:22
2024-02-14 23:57:50	BA 0912 QA	5		2024-02-14 23:57:50
2024-02-15 14:15:06	BA 1117 AN	1		2024-02-16 11:59:19

Gambar 5.12 Website Menampilkan Data kendaraan

18. Data kendaraan bisa juga dicari di halaman website tersebut dengan memasukan no plat kendaraan. seperti Gambar 5.13.



Gambar 5.13 Website Pencarian Data Kendaraan di Gedung Parkir

BAB VI

PENUTUP

Berdasarkan hasil perancangan sistem dan pembuatan alat ini, maka dapat diambil kesimpulan dan saran-saran yang nantinya bisa bermanfaat untuk pengembangan sistem berikutnya:

6.1 Kesimpulan

Berdasarkan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Arduino Mega 2560 mampu mengontrol sistem monitoring sistem parkir berbasis database dan computer vision. Penggunaan ultrasonik dapat mengatasi permasalahan sistem monitoring area parkir yaitu untuk mengetahui ketersediaan ruang parkir pada gedung bertingkat.
2. Dengan penggunaan dfplayer dan speaker, pengguna kendaraan dapat mengetahui sistem navigasi area parkir dengan mudah untuk menemukan slot parkir yang kosong dan dapat meningkatkan efisiensi pencarian parkir secara realtime.
3. Implementasi sistem database dan computer vision dapat mengimplementasikan sistem pelacakan posisi parkir kendaraan yang terintegrasi dengan sebuah website dan pengguna dapat dengan mudah menemukan kembali posisi parkir mereka, termasuk informasi tentang lantai dan slot parkir yang digunakan. Tetapi implementasi sistem database dan computer vision bergantung dengan jaringan, jika jaringan terlalu

lambat atau lemah pendeteksian nomor plat kendaraan terjadi error yang tidak dapat membaca nomor plat kendaraan untuk disimpan ke database.

6.2 Saran

Berdasarkan pengalaman yang didapatkan dari perancangan dan pembuatan alat, ada beberapa saran yang ingin disampaikan agar berguna untuk pengembangan alat selanjutnya adalah sebagai berikut:

1. Pada rancangan sistem ini tidak memiliki sebuah fitur yang dapat memesan atau membooking slot parkir. Pengembangan selanjutnya disarankan ditambahkan sistem aplikasi booking tempat parkir, dimana di sistem ini bisa memesan slot parkir sebelum datang ke gedung parkirnya.
2. Pada rancangan sistem ini menggunakan kamera ESP32-Cam yang mana masih keterbasan dengan kejernihan dan pencahayaan dalam mencapture gambar dan itu mempengaruhi dalam OCR. Pengembangan berikutnya, disarankan bisa menggunakan kamera yang lebih bagus kualitasnya dan pencahayaan yang lebih bagus juga supaya pembacaan OCR nya bersih dan bagus.
3. Pada rancangan sistem ini masih menggunakan sensor sensor yang terbilang masih biasa belum cukup bagus untuk pendeteksian dan pembacaan. Pengembangan berikutnya, disarankan bisa menggunakan komponen-komponen elektronika yang lebih berkualitas agar dapat membuat alat ini bekerja lebih sempurna dari sebelumnya.

Berdasarkan kesimpulan dan saran diatas, diharapkan bisa dipahami dengan sebaik mungkin supaya alat ini dapat dikembangkan menjadi lebih baik lagi dan bisa menjadi sumber referensi untuk penelitian berikutnya.

DAFTAR PUSTAKA

- Amalia Yunia Rahmawati. (2020). *Rancang bangun Portal Parkir*. etheses.iainponorogo.ac.id. *July*, 1–23.
- BETA, S., & Astuti, S. (2019). MODUL TIMBANGAN BENDA DIGITAL DILENGKAPI LED RGB DAN DFPLAYER MINI. *Orbit*, 15(1), 10–15.
- Budiharto, W. (2020). *Menguasai Pemrograman Arduino dan Robot*. 92.
- Budiman, I., Saori, S., Anwar, R. N., Pangestu, M. Y., & Fitriani. (2021). ANALISIS PENGENDALIAN MUTU DI BIDANG INDUSTRI MAKANAN (Studi Kasus: UMKM Mochi Kaswari Lampion Kota Sukabumi). *Jurnal Inovasi Penelitian*, 1(0.1101/2021.02.25.432866), 1–15.
- Dwi Cahyono, B., Iqbal Nugraha, M., Sultan Ageng Tirtayasa, U., Raya Palka NoKm, J., Cipocok Jaya, K., & Serang, K. (2023). Pengembangan Media Pembelajaran Elektronika Dasar Untuk Memahami Nilai Resistor Berdasarkan Kode Warna 3 Gelang Dan 4 Gelang Bagi Siswa SMK Kelas X Jurusan Teknik Otomasi Industri. *Journal on Education*, 05(04), 11547–11557.
- Friadi, R., & Junadhi. (2019). Sistem Kontrol Intensitas Cahaya, Suhu dan Kelembaban Udara Pada Greenhouse Berbasis Raspberry PI. *Journal of Technopreneurship and Information System (JTIS)*, 2(1), 30–37. <https://doi.org/10.36085/jtis.v2i1.217>
- Harani, N. H., & Hasanah, M. (2020). *DETEKSI OBEK DAN PENGENALAN*

KARAKTER PLAT NOMOR KENDARAAN INDONESIA BERBASIS PYTHON (Y. H. Setyawan (ed.)). kreatif industri nusantara.
[https://books.google.co.id/books?hl=en&lr=&id=saD6DwAAQBAJ&oi=fn&d&pg=PR6&dq=deteksi+plat+kendaraan&ots=GI394-OQtK&sig=eb9BaZINRr2BfY7K5UXa1020Pnk&redir_esc=y#v=onepage&q=deteksi plat kendaraan&f=false](https://books.google.co.id/books?hl=en&lr=&id=saD6DwAAQBAJ&oi=fn&d&pg=PR6&dq=deteksi+plat+kendaraan&ots=GI394-OQtK&sig=eb9BaZINRr2BfY7K5UXa1020Pnk&redir_esc=y#v=onepage&q=deteksi+plat+kendaraan&f=false)

Haviluddin. (2021). Memahami Penggunaan Diagram Arus Data. *Jurnal Informatika Mulawarman*, 4(3).

Hidayat, L., Kurniawan, E., & Ramdhani, M. (2022). Perancangan Sistem Palang Parkir Otomatis Dan Pendeteksi Slot Parkir Berbasis Iot. *E-Proceeding of Engineering*, 9(2), 174–180.

Irsyam, M., & Wiranata, A. (2020). Perancangan Sistem Parkir Mobil Otomatis Menggunakan Nfc Reader Pn532 Berbasis Arduino. *Sigma Teknika*, 3(1), 22–32. <https://doi.org/10.33373/sigma.v3i1.2450>

Kustina, K. T., Nurhayati, Heratati, L., Qodari, A., Jaya, A., & Marthalia, D. (2022). Sistem Informasi Manajemen. In *CV. Pena Persada* (Issue April). <http://max21487.blogspot.com/2012/04/tujuan-sistem-informasi-manajemen.html>

Mallisza, D., Hadi, H. S., & Aulia, A. T. (2022). Implementasi Model Waterfall Dalam Perancangan Sistem Surat Perintah Perjalanan Dinas Berbasis Website Dengan Metode SDLC. *Jurnal Teknik, Komputer, Agroteknologi Dan Sains*, 1(1), 24–35. <https://doi.org/10.56248/marostek.v1i1.9>

- Mardiyati, S., Khoir Rahman, A., & Nugraha, Y. (2022). Perancangan Sistem Informasi Penjualan barang Berupa Alat Music Di Toko Martmusic. *Jurnal Inovasi Informatika*, 7(1), 86–95. <https://doi.org/10.51170/jii.v7i1.214>
- Mindasari, S., & Meilantika, D. (2022). Sistem Keamanan Kotak Amal di Musala Sabilul Khasanah Berbasis Arduino UNO. *Sistem Keamanan Kotak Amal Di Musala Sabilul Khasanah Berbasis Arduino UNO*, 5(2), 2–3. [file:///C:/Users/LENOVO/Downloads/239-File Utama Naskah-862-1-10-20230726.pdf](file:///C:/Users/LENOVO/Downloads/239-File%20Utama%20Naskah-862-1-10-20230726.pdf)
- Mualim, I. (2021). Sistem Komputerisasi Absen Guru dan Jadwal Mengajar Pada SMK Darul Amal Kota Metro. *Electrician*, 15(1), 12–19. <https://doi.org/10.23960/elc.v15n1.2178>
- Nafsin, M., Qashlim, A. A., & Khairat, U. (2020). Sistem Informasi Data Siswa Berbasis Optical Character Recognition (Ocr). *Jurnal Media Informatika Budidarma*, 4(1), 50–56. <http://ejournal.stmik-budidarma.ac.id/index.php/mib/article/view/2432>
- Palinggi, N., Abdul Aziz, N. M., Kambuno, D., Aminah, N., & Praminasari, R. (2021). Alat Monitoring Pengunjung Mall dengan Standar Covid-19 Berbasis Arduino. *Prosiding Seminar Nasional Teknik Elektro Dan Informatika (SNTEI), September*, 276–280.
- Pulungan, A. I., Sumarno, S., Gunawan, I., Tambunan, H. S., & Damanik, A. R. (2022). Rancang Bangun Sistem Parkir dan Ketersediaan Slot Parkir Otomatis Menggunakan Arduino. *Jurnal Ilmu Komputer Dan Informatika*,

2(2), 127–136. <https://doi.org/10.54082/jiki.33>

Rahardjo, P. (2022). Sistem Penyiraman Otomatis Menggunakan Sensor Kelembaban Tanah Berbasis Mikrokontroler Arduino Mega 2560 Pada Tanaman Mangga Harum Manis Buleleng Bali. *Majalah Ilmiah Teknologi Elektro*, 21(1), 31. <https://doi.org/10.24843/mite.2022.v21i01.p05>

Reza Eka Alfarisi. (2020). *Rancang Bangun Aplikasi Terjemahan Bahasa Jepang - Indonesia Berbasis Android Menggunakan Tesseract OCR*.

Romadhon, A. S., & Umam, F. (2021). *PROJECT SISTEM KONTROL BERBASIS ARDUINO* (Tim MNC Publishing (ed.); 1st ed.). Media Nusa Creative. https://www.google.co.id/books/edition/Project_Sistem_Kontrol_Berbasis_Arduino/ormeEAAAQBAJ?hl=id&gbpv=1&dq=aplikasi+arduino&printsec=frontcover

Santoso, S. P., & Wijayanto, F. (2022). RANCANG BANGUN AKSES PINTU DENGAN SENSOR SUHU DAN HANDSANITIZER OTOMATIS BERBASIS ARDUINO. *Jurnal Elektro*, 10(1), 1–23.

Silaban, J., Nasution, A. A., & Roza, I. (2020). Pemanfaatan Thermo Electric Generator Dari Konversi Energi Panas Menjadi Listrik Untuk Charger Ponsel. *JiTEKH*, 8(2), 71–77. <https://doi.org/10.35447/jitekh.v8i2.295>

Sunarwan. (2020). *PERANCANGAN DAN PEMBUATAN ALAT PEMUTAR NARASI AUDIO TENTANG OBJEK MUSEUM MENGGUNAKAN TEKNOLOGI RFID BERBASIS ARDUINO*. 28–29.

- Sutanti, A., MZ, M. K., Mustika, M., & Damayanti, P. (2020). Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur. *Komputa: Jurnal Ilmiah Komputer Dan Informatika*, 9(1), 1–8. <https://doi.org/10.34010/komputa.v9i1.3718>
- Suyono, H., & Hambali, H. (2020). Perancangan Alat Pengukur Kadar Gula dalam Darah Menggunakan Teknik Non-Invasive Berbasis Mikrokontroler Arduino Uno. *JTEV (Jurnal Teknik Elektro Dan Vokasional)*, 6(1), 69. <https://doi.org/10.24036/jtev.v6i1.107482>
- Tirtana, E., Gunadi, K., & Sugiarto, I. (2021). Penerapan Metode YOLO dan Tesseract-OCR untuk Pendataan Plat Nomor Kendaraan Bermotor Umum di Indonesia Menggunakan Raspberry Pi. *Jurnal Infra*, 9(2), 241–247. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/11454>
- Ummah, H. A., Sodikin, I., & Susetyo, J. (2019). Perancangab Sistem Informasi rental & Inventaris Alat Multimedia Berbasis Web Menggunakan Metode Customer Relationship Management. *Jurnal Rekavasi*, 7(1), 15–24.
- Wicaksono, M. F. (2019). *APLIKASI ARDUINO dan SENSOR* (1st ed.). INFORMATIKA.
- Widharma, G. S., & Wiranata, L. F. (2022). *MIKROKONTROLER DAN APLIKASI* (N. Wahid (ed.); 1st ed.). Wawasan Ilmu. https://books.google.co.id/books?id=AsKAEAAAQBAJ&printsec=frontcover&hl=id&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

- Widodo, B., & Almasri. (2021). Rancang Sistem Informasi Parkir Otomatis dengan Menentukan Posisi Parkir Berbasis Telegram Menggunakan Arduino Mega2560. *Jurnal Pendidikan Tambusai*, 5(3), 7055–7074.
- Wisoso, A. P., Irawan, D., & Astutik, R. P. (2022). Rancang Bangun Sistem Informasi Ketersediaan Slot Parkir Dalam Mall. *E-Link: Jurnal Teknik Elektro Dan Informatika*, 17(2), 19. <https://doi.org/10.30587/e-link.v17i2.4640>
- Yuliani, F. (2022). *Analisis Dan Implementasi Object Tracking Pada Kamera Webcam Dengan Image Processing Menggunakan Metode Mean Shift*.

LAMPIRAN

Listing program

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DFPlayer_Mini_Mp3.h>
#include <Servo.h>
#include <DS3231.h>

int echoPins[] = {26, 28, 30, 32, 34, 42, 44, 46, 48};
int trigPins[] = {27, 29, 31, 33, 35, 43, 45, 47, 49};
int IRSensor[] = {5, 2};
int g;
const int buttonPin[] = {23, 25};

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo masuk, keluar;
DS3231 rtc(SDA, SCL);
Time t;

bool slotKosong[9];
bool salahParkir = false;
int closestEmptySlot = -1;
int nilaiclosestEmptySlot = -1;

void setup(){
  Serial.begin(9600);
  rtc.begin();
  lcd.begin();
  mp3_set_serial (Serial);
  delay(5);
  mp3_set_volume (30);
  lcd.backlight();
  lcd.clear();
  for (int i = 0; i < 9; i++) {
    pinMode(trigPins[i], OUTPUT);
    pinMode(echoPins[i], INPUT);
  }
  for (int i = 0; i < 2; i++) {
    pinMode(buttonPin[i], INPUT_PULLUP);
    pinMode(IRSensor[i], INPUT);
  }
  masuk.attach(4);
  keluar.attach(3);
  masuk.write(10);
  keluar.write(103);
}
```

```

//setting pertama download program
// rtc.setTime(9, 27, 0);
// rtc.setDate(15, 2, 2024);
}

void loop(){
  bacaslot();
  portalkeluar();
  cekSalahParkir();
}

void bacaslot(){
  String distances = "";
  int ketersedian = 0;
  int closestEmptySlot = -1;

  for (int i = 0; i < 9; i++) {
    digitalWrite(trigPins[i], LOW);
    delayMicroseconds(2);
    digitalWrite(trigPins[i], HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPins[i], LOW);

    long duration = pulseIn(echoPins[i], HIGH);
    int distance = duration * 0.034 / 2;

    distances += "Sensor " + String(i + 1) + ": " + String(distance) + " cm ";

    if(distance > 10){
      ketersedian++;
      slotKosong[i] = true;
      if (distance > 10 && (closestEmptySlot == -1 || distance <
pulseIn(echoPins[closestEmptySlot], HIGH) * 0.034 / 2)) {
        closestEmptySlot = i;
      }
    }else {
      slotKosong[i] = false;
    }
    delay(100);
  }

  // Serial.println("Status Slot Kosong:");
  // Serial.println(String(nilaiclosestEmptySlot));
  // for (int i = 0; i < 9; i++) {
  //   Serial.print("Slot ");

```

```

// Serial.print(i + 1);
// Serial.print(": ");
// Serial.println(slotKosong[i]);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Available Slots:");
lcd.setCursor(0, 1);
lcd.print(ketersedian);

// Serial.println(distances);
if (digitalRead(buttonPin[0]) == LOW) {
  Serial.write("\n");
  Serial.println("Masuk");
  jamkalender();
  if (closestEmptySlot == -1){
    mp3_play(12);
    delay(2000);
  }
  nilaiclosestEmptySlot = closestEmptySlot;
  Serial.println(String(nilaiclosestEmptySlot));
  Serial.println(String(closestEmptySlot + 1));
  mp3_play (closestEmptySlot + 1);
  delay (6000);
  bool buttonPressed = true;
  while(buttonPressed){
    int a = 0;
    if(a ==0){
      masuk.write(80);
      int sensorStatus1 = digitalRead(IRSensor[0]);
      if(sensorStatus1==0){
        masuk.write(80);
        g++;
      }
    }
    if(g > 1 && sensorStatus1==1){
      g=0;
      masuk.write(10);
      break;
    }
  }
}
}

void cekSalahParkir() {
  bool adaYangBenarParkir = false;

```

```

if (nilaiclosestEmptySlot == -1) {
    salahParkir = false;
    return;
}

for (int i = 0; i < 9; i++) {
    if (i != nilaiclosestEmptySlot) { // Pengecualian slot closestEmptySlot
        if (slotKosong[i] == false) { // Jika slot tidak kosong
            salahParkir = true;
            mp3_play(11);
            delay(4500);
//      Serial.println("Ada yang Salah Parkir!");
            return; // Keluar dari loop segera setelah menemukan slot yang salah parkir
        }
    } else { // Slot closestEmptySlot
        if (slotKosong[nilaiclosestEmptySlot] == false) { // Jika slot kosong
            adaYangBenarParkir = true;
        }
    }
}

// if (!salahParkir && adaYangBenarParkir) {
// //   Serial.println("Semua Parkir Benar!");
// }
//
// if (salahParkir) {
//   mp3_play(11);
//   delay(4500);
// //   Serial.println("Ada yang Salah Parkir!");
// }

    salahParkir = false; // Reset status "salah parkir"
}

void portalkeluar() {
    int buttonState2 = digitalRead(buttonPin[1]);
//   Serial.println("KeluarButton = " + String(buttonState2) + ", KeluarIR = " +
String(sensorStatus2));

    if (buttonState2 == 1) {
        keluar.write(103);
    }
    if (digitalRead(buttonPin[1]) == LOW) {
        Serial.write('\n');
        Serial.println("Keluar");
    }
}

```

```

jamkalender();
mp3_play (10);
delay (3500);
bool buttonPressed = true;
while(buttonPressed){
  int a = 0;
  if(a ==0){
    keluar.write(30);
    int sensorStatus2 = digitalRead(IRSensor[1]);
    if(sensorStatus2==0){
      keluar.write(30);
      g++;
    }
    if(g > 1 && sensorStatus2==1){
      g=0;
      keluar.write(103);
      break;
    }
  }
}
}
}
}
}

```

```

void jamkalender(){
  t = rtc.getTime();

  // Send date over serial connection
  Serial.print(t.year, DEC);
  Serial.print("-");
  Serial.print(t.mon);
  Serial.print("-");
  Serial.print(t.date, DEC);
  Serial.print(" ");

  // Send Day-of-Week and time
  Serial.print(t.hour, DEC);
  Serial.print(":");
  Serial.print(t.min, DEC);
  Serial.print(":");
  Serial.println(t.sec, DEC);
  // Wait one second before repeating :)
  delay (1000);
}

```

Listing program kedua

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int num_data = 10;
String data_plat[num_data];
LiquidCrystal_I2C lcd3(0x25, 16, 2);
LiquidCrystal_I2C lcd2(0x26, 16, 2);
unsigned long previousMillis = 0;
const long interval = 1000; // Interval waktu 1 detik

void setup() {
  Serial.begin(115200);
  lcd2.begin();
  lcd2.clear();
  lcd3.begin();
  lcd3.clear();
  // Menunggu data dari Python
  while (!Serial);
}

void loop() {
  unsigned long currentMillis = millis(); // Waktu saat ini

  // Menerima data dari Python
  for (int i = 0; i < num_data; i++) {
    while (!Serial.available()); // Menunggu hingga data tersedia
    data_plat[i] = Serial.readStringUntil('\n'); // Membaca data hingga newline
  }

  // Menampilkan data yang diterima
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; // Menyimpan waktu terakhir
    static int index = 0; // Indeks data plat yang ditampilkan
    lcd2.clear();
    lcd2.setCursor(0, 0);
    lcd2.print("Plat Nomor A" + String(index + 1));
    lcd2.setCursor(0, 1);
    lcd2.print(data_plat[index]);
    lcd3.clear();
    lcd3.setCursor(0, 0);
    lcd3.print("Plat Nomor B" + String(index + 1));
    lcd3.setCursor(0, 1);
    lcd3.print(data_plat[index + 5]);
    index = (index + 1) % 5;
  }
}

```

LEMBAR PERNYATAAN

Saya yang bertanda tangan dibawah ini :

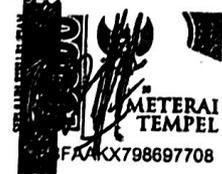
Nama : Qadri Alfisyahri
NoBP : 20101152620032
Fakultas : ILMU KOMPUTER
Jurusan : SISTEM KOMPUTER

Menyatakan Bahwa :

1. Sesungguhnya skripsi yang saya susun ini merupakan hasil karya tulis saya sendiri. Adapun bahagian-bahagian tertentu dalam skripsi yang saya peroleh dan hasil karya tulis orang lain, telah saya tuliskan sumbernya dengan jelas, sesuai dengan kaidah penulisan ilmiah.
2. Jika dalam pembuatan skripsi baik pembuatan program Maupun skripsi secara keseluruhan terbukti dibuatkan oleh orang lain, maka saya bersedia menerima sanksi yang diberikan akademik, berupa pembatalan skripsi dan mengulang penelitian serta mengajukan judul yang baru.

Demikianlah surat pernyataan ini saya buat dengan sesungguhnya tanpa ada paksaan dari pihak manapun.

Padang, Februari 2024



(Qadri Alfisyahri)
20101152620032

HALAMAN PENGESAHAN SKRIPSI

**RANCANG BANGUN SISTEM MONITORING AREA PARKIR
DAN PELACAkan POSISI KENDARAAN BERBASIS
MIKROKONTROLLER DAN COMPUTER VISION**

Yang dipersiapkan dan disusun oleh

QADRI ALFISYAHRI
20101152620032

Telah memenuhi persyaratan untuk dipertahankan di depan dewan penguji pada
ujian tahap akhir

Padang, 22 Februari 2024

Pembimbing I



(Okta Andrica Putra, S.Kom., M.Kom)
NIDN : 1005108604

Pembimbing II



(Ondra Eka Putra, S.Kom., M.Kom)
NIDN : 1006068701

HALAMAN PENGESAHAN PENGUJI SIDANG SKRIPSI

**RANCANG BANGUN SISTEM MONITORING AREA PARKIR
DAN PELACAKAN POSISI KENDARAAN BERBASIS
MIKROKONTROLLER DAN COMPUTER VISION**

OLEH :

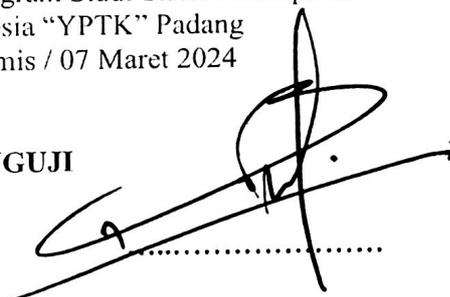
QADRI ALFISYAHRI
20101152620055

PROGRAM STUDI SISTEM KOMPUTER

Skripsi ini telah dinyatakan LULUS oleh Penguji Materi Pada Sidang Program
Studi Strata 1 Ilmu Komputer Program Studi Sistem Komputer
Universitas Putra Indonesia "YPTK" Padang
Pada Hari/Tanggal : Kamis / 07 Maret 2024

TIM PENGUJI

1. **Mardhiah Masril, S.Kom., M.Kom**
NIDN : 1012108401
2. **Anip Febriko, S.Kom., M.Kom**
NIDN : 1016028203



.....

Padang, 07 Maret 2024

Mengetahui
Dekan Fakultas Ilmu Komputer
Universitas Putra Indonesia YPTK Padang



(Prof. Dr. Yuhandri, S.Kom., M.Kom)
NIDN : 1015057301

HALAMAN PENGESAHAN LULUS SIDANG SKRIPSI

**RANCANG BANGUN SISTEM MONITORING AREA PARKIR
DAN PELACAkan POSISI KENDARAAN BERBASIS
MIKROKONTROLLER DAN COMPUTER VISION**

Yang dipersiapkan dan disusun oleh

QADRI ALFISYAHRI
20101152620032

Telah dipertahankan di depan dewan penguji
pada tanggal: 7 Maret 2024
dan dinyatakan telah lulus memenuhi syarat

Pembimbing I



(Okta Andrica Putra, S.Kom., M.Kom)
NIDN : 1005108604

Pembimbing II



(Ondra Eka Putra, S.Kom., M.Kom)
NIDN : 1006068701

Padang, 7 Maret 2024
Dekan Fakultas Ilmu Komputer
Universitas Putra Indonesia "YPTK" Padang



(Prof. Dr. Yuhandri, S.Kom, M.Kom)
NIDN : 1015057301