



UNIVERSITAS
Dinamika

**ALARM PENDETEKSI JARAK MOBIL PARKIR MENGGUNAKAN
SENSOR ULTRASONIK BERBASIS IOT**

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

UNIVERSITAS
Dinamika

Oleh:

FAHMI ARDIANSYAH PUTRA

18410200011

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

**ALARM PENDETEKSI JARAK MOBIL PARKIR MENGGUNAKAN
SENSOR ULTRASONIK BERBASIS IOT**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



UNIVERSITAS
Dinamika

Disusun Oleh:

Nama : Fahmi Ardiansyah Putra

NIM : 18410200011

Program Studi : S1 Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2023

Tugas Akhir
ALARM PENDETEKSI JARAK MOBIL PARKIR MENGGUNAKAN
SENSOR ULTRASONIK BERBASIS IOT

Dipersiapkan dan disusun oleh:

Fahmi Ardiansyah Putra

NIM : 18410200011

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: Kamis, 14 Juni 2023

Susunan Dewan Pembahas

Pembimbing:

I. Pauladie Susanto, S.Kom., M.T.

NIDN: 0729047501

II. Musayyanah, S.ST., M.T.

NIDN: 0730069102

Pembahas:

Heri Praktikno, M.T, MTCNA., MTCRE

NIDN: 0716117302



cn=Pauladie Susanto, o=FTI
Undika, ou=Prodi SI TK,
email=pauladie@dinamika.ac.id,
c=ID
2023.08.07 11:09:29 +07'00'



Digitally signed by Musayyanah
DN: cn=Musayyanah, ou=Universitas
Dinamika, ou=S1 Teknik Komputer,
email=musayyanah@dinamika.ac.id,
c=ID
Date: 2023.08.07 12:19:16 +07'00'
Adobe Acrobat Reader version:
2023.003.20244



Digitally signed by Heri Praktikno, M.T.
DN: cn=Heri Praktikno, M.T.,
o=Universitas Dinamika, ou=S1 Teknik
Komputer, email=heri@dinamika.ac.id,
c=ID
Date: 2023.08.09 15:34:02 +07'00'
Adobe Acrobat version: 11.0.23

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



Digitally signed by
Universitas
Dinamika
Date: 2023.08.16
08:31:02 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

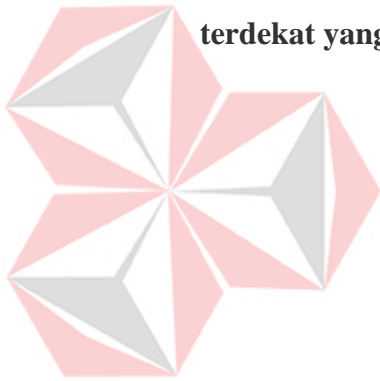
“Berbuat baik itu sejatinya membahagiakan hati diri sendiri”

~ Fahmi Ardiansyah Putra ~



UNIVERSITAS
Dinamika

Dipersembahkan kepada Orang Tua, Keluarga dan Teman Saya yang telah mendukung dan memberi motivasi dalam pengerjaan Tugas Akhir ini. Dosen yang telah membantu dan membimbing Saya. Serta seluruh orang-orang terdekat yang selalu memberikan bantuan, dukungan, dan motivasi kepada Saya.



UNIVERSITAS
Dinamika

SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, Saya :

Nama : Fahmi Ardiansyah Putra
NIM : 18410200011
Program Studi : SI Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Tugas Akhir
Judul Karya : **ALARM PENDETEKSI JARAK MOBIL PARKIR
MENGUNAKAN SENSOR ULTRASONIK BERBASIS IOT**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royal Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut di atas adalah karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 12 Juni 2023

Yang menyatakan



Fahmi Ardiansyah Putra

NIM : 18410200011

ABSTRAK

Parkir mobil pasti akan dilakukan oleh pengendara mobil untuk menempatkan mobilnya dan pengendara akan memilih tempat parkir yang memadai. Namun terdapat beberapa area parkir tidak memiliki lahan yang luas, hanya cukup untuk beberapa mobil saja. Pada area parkir yang luas dan memadai, terkadang pengendara mengalami kesulitan, terutama pada area parkir yang sempit. Pengendara juga mengalami kesulitan saat memarkirkan mobilnya dengan cara mundur. Ketika memprediksi jarak aman objek yang berada pada bagian belakang mobil dengan mobil tersebut, seringkali pengendara mengalami kesalahan yang menyebabkan benturan pada *body* mobil. Kesalahan yang terjadi dapat menyebabkan kerugian bagi pengendara. Berdasarkan masalah yang sering dialami oleh pengendara mobil ketika memarkirkan mobil, penulis bertujuan membuat *prototype* yang dapat memprediksi batas jarak aman mobil ketika parkir. Pembuatan *prototype* tersebut menggunakan LCD untuk menampilkan angka jarak aman dan *buzzer* berupa suara 'beep' sebagai peringatan, semakin dekat dengan objek maka suara yang dikeluarkan akan semakin cepat. Serta pengendara dapat mengontrol dan memonitoring alat dari *smartphone* terkait jarak yang diinginkan. Menggunakan metode *moving average* untuk mengambil 5 data terbaru dan dibagi dengan data n , kemudian menghasilkan nilai rata-rata yang dapat meningkatkan nilai keakuratan. Dari penelitian yang telah dilakukan, alat yang dirancang dapat melakukan komunikasi menggunakan MQTT *Broker*, hasil akurasi pada pengujian komunikasi MQTT sebesar 93.33%. Terdapat 3 sensor ultrasonik sebagai pendeteksi jarak aman mobil, dilakukan pengujian sebanyak 30 kali pada setiap sensor ultrasonik. Pada sensor ultrasonik 1 didapatkan rata-rata *error* untuk jarak 35 cm sebesar 0,8% , untuk 25 cm sebesar 5,96%, untuk 15 cm sebesar 3,37%, untuk 7 cm sebesar 0,5%, untuk 3 cm sebesar 7,77%. Pada sensor ultrasonik 2 didapatkan rata-rata *error* untuk jarak 35 cm sebesar 1,28%, untuk 25 cm sebesar 0%, untuk 15 cm sebesar 0,93%, untuk 7 cm sebesar 0,52%, untuk 3 cm sebesar 6%. Pada sensor ultrasonik 3 didapatkan rata-rata *error* untuk jarak 35 cm sebesar 0,57%, untuk 25 cm sebesar 0,49%, untuk 15 cm 1,3%, untuk 7 cm sebesar 4,16%, untuk 3 cm sebesar 3,39%. Hal ini dapat disimpulkan bahwa *moving average* dapat meningkatkan akurasi deteksi jarak oleh ultrasonik.

Kata Kunci : *Parkir Mobil, Sensor Ultrasonik, IoT*

KATA PENGANTAR

Puji syukur Saya panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul “Alarm Pendeteksi Jarak Mobil Parkir Menggunakan Sensor Ultrasonik Berbasis IoT”. Penulisan laporan ini sebagai salah satu syarat untuk lulus dan mendapatkan gelar Sarjana Teknik.

Dalam menyelesaikan penulisan laporan ini, penulis banyak mendapat bantuan dari berbagai pihak yang memberi dukungan dan membimbing. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa karena dengan rahmatnya dan hidayahnya penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Orang Tua dan seluruh keluarga penulis tercinta yang telah memberikan dorongan dan bantuan baik moril maupun materi, sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Dosen Program Studi S1 Teknik Komputer sekaligus pembimbing 1 penulis. Terima kasih atas izin dan bimbingan yang diberikan dan kesempatannya serta tuntunan baik itu materi secara tertulis maupun lisan.
4. Ibu Musayyanah, S.ST., M.T., selaku Dosen Pembimbing 2 yang selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembahas yang selalu memberi waktu dan bimbingan dalam menyelesaikan laporan Tugas Akhir ini.
6. Dianita Azizah Ariesanty yang telah memberikan dukungan, motivasi dan mendampingi dalam pengerjaan Tugas Akhir ini.
7. Seluruh teman-teman S1 Teknik Komputer angkatan 2018 yang memberikan semangat pantang menyerah dan selalu menemani selama proses mengerjakan Tugas Akhir ini.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu penulis sangat mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk menjadi lebih baik lagi di masa yang akan datang.

Surabaya, 10 Juni 2023

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	3
BAB II LANDASAN TEORI	4
2.1 Sensor Ultrasonik HC-SR04	4
2.2 WEMOS D1 R32	8
2.3 <i>Buzzer</i>	9
2.4 LCD 20x4 dan I2C	10
2.5 MQTT	11
2.6 <i>Push Button Switch</i>	13
2.7 <i>Moving Average (MA)</i>	14
BAB III METODOLOGI PENELITIAN	15
3.1 Perancangan <i>Hardware</i>	15
3.2 Perancangan Rangkaian Skematik	16
3.3 Setting Pengguna	18
3.4 <i>Flowchart</i>	19
3.5 Desain Model Prediksi Jarak Parkir Mobil	22
BAB IV HASIL DAN PEMBAHASAN	23

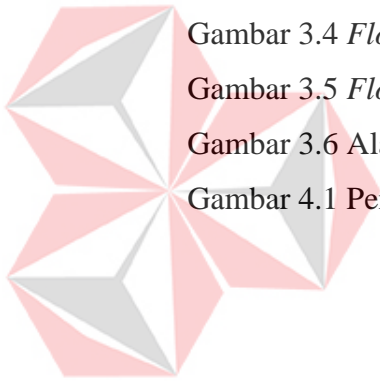
4.1 Pengujian Komunikasi MQTT	23
4.2 Pengujian Pembacaan Sensor Ultrasonik	26
4.3 Tabel Kebenaran jarak.....	32
BAB V PENUTUP.....	35
5.1 Kesimpulan.....	35
5.2 Saran	36
DAFTAR PUSTAKA	37
LAMPIRAN.....	39
BIODATA PENULIS	70



UNIVERSITAS
Dinamika

DAFTAR GAMBAR

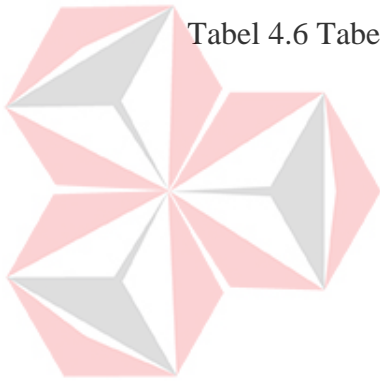
Gambar 2.1 Sensor Ultrasonik HC-SR04	4
Gambar 2.2 <i>Timing Diagram</i> Sensor Ultrasonik HC-SR04	5
Gambar 2.3 Ilustrasi Cara Kerja Sensor Ultrasonik.....	7
Gambar 2.4 Wemos D1 R32	9
Gambar 2.5 <i>Buzzer</i>	10
Gambar 2.6 LCD 20x4 dan I2C	11
Gambar 2.7 Sistem MQTT Menggunakan IoT	12
Gambar 2.8 <i>Push Button Switch</i>	14
Gambar 3.1 Blok Diagram	15
Gambar 3.2 Rangkaian Skematik.....	16
Gambar 3.3 <i>Setting</i> Pengguna.....	18
Gambar 3.4 <i>Flowchart</i> Pendeteksi Jarak Mobil Parkir.....	19
Gambar 3.5 <i>Flowchart</i> Sistem MQTT	21
Gambar 3.6 Alat Pendeteksi Jarak Mobil Parkir.....	22
Gambar 4.1 Pengujian pada Aplikasi MQTT Tool.....	25



UNIVERSITAS
Dinamika

DAFTAR TABEL

Tabel 2.1 Spesifikasi Sensor Ultrasonik HC-SR04.....	5
Tabel 2.2 Tabel Pin Sensor Ultrasonik HC-SR04.....	6
Tabel 2.3 <i>Library</i> Sensor Ultrasonik.....	7
Tabel 2.4 <i>Library</i> LCD.....	11
Tabel 2.5 <i>Library</i> MQTT	12
Tabel 3.1 Pin <i>input output</i> dari komponen rangkaian	16
Tabel 4.1 Hasil Pengujian MQTT	23
Tabel 4.2 Tabel Hasil Pengujian Sensor Ultrasonik 1	27
Tabel 4.3 Tabel Hasil Pengujian Sensor Ultrasonik 2	29
Tabel 4.4 Tabel Hasil Pengujian Sensor Ultrasonik 3	30
Tabel 4.5 Tabel <i>setting default</i> jarak mobil dengan objek.....	33
Tabel 4.6 Tabel Hasil Pengujian Paralel	33



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

Lampiran 1 <i>Source Code</i> Program.....	39
Lampiran 2 Hasil Plagiasi	64
Lampiran 3 Foto Keseluruhan Alat.....	67
Lampiran 4 Penempatan Mikrokontroler dan Kabel.....	67
Lampiran 5 Keseluruhan Alat Tampak Samping.....	68
Lampiran 6 Keseluruhan Alat Tampak Depan.....	68
Lampiran 7 Keseluruhan Alat Tampak Belakang Dengan Miniatur Mobil.....	69



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pengendara mobil akan menempatkan mobilnya di tempat parkir. Namun tidak semua area parkir memiliki lahan yang luas, terkadang area parkir hanya cukup untuk beberapa mobil saja. Hal tersebut membuat pengendara merasa kesulitan ketika akan memarkirkan mobilnya di area parkir. Terutama ketika akan menempatkan posisi mobil dengan cara memundurkan mobil. Karena adanya beberapa mobil, terutama mobil keluaran tahun lama yang tidak sama seperti dengan mobil keluaran terbaru, yang sudah memiliki sensor pendeteksi jarak di bagian belakang mobil.

Pengendara mengalami kesulitan untuk memprediksi jarak belakang mobil dengan objek seperti dinding ataupun benda yang berada pada belakang mobil. Jika tidak dapat memprediksi dengan benar, maka akan terjadi benturan pada *body* mobil. Benturan tersebut dapat menyebabkan kerusakan pada mobil yang cukup parah, tergantung benturan yang terjadi. Jika mobil mengalami kerusakan maka pengendara akan mengalami kerugian karena harus memperbaiki kerusakan yang terjadi.

Kesalahan pengendara dalam memprediksi jarak pada mobil merupakan sebuah masalah yang sering terjadi. Untuk mengurangi permasalahan tersebut maka dibutuhkan alat yang dapat memprediksi jarak mobil dengan objek yang berada di belakang mobil tersebut. Alat tersebut yaitu “Alarm Pendeteksi Jarak Mobil Parkir Menggunakan Sensor Ultrasonik Berbasis IoT”. Alat yang akan memudahkan pengendara ketika memarkirkan mobilnya di area parkir. Alat tersebut akan mengeluarkan suara peringatan, jika jarak mobil semakin dekat dengan objek maka suara yang dikeluarkan akan semakin keras dan cepat.

Pada penelitian sebelumnya yang dituliskan oleh Dimas Attala Naoval pada jurnal Universitas Dinamika. Pada penelitian yang dituliskan oleh Dimas, menghasilkan fitur jarak aman menggunakan sensor ultrasonik. Alat tersebut terdapat pada *body* mobil sebagai *input* untuk mendeteksi jarak aman mobil listrik.

Pada penelitian tersebut sensor ultrasonik telah digunakan untuk mendeteksi jarak aman (Naoval, 2022). Namun sensor digunakan untuk aplikasi pengereman, sedangkan pada penelitian ini, sensor ultrasonik untuk aplikasi *warning system* pada parkir.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan masalah pada Tugas Akhir ini sebagai berikut :

1. Bagaimana menjaga jarak aman ketika memarkirkan mobil di area parkir?
2. Bagaimana mengetahui batas jarak aman mobil parkir dengan objek?
3. Bagaimana mengontrol jarak aman mobil parkir dari jarak jauh?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, pembahasan masalah dibatasi pada beberapahal berikut :

1. Alat yang dirancang merupakan sebuah *prototype*.
2. Pendeteksi hanya dapat digunakan pada area parkir yang berlangganan.
3. *Output* yang dihasilkan berupa bunyi '*beep*' dan tampilan pada LCD.
4. Sistem kontrol jarak jauh dapat dilakukan pada *smartphone*, dan dapat bekerja jika terhubung dengan internet.

1.4 Tujuan

Berdasarkan latar belakang dan rumusan masalah di atas, mendapatkan tujuan pada Tugas Akhir ini sebagai berikut :

1. Menjaga jarak aman ketika memarkirkan mobil di area parkir.
2. Mengetahui batas jarak aman mobil parkir dengan objek.
3. Mengontrol jarak aman mobil parkir dari jarak jauh.

1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh manfaat sebagai berikut :

1. Memudahkan pengemudi ketika akan memarkirkan mobilnya di area parkir.
2. Admin dapat mengontrol jarak aman mobil parkir dari *smartphone*.
3. Memberikan informasi kepada pengendara terkait batas jarak menggunakan suara, semakin cepat suara yang dikeluarkan maka semakin dekat jarak mobil dengan objek.

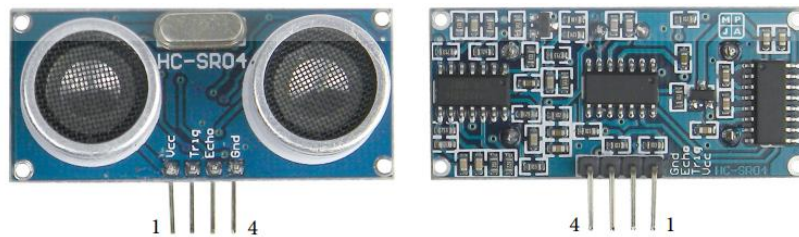


UNIVERSITAS
Dinamika

BAB II LANDASAN TEORI

2.1 Sensor Ultrasonik HC-SR04

Sensor Ultrasonik HC-SR04 merupakan alat sensor yang dapat digunakan untuk mendeteksi suatu objek yang terdapat di depan sensor tersebut. Sensor bekerja sesuai konsep pantulan dari gelombang suara. Memiliki dua unit, unit tersebut diantaranya adalah unit pemancar dan unit penerima. Kristal *piezoelectric* merupakan sebuah struktur dari dua unit tersebut, yang menghubungkan mekanik jangkar lalu dihubungkan oleh diafragma penggetar. Frekuensi kerja gelombang suara sebesar 40 KHz - 400 KHz. Struktur atom yang ada pada *piezoelectric* akan mengikat lalu mengembang ataupun menyusut pada tegangan yang telah terjadi. Tegangan tersebut merupakan efek *piezoelectric*. Kemudian diteruskan ke diafragma penggetar yang menyebabkan terjadinya gelombang ultrasonik yang dipancarkan ke udara. Apabila ada objek di depannya maka gelombang ultrasonik akan terjadi, kemudian unit sensor penerima akan menerima kembali gelombang ultrasonik. Selanjutnya mengirimkan getaran, dan yang terjadi efek *piezoelectric* akan mengeluarkan sebuah tegangan bolak-balik (Shaputra, Gunoto, & Irsyam, 2019).



Gambar 2.1 Sensor Ultrasonik HC-SR04
(Sumber: Alawiah, dkk, 2017)

Sensor ultrasonik menggunakan dasar pantulan dari gelombang ultrasonik. Saat gelombang dikeluarkan oleh sensor ultrasonik, yang terjadi yaitu gelombang memantul, dan sensor memberikan data kepada mikrokontroler. Pada Gambar 2.2 adalah *timing diagram* dari prinsip kerja sensor ultrasonik. Pada periode 10 us, sensor ultrasonik akan bekerja jika dipicu menggunakan pulsa. Setelah itu, sensor mengeluarkan gelombang ultrasonik 8 siklus dengan frekuensi 40 kHz. Saat gelombang telah dipantulkan, penerima memberikan sinyal pulsa ke

mikrokontroler. Maka dari itu, jarak pantulan antara sensor dengan objek dapat dihitung dengan menggunakan persamaan (1) dan (2) sesuai dengan *timing diagram*. Rumus deteksi jarak antara sensor dengan objek adalah sebagai berikut:

$$d(cm) = \frac{\tau}{58} \quad (1)$$

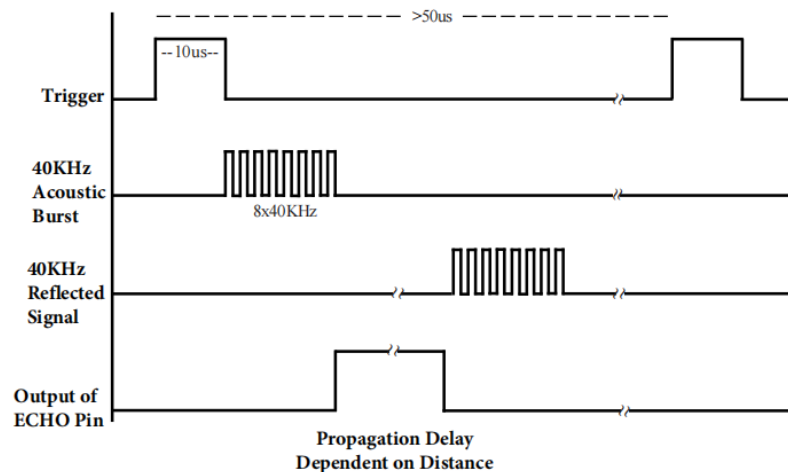
$$d(inch) = \frac{\tau}{148} \quad (2)$$

Diketahui d merupakan jarak dan τ yaitu periode *high* sinyal ultrasonik pada mikrodetik (Alawiah & Al Tahtawi, 2017).

Tabel 2.1 Spesifikasi Sensor Ultrasonik HC-SR04

Parameter	Nilai
Tegangan Kerja	5 V (DC)
Arus Kerja	15 mA
Frekuensi Kerja	40 kHz
Jarak Maksimum	4 m
Jarak Minimum	2 cm
Sudut Pengukuran	15 derajat
Sinyal Input Trigger	10 us pulsa TTL
Sinyal Output Echo	TTL level signal, proporsional terhadap jarak
Dimensi	1-13/16" x 13/16" x 5/8"
Koneksi	4 pin (Vcc, Gnd, Echo, Trigger)

(Sumber: Datasheet HC-SR04)



Gambar 2.2 *Timing Diagram* Sensor Ultrasonik HC-SR04
(Sumber: Datasheet HC-SR04)

Kemampuan deteksi jarak dari sensor ultrasonik, minimal sebesar 2 cm dan maksimal sebesar 400 cm, dengan tingkat ketepatan sebesar 0,3 cm. Membutuhkan tegangan +5V dan arus dibawah angka 2mA. Sudut deteksi yang dapat ditangani dibawah 15°. Sensor ultrasonik memiliki pin yang berjumlah 4, keterangan pin sensor ultrasonik dapat dilihat pada Tabel 2.1.

Tabel 2.2 Tabel Pin Sensor Ultrasonik HC-SR04

Pin	Keterangan
Pin 1	VCC (terhubung dengan tegangan +5V)
Pin 2	Trig (mengirim gelombang suara)
Pin 3	Echo (menerima pantulan gelombang suara)
Pin 4	Gnd (terhubung dengan ground)

(Sumber: Purwanto, dkk, 2019)

Jarak sensor dengan objek yang menghasilkan pantulan gelombang suara dapat dihitung menggunakan Persamaan (3):

$$Jarak = Kecepatan\ suara * \frac{T}{2} \quad (3)$$

T merupakan waktu yang ditempuh ketika sinyal telah dihasilkan sampai kembali. Dengan kecepatan suara sebesar 343 m/s. Prinsip pengiriman sinyal dari trig dan penerimaan sinyal dari echo adalah sebagai berikut:

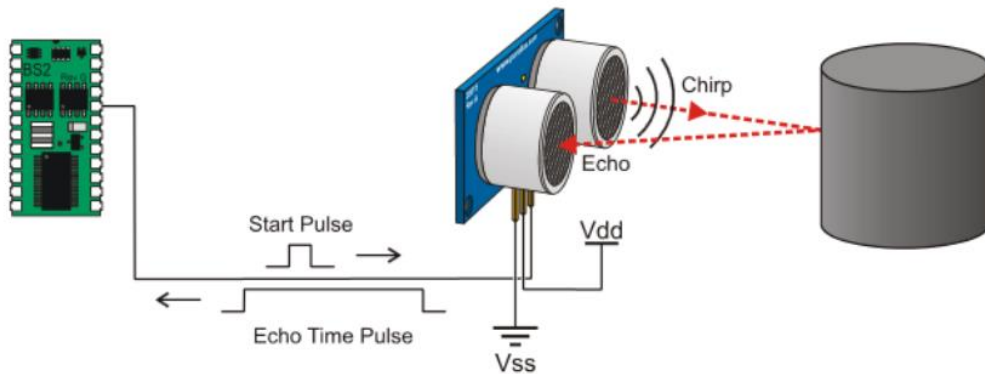
- a. Memastikan bahwa trig harus dengan keadaan HIGH, minimal selama 10ms.
- b. Gelombang kotak dengan frekuensi sebesar 40 KHz akan dikirimkan oleh modul ultrasonik.
- c. Modul ultrasonik akan memantau secara otomatis terkait gelombang yang dikirim. Maka dari itu, T merupakan waktu yang didapatkan ketika sinyal terkirim hingga sinyal dapat diterima kembali. Pada saat itu pin echo pada posisi HIGH. Waktu T bisa didapatkan dengan memberi perintah pada Arduino.

$$T = \text{pulseIn} (\text{PIN_ECHO}, \text{HIGH});$$

- d. Setelah T berhasil didapatkan, maka dapat dihitung menggunakan rumus yang ada.

Tujuan dari rumus T dibagi 2 dikarenakan T merupakan waktu yang diperoleh untuk menempuh jarak dari sensor ke objek dan sebaliknya. Dengan nilai kecepatan suara yaitu 343 m/s atau 343000 cm/s, maka dapat diperoleh seperti Persamaan (4):

$$\begin{aligned}
 \text{Jarak} &= 343000 * \frac{\left(\frac{T}{10-6}\right)}{2 \text{ cm}} \\
 &= 0,0343 * \frac{T}{2 \text{ cm}} \quad (4)
 \end{aligned}$$



Gambar 2.3 Ilustrasi Cara Kerja Sensor Ultrasonik HC SR-04
(Sumber : Purwanto, dkk, 2019)

Pada Gambar 2.2 merupakan ilustrasi cara kerja sensor ultrasonik, mulai dari sinyal yang dikirim oleh pin Trig sampai sinyal diterima kembali oleh pin Echo (Purwanto, Riyadi, Astuti, & Kusuma, 2019).

Pada sistem yang telah dirancang agar dapat mengakses dan membuat sensor ini dapat bekerja sesuai rumusnya, mulai dari sinyal yang dikirim oleh pin Trig dan diterima kembali oleh pin Echo, maka dibutuhkan *library* atau modul untuk Arduino Uno bernama ultrasonik HC-SR04 by Erick Simoes. Fungsi yang digunakan adalah untuk pinMode trig untuk mengirim sinyal sampai diterima kembali oleh pinMode Echo. Beberapa *Function* yang digunakan sebagai berikut:

Tabel 2.3 *Library* Sensor Ultrasonik

Nama <i>Function</i>	Penjelasan
<code>#include <Ultrasonic.h></code>	<i>Library</i> yang digunakan untuk memudahkan dalam menjalankan program untuk sensor ultrasonik.
<code>pinMode(trigPin1, OUTPUT);</code>	Baris program ini berfungsi untuk inialisasi pin triggerPin sebagai <i>output</i> .
<code>pinMode(echoPin1, INPUT);</code>	Baris program ini berfungsi untuk inialisasi pin echoPin sebagai <i>input</i> .
<code>Serial.begin(115200);</code>	Baris program ini berfungsi untuk memulai komunikasi serial pada laptop.

Nama <i>Function</i>	Penjelasan
<code>digitalWrite(trigPin1, LOW);</code> <code>delayMicroseconds(10);</code>	Baris program ini berfungsi untuk memberi logika LOW pada triggerPin selama 2 mikro detik.
<code>digitalWrite(trigPin1, HIGH);</code> <code>delayMicroseconds(10);</code>	Baris program ini berfungsi untuk memberi logika HIGH pada triggerPin selama 10 mikro detik.
<code>digitalWrite(trigPin1, LOW);</code>	Baris program ini berfungsi memberi logika LOW pada triggerPin.
<code>duration1 = pulseIn(echoPin1, HIGH);</code>	Baris program ini berfungsi untuk menghitung waktu yang digunakan suara dari mulai ditembakkan oleh <i>transmitter</i> hingga suara memantul dan kembali ke <i>receiver</i> .
<code>distanceCm1 = duration1 / 58.2;</code>	Baris program ini berfungsi untuk memperoleh jarak antara sensor dengan benda di depannya. 58,2 adalah kecepatan suara (58.2 uS/cm).

2.2 WEMOS D1 R32

Wemos D1 R32 merupakan mikrokontroler berbasis ESP32 yang mempunyai kecepatan *clock* sebesar 240Mhz yang berbasis *wifi*. Pada Wemos D1 R32 terdapat 32 pin yang terpasang, terdapat pin digital dan analog yang disambungkan ke sumber menggunakan kabel USB. Mikrokontroler tersebut diprogram melalui aplikasi Arduino IDE dan menggunakan bantuan *library* yang terdapat di aplikasi tersebut (Zuhri & Okselia, 2021).

Wemos D1 R32 bentuknya menyerupai Arduino Uno, tetapi lebih unggul Wemos D1 R32 dari spesifikasinya. Komponen inti Wemos D1 R32 mempunyai *Dual-Core 32 bit*, sedangkan pada Arduino Uno memiliki 8 *bit*.

Fitur Wemos D1 R32 :

- Kinerja tinggi dan harga yang terjangkau.
- Bentuk yang kecil, mudah disematkan dengan produk yang lain.
- Dc 5V-12V
- 1 analog *input* (3.2V *max input*)
- Koneksi USB *micro*
- Kompatibel dengan Arduino
- D1 R32 : Wifi+Bluetooth+UNO (Khotimah, 2021)



Gambar 2.4 Wemos D1 R32
Sumber: (Zuhri, dkk, 2021)

2.3 Buzzer

Buzzer merupakan suatu alat elektronika yang dapat mengubah sinyal listrik dan menghasilkan getaran suara. Aliran listrik yang mengalir pada rangkaian *buzzer* menyebabkan terjadinya pergerakan mekanis. Terjadi perubahan energi listrik yang kemudian berubah menjadi energi suara yang dapat didengar oleh siapapun. *Buzzer piezoelectric* bekerja pada tegangan 3 - 12 volt DC. Terdapat 2 jenis *buzzer* berdasarkan bunyinya, diantaranya *Active Buzzer* dan *Passive Buzzer*. *Active Buzzer* sering disebut *buzzer* yang berdiri sendiri atau *buzzer stand alone*, karena telah mempunyai suara sendiri ketika disambungkan dengan listrik. Sedangkan *Passive Buzzer* yaitu *buzzer* yang cocok ketika digabungkan dengan Arduino karena tinggi rendahnya nadanya dapat diatur, *buzzer* ini tidak mempunyai suara sendiri.

Fungsi *buzzer* untuk menghasilkan *output* berupa suara peringatan seperti bunyi 'beep'. *Buzzer* digunakan sebagai indikator suara, *timer*, dan alarm. Buzzer terdiri dari 3 komponen, diantaranya adalah *Piezoelectric*, sumber bunyi yang dikeluarkan melalui komponen tersebut, berwarna hitam dan berbentuk tabung. Terdapat juga kaki pin *negative* dan kaki pin *positive*. Kaki pin *negative* adalah kaki yang lebih pendek dan digunakan untuk disambungkan dengan arus *negative*/GND. Sedangkan kaki pin *positive*, kaki yang lebih panjang yang dihubungkan pada arus positif atau VCC/5V (Setiadi, 2021).



Gambar 2.5 *Buzzer*
(Sumber: Setiadi, 2021)

2.4 LCD 20x4 dan I2C

LCD (*Liquid Crystal Display*) adalah sebuah alat yang sering digunakan sebagai perangkat penampil. Untuk menghasilkan cahaya, konstruksi LCD menggunakan *gallium* atau silikon yang berbentuk kristal cair. Terdapat 4 operasi dasar yang ada pada LCD, diantaranya adalah mengakses proses *internal*, penulisan data, pembacaan kondisi, dan selanjutnya pembacaan data. Kelebihan LCD yaitu ukuran yang proporsional, memudahkan dalam pembuatan program tampilan karena dapat menampilkan karakter ASCII, menggunakan daya yang kecil, hanya menggunakan 8 bit data dan 3 bit *control* sehingga mudah terhubung dengan *port I/O*.

I2C (*Inter Integrated Circuit*) adalah tambahan *chip* modul yang digunakan untuk memudahkan *programmer* mengakses LCD. I2C terdiri dari dua saluran utama, diantaranya adalah saluran SCL (*serial clock*) dan SDA (*serial data*). Saluran utama sebagai pembawa informasi terkait data antara I2C dengan sistem kontrol. Perangkat yang terhubung pada I2C sebagai *master/slave*. *Master* merupakan perangkat yang melakukan pengiriman data dengan menggunakan sinyal *stop*, lalu menaikkan sinyal *clock*. *Master* memberikan alat tersebut kepada perangkat yang bernama *slave* (Thayeb, 2018).



Gambar 2.6 LCD 20x4 dan I2C

(Sumber: Thayeb, 2018)

Pada sistem yang dirancang agar dapat mengakses dan membuat LCD dapat berjalan dengan sesuai, maka dibutuhkan *library* atau modul untuk Arduino Uno bernama *LiquidCrystal* by Arduino. Ada beberapa *Function* yang digunakan untuk menjalankan LCD I2C kolom dan baris 20x4. Beberapa *Function* yang digunakan beserta penjelasannya sebagai berikut:

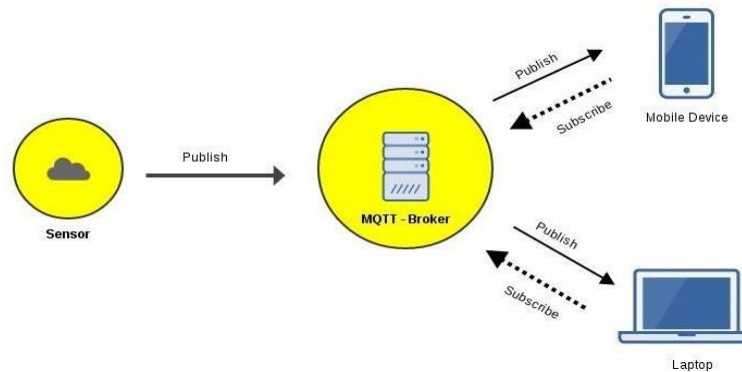
Tabel 2.4 *Library* LCD

Nama <i>Function</i>	Penjelasan
<code>#include <LiquidCrystal_I2C.h</code>	Library yang digunakan untuk memudahkan dalam menjalankan program untuk LCD 20x4.
<code>LiquidCrystal_I2C lcd(0x27, lcdkolom, lcdbaris);</code>	Untuk inisialisasi LCD_I2C, Lcd kolom dan baris.
<code>lcd.init();</code>	Berfungsi untuk inisialisasi LCD.
<code>lcd.backlight();</code>	Berfungsi untuk menghidupkan backlight LCD.
<code>lcd.setCursor(0, 0);</code>	Berfungsi untuk mengatur atau menandai alamat dimulai dari baris dan kolom beberapa.
<code>lcd.print("P1:");</code> <code>lcd.print(moving1);</code>	Untuk menampilkan output yang berupa tulisan dan hasil perhitungkan Moving Average.

2.5 MQTT

MQTT adalah singkatan dari *Message Queuing Telemetry Transport*, merupakan protokol yang berguna untuk menghubungkan dengan IoT (*Internet of Things*) yang didesain secara sederhana, mudah diimplementasikan, terbuka dan ringan. Protokol MQTT bergerak diatas *stack* TCP/IP, memiliki kapasitas paket data yang kecil sehingga berpengaruh pada konsumsi daya yang kecil. Data yang

dikirimkan pada protokol ini yaitu *teks*, data *binary*, *xml* atau *JSON*. Protokol MQTT menggunakan model *publish/subscribe* dibandingkan model *client-server* (Putra, Bhawiyuga, & Data, 2018).



Gambar 2.7 Sistem MQTT Menggunakan IoT
(Sumber: Putra, dkk, 2018)

Pada sistem yang dirancang agar dapat menjalankan *system* MQTT maka dibutuhkan *library* yang bernama PubSubClient by Nick o'Leary, menyesuaikan *library* dengan Arduino yang dipakai. Penulis menggunakan Arduino Uno berbasis ESP8266, sehingga *library* ini cocok untuk digunakan untuk ESP8266. *Library* ini berfungsi untuk pengaturan awal untuk menghubungkan Arduino Uno dengan internet, lalu jika internet terputus maka otomatis akan menghubungkan kembali dengan internet. Fungsi selanjutnya yaitu untuk menginputkan data melalui aplikasi MQTT Tool yang dikirim melalui *cloud* dan diterima oleh Arduino, begitu pula sebaliknya, jika Arduino Uno mengirim data melalui *cloud* dan diterima oleh MQTT Tool. Beberapa *Function* yang di gunakan sebagai berikut :

Tabel 2.5 *Library* MQTT

Nama <i>Function</i>	Penjelasan
<code>#include <WiFi.h></code>	<i>Library</i> yang digunakan untuk mengakses <i>WiFi</i> .
<code>#include <PubSubClient.h></code>	<i>Library</i> yang digunakan untuk membangun komunikasi.
<code>const char* ssid = "VERAXAZ"; const char* password = "11111111";</code>	Dipergunakan untuk <i>login</i> ssid dan <i>password</i> agar terhubung dengan koneksi internet.
<code>const char* mqtt_server = "broker.hivemq.com";</code>	Untuk memberikan konfigurasi alamat <i>cloud/broker</i> .

Nama <i>Function</i>	Penjelasan
<code>client.setServer(mqtt_server, 1883);</code>	Merupakan pengaturan untuk mengatur server yang digunakan dengan parameter nilai <i>variable</i> <code>mqtt_server</code> dan <i>port</i> 1883.
<code>void setup_wifi() {}</code>	Merupakan <i>code</i> fungsi <i>void</i> <code>setup_wifi()</code> untuk mengatur proses koneksi mikrokontroler Wemos D1 R32 dengan jaringan internet dan MQTT, yaitu dengan mengambil nilai variabel <code>ssid</code> dan <i>password</i> yang telah dideklarasikan sebelumnya.
<code>void reconnect() {}</code>	Merupakan <i>code</i> fungsi <i>void</i> <code>reconnect()</code> untuk mengatur kembali proses koneksi mikrokontroler Wemos D1 R32 dengan jaringan internet dan MQTT, jika koneksi internet terganggu dan tidak bisa terhubung.
<code>void callback(char* topic, byte* payload, unsigned int length) {}</code>	Merupakan <i>code</i> fungsi <i>void</i> <code>callback()</code> untuk menerima <i>topic</i> yang dikirim melalui MQTT dan untuk membandingkan beberapa <i>topic</i> yg diterima, atau berfungsi untuk menerima nilai balik (<i>subscribe</i>).

2.6 Push Button Switch

Push button switch merupakan sebuah saklar tekan yang digunakan untuk memisahkan atau menghubungkan komponen dari sebuah instalasi listrik dengan instalasi listrik yang lain. Saklar tekan terdiri dari *start*, *stop*, *reset*, dan *emergency*. *Push button switch* mempunyai kontak *normally close*, sedangkan *switch no* yaitu *normally open*. *Push button switch* dengan sistem kerja tidak mengunci (*unlock*) yang digunakan untuk menghubungkan/memutuskan aliran listrik. Sistem kerja *unlock* yaitu saklar sebagai pelaku utama untuk menghubungkan/memutuskan aliran listrik ketika tombol ditekan, dan jika tombol dilepas maka akan kembali seperti semula (Widodo, Irawan, & Prastowo, 2020).

Perangkat ini menjadi penghubung dan pemutus, pada *push button switch* memiliki 2 keadaan, yaitu posisi *on* dan *off* (1 dan 0). Semua sumber energi listrik yang menggunakan perangkat listrik, pasti sangat membutuhkan keadaan *On* dan *Off*, maka dari itu hal tersebut sangat penting. *Push button switch* merupakan

perangkat paling utama yang sering digunakan untuk memulai dan mengakhiri kerja mesin di industri, karena cara kerjanya yang langsung terhubung dengan operator dan *unlock*. Semua mesin secanggih apapun, pasti akan menggunakan saklar seperti *push button switch* atau sejenis lainnya yang dapat mengatur kondisi *On* dan *Off* (Sutono & Nursoparisa, 2019).



Gambar 2.8 *Push Button Switch*
(Sumber: Sutono, dkk, 2019)

2.7 *Moving Average* (MA)

Moving average adalah metode rata-rata tunggal yang menggunakan beberapa data aktual permintaan terbaru agar meningkatkan nilai ramalan yang digunakan untuk permintaan pada masa yang akan datang. Terdapat dua sifat khusus pada metode ini, yaitu pada pembuatan *forecast* diperlukan data historis dalam rentan waktu tertentu, jika *moving average* semakin panjang maka *moving averages* yang dihasilkan akan semakin halus. Persamaan dari *moving average* adalah (1) :

$$F_t = A_{t-1} + A_{t-2} + A_{t-3} \dots / n \quad (1)$$

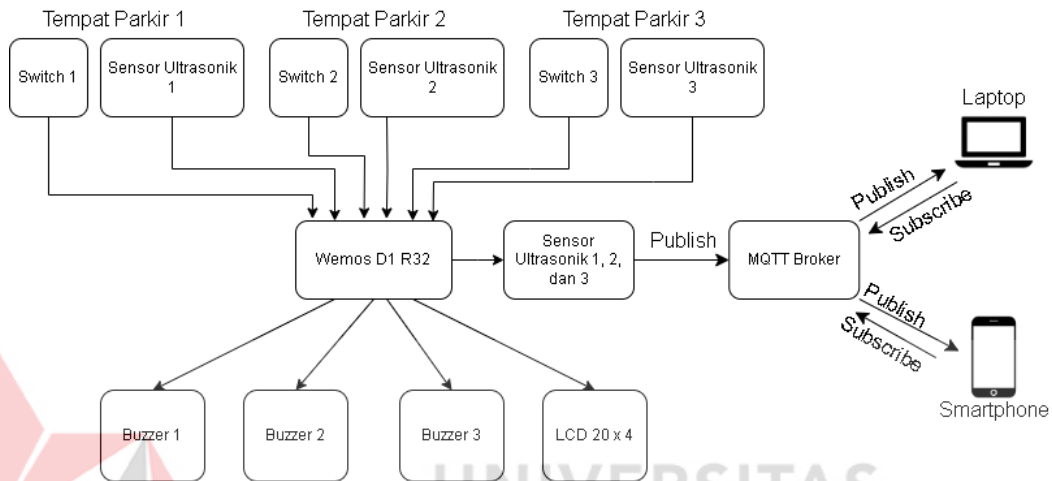
F_t adalah periode yang akan diperiksa nilai ramalannya. Sedangkan $A_{t-1} + A_{t-2} + A_{t-3} \dots$ adalah data aktual pada periode sebelumnya yang akan dicari nilai ramalannya. Serta n merupakan jumlah periode yang digunakan sebagai perhitungan peramalan pada periode tertentu (Maricar, 2019).

Moving average merupakan salah satu jenis *filter* data yang diimplementasikan pada pembacaan sensor. Berfungsi untuk memperbaiki sinyal dan dapat mengurangi *noise* sehingga *output* yang dihasilkan akan lebih menarik dan dapat dipahami dengan mudah. Sistem kerja dari MA yaitu menggunakan nilai rata-rata dari kumpulan data yang diterima pada waktu tertentu. Filter data tersebut akan membuang *outers* data, maka data yang diperoleh ialah data yang nilainya relevan (Trisna & Maharani, 2022).

BAB III METODOLOGI PENELITIAN

3.1 Perancangan *Hardware*

Dibawah ini merupakan gambar perancangan perangkat keras pada alat pendeteksi jarak mobil parkir.



Gambar 3.1 Blok Diagram

Dari rancangan bangun alarm pendeteksi jarak mobil parkir terdiri dari beberapa komponen utama. Cara kerja atau blok diagram pada alat ini adalah sebagai berikut:

1. *Input*

- a. Sensor Ultrasonik 1, 2, dan 3 ini adalah *input* untuk memberitahukan mendeteksi adanya kendaraan atau tidaknya.
- b. Sensor Ultrasonik 1, 2, dan 3 juga untuk memberitahukan jarak antara sensor dengan mobil.
- c. *Switch* 1, 2, dan 3 untuk memberitahukan jika adanya terjadi tabrakan dengan mobil.

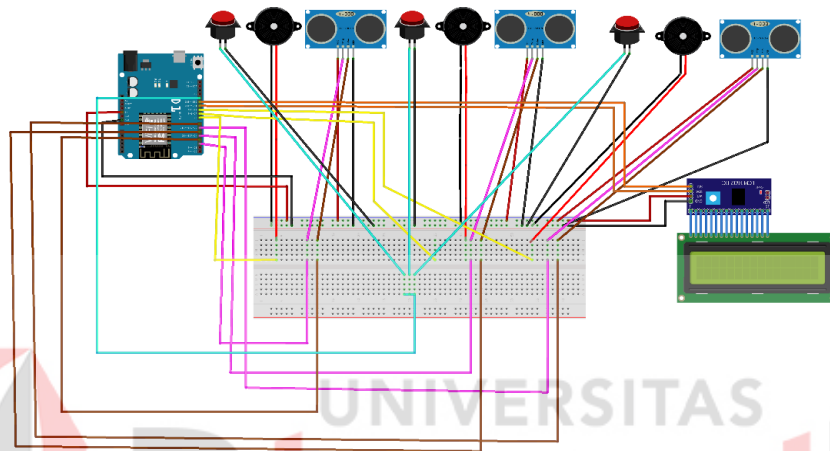
2. *Proses*

- a. Wemos D1 R32 adalah sebuah *board* mikrokontroler sebagai pengelola data dan pengiriman data *device*.

3. Output

- a. *Buzzer* 1, 2, dan 3 untuk memberikan sinyal kepada pengguna untuk mengetahui jarak antara mobil dengan dinding, semakin dekat kendaraan dengan dinding, bunyi akan semakin cepat.
- b. MQTT yang berfungsi sebagai pengiriman data ke admin melalui *smartphone*.
- c. LCD yang berfungsi untuk menampilkan angka jarak aman mobil.

3.2 Perancangan Rangkaian Skematik



Gambar 3.2 Rangkaian Skematik

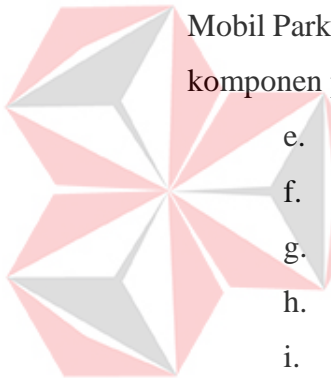
Tabel 3.1 Pin *input output* dari komponen rangkaian

Komponen	Pin	Pin Wemos D1 R32
Ultrasonik 1	Echo	26
	Trigger	25
	Vcc	+5v
	Gnd	Ground
Ultrasonik 2	Echo	17
	Trigger	16
	Vcc	+5v
	Gnd	Ground
Ultrasonik 3	Echo	27
	Trigger	14
	Vcc	+5v
	Gnd	Ground
Switch 1	Vcc	39
	Gnd	Ground
Switch 2	Vcc	36
	Gnd	Ground
Switch 3	Vcc	34
	Gnd	Ground
Buzzer 1	Vcc	18
	Gnd	Ground
Buzzer 2	Vcc	19
	Gnd	Ground

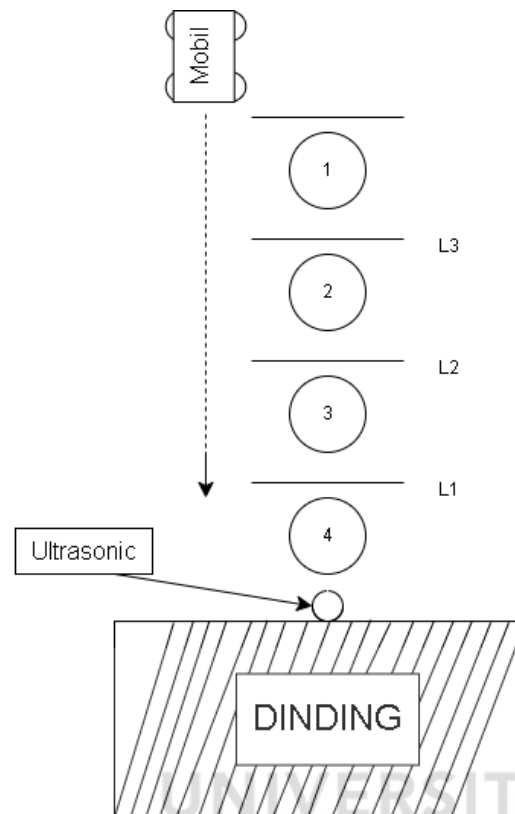
Komponen	Pin	Pin Wemos D1 R32
<i>Buzzer</i> 3	Vcc	23
	Gnd	Ground
LCD	SDA	SDA
	SCL	SCL
	Vcc	+5v
	Gnd	Ground

Pada Gambar 3.2 diatas merupakan rangkaian skematik pada pendeteksi jarak mobil parkir. Rangkaian terdiri dari Sensor Ultrasonik, Mikrokontroler Wemos D1 R32, *Buzzer*, LCD, dan *Push Button Switch*. Jalur penghubung antara Sensor Ultrasonik dengan *Buzzer* menggunakan Mikrokontroler Wemos D1 R32. Sensor Ultrasonik terhubung pin ECHO dengan pin 15, pin TRIGGER terhubung pada pin 14, pin GND terhubung pada pin GND, pin VCC pada 5V. Sedangkan *Buzzer* terhubung dengan pin 16. Dari Gambar Rangkaian Skematik diatas merupakan desain perancangan komponen untuk pembuatan alat yaitu “Alarm Pendeteksi Jarak Mobil Parkir Menggunakan Sensor Ultrasonik Berbasis IoT”. Terdiri dari beberapa komponen penting yaitu:

- e. 3 Sensor Ultrasonik
- f. Wemos D1 R32
- g. 3 *Buzzer*
- h. LCD 20x4
- i. 3 *Push Button Switch*



3.3 Setting Pengguna

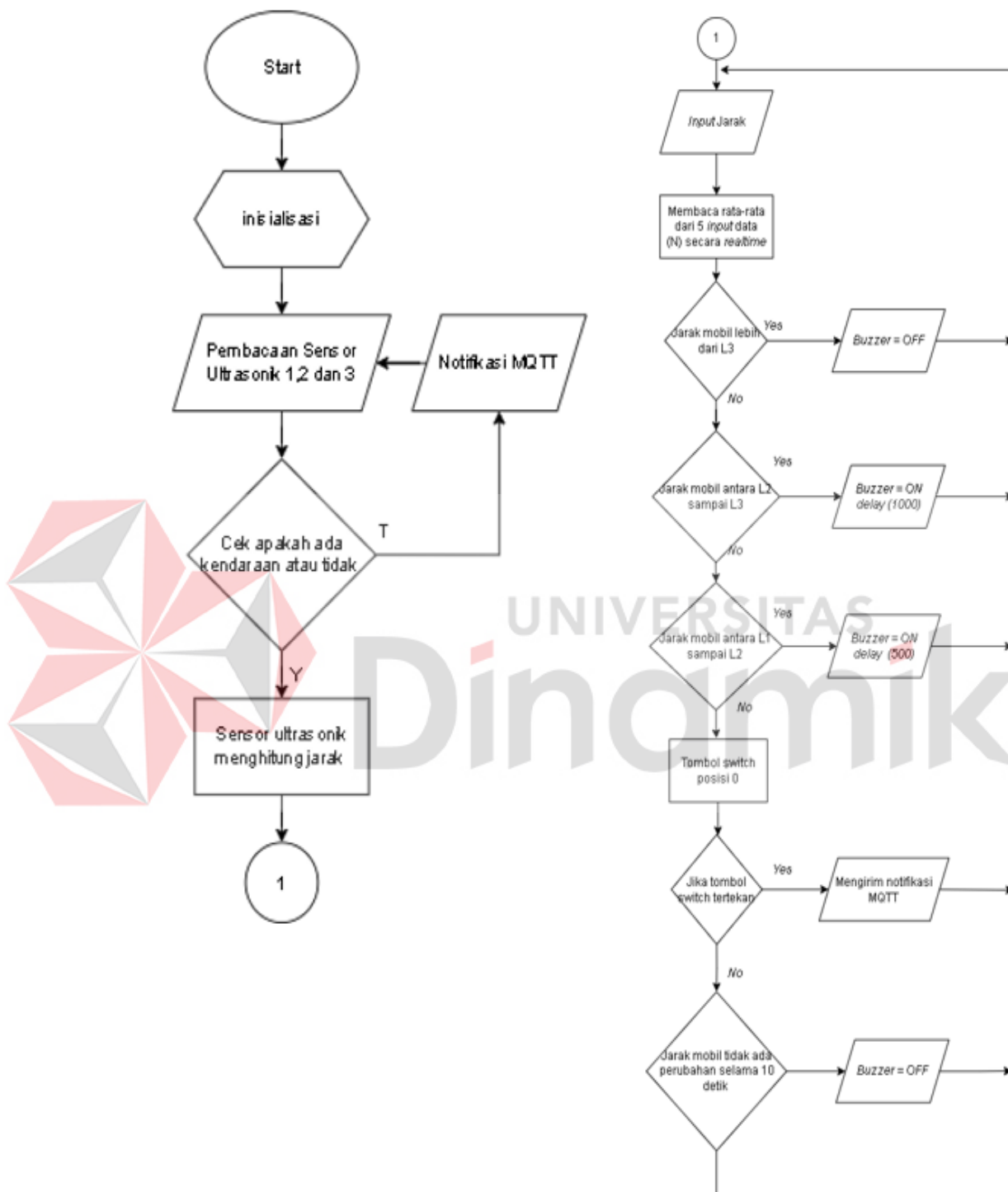


Gambar 3.3 Setting Pengguna

Pada alat alarm pendeteksi jarak mobil parkir ini dikhususkan untuk pengguna yang berlangganan di lahan parkir. Pada awal berlangganan pengguna diminta untuk mengisi data-data yang dibutuhkan seperti nama, mobil, plat, dan batas jarak. Batas jarak diisi sesuai keinginan pengguna agar dapat menyesuaikan jarak aman ketika parkir. Hal tersebut untuk menghindari terjadinya ketidakcocokan jarak aman tiap pengguna. Pada Gambar 3.3 diatas terdapat 3 *level setting* pengguna yang digunakan untuk keamanan mobil agar pengguna merasa nyaman. *Setting default* pada sistem ialah level 1, jarak mobil antara 5-15 cm. Level 2 yaitu jarak mobil antara 15-30 cm. Sedangkan level 3, jarak mobil lebih dari 30 cm. Pengguna dapat merubah jarak dari level 1-3 sesuai dengan yang diinginkan.

3.4 Flowchart

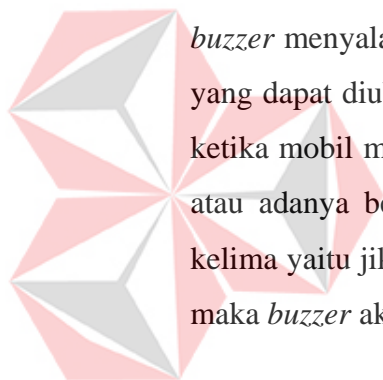
3.4.1 Kontrol Sistem



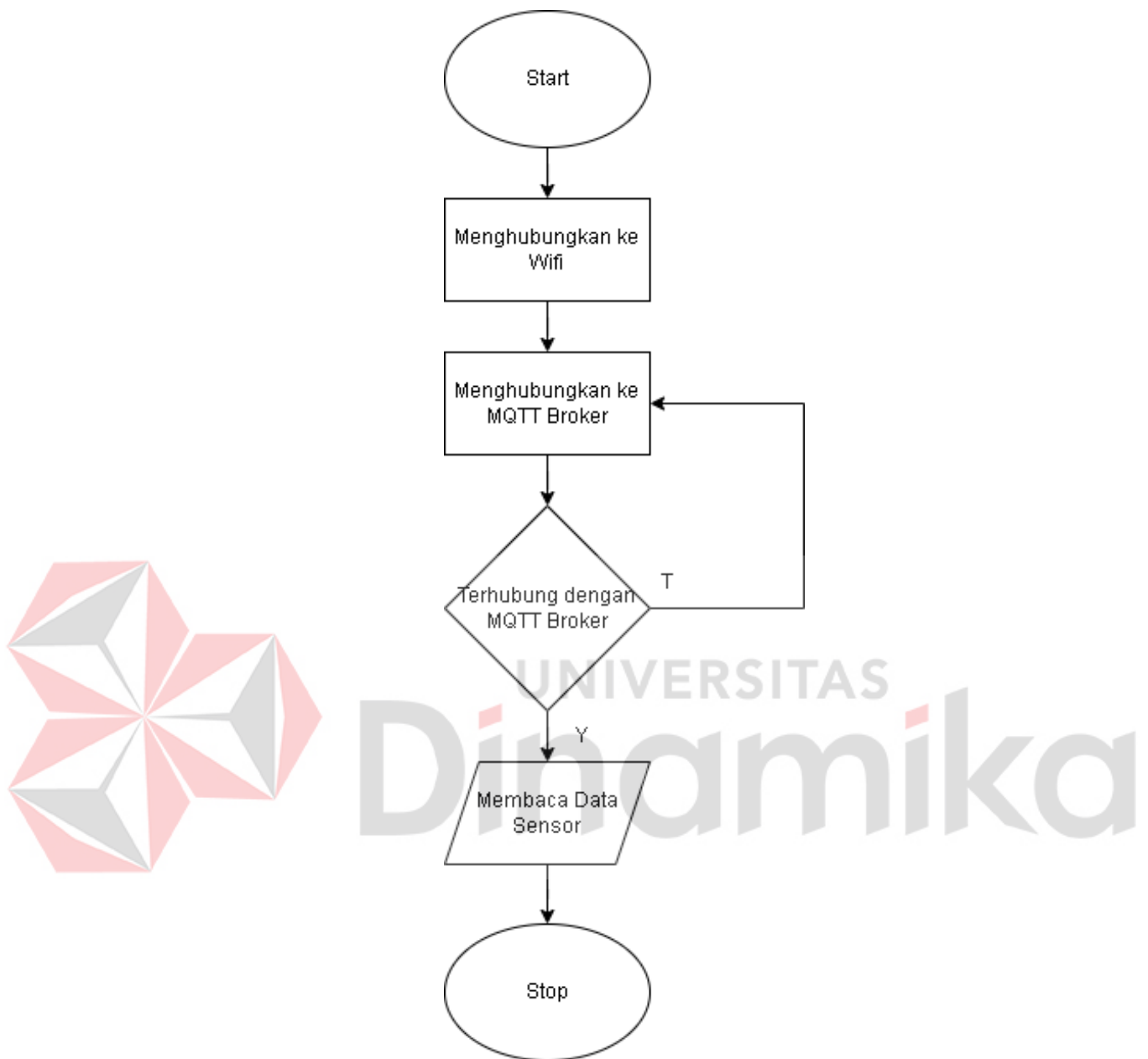
Gambar 3.4 Flowchart Pendeteksi Jarak Mobil Parkir

Flowchart dapat dilihat pada Gambar 3.4. Langkah pertama adalah inisialisasi, lalu dilanjutkan dengan pembacaan sensor ultrasonik 1-3. Selanjutnya pengecekan adanya kendaraan atau tidak. Jika tidak maka akan mengirimkan notifikasi MQTT, dan kembali pada proses pembacaan sensor ultrasonik. Jika terdapat kendaraan, maka langkah selanjutnya sensor ultrasonik akan menghitung jarak mobil parkir. proses selanjutnya akan melanjutkan pada bagian 1.

Pada bagian 1, proses yang pertama yaitu menginput jarak mobil parkir pada sensor ultrasonik yang akan dikirimkan ke mikrokontroler Wemos D1 R32. Lalu terjadi proses perhitungan rata-rata dari 5 input data (n) yang dilakukan secara *realtime*, proses ini dinamakan *Moving Average (MA)*. Selanjutnya terdapat 5 kondisi, kondisi pertama yaitu jarak lebih dari (L3), *buzzer* tidak menyala. Kondisi kedua, jarak objek pada mobil antara (L2) sampai (L3), *buzzer* menyala tetapi dengan *delay* 1 detik. Kondisi ketigayaitu jarak objek pada mobil (L1) sampai (L2), *buzzer* menyala dengan *delay* 0,5 detik. L1, L2, L3 merupakan angka jarak aman yang dapat diubah sesuai yang diinginkan oleh pengguna. Kondisi keempat yaitu ketika mobil menekan tombol *switch* maka akan menandakan terjadinya *accident* atau adanya benturan mobil dan akan mengirimkan notifikasi MQTT. Kondisi kelima yaitu jika jarak mobil dengan sensor tidak ada perubahan selama 10 detik, maka *buzzer* akan mati.



3.4.2 Komunikasi MQTT



Gambar 3.5 *Flowchart* Sistem MQTT

Pada Gambar 3.5 diatas merupakan *flowchart* dari sistem MQTT, yang dilakukan pertama kali yaitu menghubungkan ke *wifi*, selanjutnya menghubungkan ke MQTT *broker*. Jika tidak dapat terhubung proses akan kembali mengulang menghubungkan ke MQTT *broker*, jika berhasil terhubung dengan MQTT *broker*, setelah itu membaca data sensor.

3.5 Desain Model Prediksi Jarak Parkir Mobil

Model perancangan alat pada penelitian Tugas Akhir yang berjudul “Alarm Pendeteksi Jarak Mobil Parkir Menggunakan Sensor Ultrasonik Berbasis IoT” dapat dilihat pada Gambar 3.6. Menggunakan *input* Sensor Ultrasonik, Wemos D1 R32, dan *push button switch*. Sedangkan *output* menggunakan *Buzzer* untuk mengeluarkan bunyi *alarm*, dan juga LCD untuk menampilkan angka jarak aman pada mobil.



Gambar 3.6 Alat Pendeteksi Jarak Mobil Parkir

Pada Gambar 3.6 adalah alat pendeteksi jarak mobil parkir, alat tersebut dirancang seperti area parkir pada umumnya. Terdapat 3 buah mobil dan pada bagian belakang mobil masing-masing terdapat sensor ultrasonik, *buzzer*, dan *push button switch* yang menempel pada dinding. Pada bagian depan mobil terdapat LCD 20x4 yang menampilkan angka jarak aman, bagian bawah mobil akan berbentuk seperti loker yang berfungsi untuk menempatkan mikrokontroler Wemos D1 R32 beserta kabel-kabel yang terhubung, dan ditutupi dengan kayu agar terlihat lebih rapi dan tidak mengganggu pengendara mobil.

BAB IV HASIL DAN PEMBAHASAN

4.1 Pengujian Komunikasi MQTT

Pengujian komunikasi MQTT dilakukan untuk mengetahui kondisi jarak mobil parkir yaitu dengan cara mengirimkan sensor ultrasonik lalu dapat diterima oleh *device* dengan menggunakan protokol MQTT. Dalam pengujian kali ini, berikut adalah langkah – langkah yang harus dilakukan:

- a. Pastikan PC atau Laptop dalam keadaan terhubung ke *WiFi*.
- b. Membuka aplikasi Arduino IDE dan membuka *file* program yang sudah dibuat.
- c. Mengkonfigurasi *WiFi* dengan program untuk menghubungkan Wemos D1 R32 dengan *WiFi* dan MQTT.
- d. Menekan tombol *upload* untuk mengirim program ke Wemos D1 R32.
- e. Setelah program di *upload* maka nilai sensor dapat terbaca dan diteruskan ke *Broker* MQTT
- f. Komunikasi MQTT dika takan berhasil jika data dapat dikirimkan ke *Broker*, jika *Broker* tidak menerima data maka komunikasi tersebut maka dianggap gagal.

Tabel 4.1 Hasil Pengujian MQTT

No.	Komunikasi MQTT		Keterangan
	Terkirim	Tidak Terkirim	
1	-	✓	Gagal
2	-	✓	Gagal
3	✓	-	Berhasil
4	✓	-	Berhasil
5	✓	-	Berhasil
6	✓	-	Berhasil
7	✓	-	Berhasil
8	✓	-	Berhasil
9	✓	-	Berhasil
10	✓	-	Berhasil

No.	Komunikasi MQTT		Keterangan
	Terkirim	Tidak Terkirim	
11	✓	-	Berhasil
12	✓	-	Berhasil
13	✓	-	Berhasil
14	✓	-	Berhasil
15	✓	-	Berhasil
16	✓	-	Berhasil
17	✓	-	Berhasil
18	✓	-	Berhasil
19	✓	-	Berhasil
20	✓	-	Berhasil
21	✓	-	Berhasil
22	✓	-	Berhasil
23	✓	-	Berhasil
24	✓	-	Berhasil
25	✓	-	Berhasil
26	✓	-	Berhasil
27	✓	-	Berhasil
28	✓	-	Berhasil
29	✓	-	Berhasil
30	✓	-	Berhasil

Pada Tabel 4.1 dapat dilihat hasil pengujian komunikasi MQTT yang telah dilakukan. Pengujian komunikasi MQTT telah dilakukan sebanyak 30 kali, berhasil sebanyak 28 kali dan mengalami gagal (*error*) sebanyak 2 kali. Kegagalan tersebut dikarenakan sinyal yang kurang stabil dan sering terjadi *reconnecting* dari Wemos ke *WiFi*. Tingkat keberhasilan data pada alat ini sebesar 93,33% yang didapatkan dari rumus (1):

$$\begin{aligned}
 \text{Tingkat keberhasilan} &= \left| \frac{\text{Jumlah pesan yang terkirim}}{\text{Jumlah total pengujian}} \right| \times 100\% \\
 &= \left| \frac{28}{30} \right| \times 100\% \\
 &= 93.33\% \qquad (1)
 \end{aligned}$$

Perhitungan rumus untuk mendapatkan nilai tingkat keberhasilan didapatkan dari hasil perhitungan jumlah pesan yang terkirim dibagi dengan jumlah total

pengujian yang telah dilakukan. Kemudian agar mendapatkan nilai tingkat keberhasilan dalam bentuk persentase(%), hasil pembagian sebelumnya dikali dengan 100. Dari hasil perhitungan tersebut, didapatkan nilai tingkat keberhasilan komunikasi sebesar 93.33%. Pada saat melakukan percobaan terdapat beberapa masalah yang disebabkan oleh sinyal internet yang tidak stabil.



Gambar 4.1 Pengujian pada Aplikasi MQTT Tool

Pengujian yang dilakukan dikirim melalui aplikasi MQTT Tool sebanyak 30 kali. Terdapat beberapa *topic* diantaranya adalah 1a yaitu level 3 dengan angka 30 cm, 2a yaitu level 2 dengan angka 15 cm, 3a yaitu level 1 dengan angka 5 cm, 1b yaitu level 3 dengan angka 30 cm, 2b yaitu level 2 dengan angka 15 cm, 3b yaitu level 3 dengan angka 5 cm, 1c yaitu level 3 dengan angka 30 cm, 2c yaitu level 2 dengan angka 15 cm, 3c yaitu level 1 dengan angka 5 cm. Sebagai contoh pada Gambar 4.1 terdapat *topic* 3c yang diatur untuk tempat parkir ke 3 dengan level 1 pada angka 5 cm, ketika status telah berubah menjadi “*publish succeeded*” maka telah berhasil dikirim ke *Cloud*, lalu diterima oleh Arduino Uno.

4.2 Pengujian Pembacaan Sensor Ultrasonik

Pada pengujian sensor alat ini yang dilakukan hanya pengujian sensor ultrasonik. Pengujian pertama dilakukan untuk mengetahui apakah sensor dapat digunakan dengan seharusnya. Selanjutnya melakukan pengujian jarak sensor dengan penggaris (cm), dengan cara mencoba memasukkan program ke dalam sensor ultrasonik. Jika sensor ultrasonik telah berjalan dan berfungsi dengan baik maka dapat diketahui seberapa besar tingkat keakuratannya dalam mengambil data. Pengujian data sensor ultrasonik kali ini untuk menentukan nilai *error* sensor ultrasonik dengan jarak *real* atau menggunakan penggaris (cm). Uji coba dapat dilakukan dengan cara:

- a. Memastikan PC atau Laptop dalam keadaan terhubung ke *WiFi*.
- b. Membuka aplikasi Arduino IDE dan membuka program yang sudah dibuat.
- c. Wemos D1 R32 dapat dihubungkan melalui *WiFi* dan MQTT namun harus melakukan konfigurasi terlebih dahulu.
- d. Menekan Tombol *upload* digunakan untuk mengirim program ke Wemos D1 R32.
- e. Setelah program di *upload* dan terhubung lalu membuka serial *monitor*.
- f. Tunggu beberapa detik hingga pembacaan data sensor dapat membaca jarak dari sensor ke mobil. Data jarak sensor ultrasonik akan ditampilkan pada serial *monitor*.
- g. Catat hasil pembacaan dari sensor ultrasonik pada serial *monitor*.

Dari data yang didapatkan akan dihitung menggunakan penggaris agar dapat menghitung nilai *error* rata – rata dengan rumus (2) sebagai berikut:

$$Error(\%) = \left| \frac{\text{Nilai Penggaris} - \text{Data Sensor Ultrasonik}}{\text{Nilai Penggaris}} \right| * 100\% \quad (2)$$

Dengan menggunakan rumus diatas tersebut, untuk mendapatkan nilai *error* dilakukan dengan cara menghitung nilai penggaris dikurangi data sensor ultrasonik, kemudian hasilnya dibagi dengan nilai penggaris. Selanjutnya dari hasil terakhir perlu dikalikan dengan 100 untuk mendapatkan nilai dalam bentuk persentase(%). Nilai *error* setiap pengujian yang telah dilakukan ditampilkan pada tabel pengujian dalam bentuk persentase(%).

Persentase rata-rata nilai *error* ditampilkan pada setiap tabel pengujian yang telah dilakukan. Rumus rata-rata nilai *error* (3) adalah sebagai berikut:

$$\text{Rata – Rata Nilai Error} = \frac{\text{Jumlah semua data error}}{\text{Jumlah total pengujian}} \quad (3)$$

Menjumlahkan semua nilai *error* yang ada kemudian dibagi dengan banyaknya percobaan, maka akan didapatkan persentase rata-rata nilai *error* sesuai dengan jarak dan masing-masing sensor ultrasonik. Pengujian pada ketiga sensor ultrasonik akan mengirimkan data dan menampilkan hasilnya sebagai berikut:

Tabel 4.2 Tabel Hasil Pengujian Sensor Ultrasonik 1

No.	Penggaris (cm)	Pembacaan sensor ultrasonik 1 (cm)	Selisih	Error (%)	Buzzer	Accident
1.	35	35,36	0,36	1,02	mati	-
2.		35,23	0,23	0,66	mati	-
3.		35,04	0,04	0,11	mati	-
4.		34,81	0,19	0,54	mati	-
5.		34,71	0,29	0,83	mati	-
6.		34,42	0,58	1,66	mati	-
Rata-rata <i>error</i> pada jarak 35 cm				0,8		
7.	25	23,51	1,49	5,96	bunyi setiap 1 detik	-
8.		23,51	1,49	5,96	bunyi setiap 1 detik	-
9.		23,51	1,49	5,96	bunyi setiap 1 detik	-
10.		23,51	1,49	5,96	bunyi setiap 1 detik	-
11.		23,51	1,49	5,96	bunyi setiap 1 detik	-
12.		23,51	1,49	5,96	bunyi setiap 1 detik	-
Rata-rata <i>error</i> pada jarak 25 cm				5,96		
13.	15	14,44	0,56	3,73	bunyi setiap 0,5 detik	-
14.		14,44	0,56	3,73	bunyi setiap 0,5 detik	-
15.		14,44	0,56	3,73	bunyi setiap 0,5 detik	-
16.		14,44	0,56	3,73	bunyi setiap 0,5 detik	-
17.		14,44	0,56	3,73	bunyi setiap 0,5 detik	-
18.		14,44	0,56	3,73	bunyi setiap 0,5 detik	-
Rata-rata <i>error</i> pada jarak 15 cm				3,73		

No.	Penggaris (cm)	Pembacaan sensor ultrasonik 1 (cm)	Selisih	Error (%)	Buzzer	Accident
19.	7	6,97	0,03	0,43	bunyi setiap 0,5 detik	-
20.		7	0	0	bunyi setiap 0,5 detik	-
21.		7,02	0,02	0,28	bunyi setiap 0,5 detik	-
22.		7,04	0,04	0,57	bunyi setiap 0,5 detik	-
23.		7,05	0,05	0,71	bunyi setiap 0,5 detik	-
24.		7,07	0,07	1	bunyi setiap 0,5 detik	-
Rata-rata error pada jarak 7 cm				0,5		
25.	3	3,23	0,23	7,67	bunyi setiap saat	terjadi
26.		3,23	0,23	7,67	bunyi setiap saat	terjadi
27.		3,23	0,23	7,67	bunyi setiap saat	terjadi
28.		3,23	0,23	7,67	bunyi setiap saat	terjadi
29.		3,23	0,23	7,67	bunyi setiap saat	terjadi
30.		3,23	0,23	7,67	bunyi setiap saat	terjadi
Rata-rata error pada jarak 3 cm				7,77		

Pada Tabel 4.2 diatas merupakan data-data yang didapatkan setelah melakukan pengujian pada sensor ultrasonik 1. Dilakukan pengujian sebanyak 30 kali. Terdapat 5 bagian batas jarak dengan mobil yang telah ditandai menggunakan penggaris, diantaranya angka 35, 25, 15, 7, dan 3. Pada jarak 35 cm untuk sensor ultrasonik 1 diperoleh rata-rata error sebesar 0,8%, untuk jarak 25 cm sebesar 5,96%, untuk jarak 15 cm sebesar 3,73%, untuk jarak 7 cm sebesar 0,5%, untuk jarak 3 cm sebesar 7,77%. Berdasarkan rata-rata error pada jarak yang berbeda pada ultrasonik 1 dibawah 10%. Hal ini dapat disimpulkan bahwa *moving average* dapat meningkatkan akurasi deteksi jarak oleh ultrasonik. Penerapan *moving average* dapat meningkatkan akurasi.

Tabel 4.3 Tabel Hasil Pengujian Sensor Ultrasonik 2

No.	Penggaris (cm)	Pembacaan sensor ultrasonik 2 (cm)	Selisih	Error (%)	Buzzer	Accident
1.	35	35,2	0,2	1,23	mati	-
2.		35,12	0,12	1,29	mati	-
3.		35,05	0,05	1,29	mati	-
4.		34,99	0,01	1,29	mati	-
5.		34,86	0,14	1,29	mati	-
6.		34,74	0,26	1,29	mati	-
Rata-rata <i>error</i> pada jarak 35 cm				1,28		
7.	25	25	0	0	bunyi setiap 1 detik	-
8.		25	0	0	bunyi setiap 1 detik	-
9.		25	0	0	bunyi setiap 1 detik	-
10.		25	0	0	bunyi setiap 1 detik	-
11.		25	0	0	bunyi setiap 1 detik	-
12.		25	0	0	bunyi setiap 1 detik	-
Rata-rata <i>error</i> pada jarak 25 cm				0		
13.	15	14,86	0,14	0,93	bunyi setiap 0,5 detik	-
14.		14,86	0,14	0,93	bunyi setiap 0,5 detik	-
15.		14,86	0,14	0,93	bunyi setiap 0,5 detik	-
16.		14,86	0,14	0,93	bunyi setiap 0,5 detik	-
17.		14,86	0,14	0,93	bunyi setiap 0,5 detik	-
18.		14,86	0,14	0,93	bunyi setiap 0,5 detik	-
Rata-rata <i>error</i> pada jarak 15 cm				0,93		
19.	7	7,04	0,04	0,57	bunyi setiap 0,5 detik	-
20.		7,04	0,04	0,57	bunyi setiap 0,5 detik	-
21.		7,03	0,03	0,43	bunyi setiap 0,5 detik	-
22.		7,03	0,03	0,43	bunyi setiap 0,5 detik	-
23.		7,04	0,04	0,57	bunyi setiap 0,5 detik	-
24.		7,04	0,04	0,57	bunyi setiap 0,5 detik	-

No.	Penggaris (cm)	Pembacaan sensor ultrasonik 2 (cm)	Selisih	Error (%)	Buzzer	Accident
Rata-rata error pada jarak 7 cm				0,52		
25.	3	3,18	0,18	6	bunyi setiap saat	terjadi
26.		3,18	0,18	6	bunyi setiap saat	terjadi
27.		3,18	0,18	6	bunyi setiap saat	terjadi
28.		3,18	0,18	6	bunyi setiap saat	terjadi
29.		3,18	0,18	6	bunyi setiap saat	terjadi
30.		3,18	0,18	6	bunyi setiap saat	terjadi
Rata-rata error pada jarak 3 cm				6		

Pada Tabel 4.3 diatas merupakan data-data yang didapatkan setelah melakukan pengujian pada sensor ultrasonik 2. Dilakukan pengujian sebanyak 30 kali. Terdapat 5 bagian batas jarak dengan mobil yang telah ditandai menggunakan penggaris, diantaranya angka 35, 25, 15,7, dan 3. Pada jarak 35 cm untuk sensor ultrasonik 2 diperoleh rata-rata error sebesar 1,28%, untuk jarak 25 cm sebesar 0%, untuk jarak 15 cm sebesar 0,93%, untuk jarak 7 cm sebesar 0,52%, untuk jarak 3 cm sebesar 6%. Berdasarkan rata-rata error pada jarak yang berbeda pada ultrasonik 2 dibawah 10%. Hal ini dapat disimpulkan bahwa *moving average* dapat meningkatkan akurasi deteksi jarak oleh ultrasonik. Penerapan *moving average* dapat meningkatkan akurasi.

Tabel 4.4 Tabel Hasil Pengujian Sensor Ultrasonik 3

No.	Penggaris (cm)	Pembacaan sensor ultrasonik 3 (cm)	Selisih	Error (%)	Buzzer	Accident
1.	35	35	0	0	mati	-
2.		35,1	0,1	0,29	mati	-
3.		35,16	0,16	0,46	mati	-
4.		35,22	0,22	0,63	mati	-
5.		35,26	0,26	0,74	mati	-
6.		35,46	0,46	1,31	mati	-

No.	Penggaris (cm)	Pembacaan sensor ultrasonik 3 (cm)	Selisih	Error (%)	Buzzer	Accident
Rata-rata error pada jarak 35 cm				0,57		
7.	25	25,28	0,28	1,12	bunyi setiap 1 detik	-
8.		25,23	0,23	0,92	bunyi setiap 1 detik	-
9.		25,11	0,11	0,44	bunyi setiap 1 detik	-
10.		25,09	0,09	0,36	bunyi setiap 1 detik	-
11.		24,99	0,01	0,04	bunyi setiap 1 detik	-
12.		24,98	0,02	0,08	bunyi setiap 1 detik	-
Rata-rata error pada jarak 25 cm				0,49		
13.	15	15,4	0,4	2,67	bunyi setiap 0,5 detik	-
14.		15,3	0,3	2	bunyi setiap 0,5 detik	-
15.		15,21	0,21	1,4	bunyi setiap 0,5 detik	-
16.		15,12	0,12	0,8	bunyi setiap 0,5 detik	-
17.		15,14	0,14	0,93	bunyi setiap 0,5 detik	-
18.		14,97	0,03	0,2	bunyi setiap 0,5 detik	-
Rata-rata error pada jarak 15 cm				1,3		
19.	7	7,29	0,29	4,14	bunyi setiap 0,5 detik	-
20.		7,29	0,29	4,14	bunyi setiap 0,5 detik	-
21.		7,29	0,29	4,14	bunyi setiap 0,5 detik	-
22.		7,29	0,29	4,14	bunyi setiap 0,5 detik	-
23.		7,29	0,29	4,14	bunyi setiap 0,5 detik	-
24.		7,3	0,3	4,29	bunyi setiap 0,5 detik	-
Rata-rata error pada jarak 7 cm				4,16		
25.	3	2,97	0,03	1	bunyi setiap saat	terjadi
26.		3,06	0,06	2	bunyi setiap saat	terjadi
27.		3,12	0,12	4	bunyi setiap saat	terjadi
28.		3,17	0,17	5,67	bunyi setiap saat	terjadi
29.		3,13	0,13	4,33	bunyi setiap saat	terjadi
30.		3,1	0,1	3,33	bunyi setiap saat	terjadi
Rata-rata error pada jarak 3 cm				3,39		

Pada Tabel 4.4 diatas merupakan data-data yang didapatkan setelah melakukan pengujian pada sensor ultrasonik 3. Dilakukan pengujian sebanyak 30 kali. Terdapat 5 bagian batas jarak dengan mobil yang telah ditandai menggunakan penggaris, diantaranya angka 35, 25, 15, 7, dan 3. Pada jarak 35 cm untuk sensor ultrasonik 3 diperoleh rata-rata *error* sebesar 0,57%, untuk jarak 25 cm sebesar 0,49%, untuk jarak 15 cm sebesar 1,3%, untuk jarak 7 cm sebesar 4,16%, untuk jarak 3 cm sebesar 3,39%. Berdasarkan rata-rata *error* pada jarak yang berbeda pada ultrasonik 3 dibawah 10%. Hal ini dapat disimpulkan bahwa *moving average* dapat meningkatkan akurasi deteksi jarak oleh ultrasonik. Penerapan *moving average* dapat meningkatkan akurasi.

4.3 Tabel Kebenaran jarak

Pengujian kebenaran jarak ini dilakukan untuk mengetahui apakah alat yang dirancang sudah berjalan sesuai dengan yang diinginkan, yaitu program dapat bekerja secara paralel. Pengujian kebenaran jarak sensor dilakukan dari lahan parkir pertama, kedua, dan ketiga, dengan cara memasukan program ke dalam ultrasonik. Jika ketiga sensor ultrasonik telah berjalan dan berfungsi dengan baik, maka dapat diketahui bagaimana kinerja program paralel dengan menggunakan 1 mikrokontroler. Pengujian kebenaran jarak juga dilakukan untuk menentukan apakah *buzzer* dapat berbunyi sesuai *setting default* jarak pada program, secara bersamaan dengan dua lahan parkir yang berbeda. Uji coba dapat dilakukan dengan cara:

- a. Memastikan PC atau Laptop dalam keadaan terhubung ke *WiFi*.
- b. Membuka aplikasi Arduino IDE dan membuka program yang sudah dibuat.
- c. Wemos D1 R32 dapat dihubungkan melalui *WiFi* dan MQTT namun harus melakukan konfigurasi terlebih dahulu.
- d. Menekan tombol *upload* digunakan untuk mengirim program ke Wemos D1 R32.
- e. Setelah program di *upload* dan terhubung lalu membuka serial *monitor*.
- f. Menunggu beberapa detik hingga pembacaan data sensor dapat membaca jarak dari sensor ke mobil. Data jarak sensor ultrasonik akan ditampilkan pada serial *monitor*.

- g. Mencatat hasil pembacaan dari sensor ultrasonik pada serial *monitor* dan catat lahan parkir mobil kesatu, kedua, dan ketiga bunyi *buzzer* yang berbunyi ‘*beep*’.

Tabel 4.5 Tabel *setting default* jarak mobil dengan objek

No	Level jarak tiap lahan parkir	Setting jarak default (cm)	Aksi <i>buzzer</i>
1	Level 1	5 – 15	Bunyi setiap 0,5 detik
2	Level 2	15 – 30	Bunyi setiap 1 detik
3	Level 3	30 – 60	Mati

Pada Tabel 4.5 terdapat beberapa *level setting* jarak *default* atau angka jarak paten pada program, sebelum adanya perubahan *setting* dari admin melalui MQTT. Ketika posisi mobil akan memasuki lahan parkir pada saat itu mobil sudah berada pada posisi *level 3*, yaitu mobil berada di jarak lebih dari 30 cm. Pada saat itu *buzzer* tidak mengeluarkan suara atau mati, mobil akan berada di *level 2* ketika memasuki jarak antara 15 cm sampai 30 cm, *buzzer* akan berbunyi dengan *delay* setiap 1 detik yang artinya mobil berada pada jarak sedang atau aman. Mobil akan memasuki *level 1* atau dianggap *level* bahaya, *buzzer* akan berbunyi dengan *delay* 0,5 detik yang menandakan mobil harus berhenti.

Tabel 4.6 Tabel Hasil Pengujian Paralel

No.	Lahan Parkir	Posisi Mobil (cm)	Hasil Aksi <i>Buzzer</i>	Sesuai
1.	1	23,51	Bunyi setiap 1 detik	Sesuai
	2	7,04	Bunyi setiap 0,5 detik	Sesuai
2.	1	14,44	Bunyi setiap 0,5 detik	Sesuai
	3	35	Mati	Sesuai
3.	2	14,86	Bunyi setiap 0,5 detik	Sesuai
	1	34,42	Mati	Sesuai
4.	2	34,74	Mati	Sesuai
	3	24,98	Bunyi setiap 1 detik	Sesuai
5.	3	15,21	Bunyi setiap 1 detik	Sesuai
	1	7,02	Bunyi setiap 0,5 detik	Sesuai
6.	3	7,3	Bunyi setiap 0,5 detik	Sesuai
	2	14,86	Bunyi setiap 0,5 detik	Sesuai
7.	2	3,18	Bunyi setiap saat	Sesuai
	3	25,11	Bunyi setiap 1 detik	Sesuai

No.	Lahan Parkir	Posisi Mobil (cm)	Hasil Aksi <i>Buzzer</i>	Sesuai
8.	3	24,98	Bunyi setiap 1 detik	Sesuai
	1	3,23	Bunyi setiap saat	Sesuai
9.	2	7,04	Bunyi setiap 0,5 detik	Sesuai
	1	14,44	Bunyi setiap 0,5 detik	Sesuai
10.	3	2,97	Bunyi setiap saat	Sesuai
	2	35,12	Mati	Sesuai

Pada Tabel 4.5 merupakan pengujian sensor ultrasonik yang dilakukan secara paralel atau bersamaan. Pengujian dilakukan sebanyak 10 kali, dengan menguji lahan 1, 2, dan 3 jika dijalankan secara bersama. Lalu diambil 2 data lahan untuk mengetahui apakah sistem dapat berjalan jika lahan parkir digunakan oleh pengendara-pengendara yang memarkirkan mobilnya secara bersamaan. Hasil dari pengujian tersebut adalah sesuai dengan yang diinginkan, yaitu bekerja secara paralel, sehingga lahan parkir dapat digunakan secara bersamaan oleh pengendara dengan pengendara yang lainnya.



BAB V

PENUTUP

Pada bab 5 ini, penulis memberikan kesimpulan dan saran dari hasil pengujian pada Tugas Akhir ini, agar penelitian selanjutnya dapat dikembangkan berdasarkan hasil, kesimpulan dan saran pada penelitian ini.

5.1 Kesimpulan

Hasil kesimpulan dari Tugas Akhir ini didapatkan berdasarkan dari beberapa tahapan, diantaranya: dimulai dari perancangan, hasil dan pembahasan. Adapun kesimpulannya adalah sebagai berikut:

1. Pengujian pendeteksi jarak pada ketiga sensor ultrasonik dilakukan sebanyak 30 kali pada jarak 35 cm, 25 cm, 15 cm, 7 cm, dan 3 cm. Pada sensor ultrasonik 1 didapatkan rata-rata *error* untuk jarak 35 cm sebesar 0,8% , untuk 25 cm sebesar 5,96%, untuk 15 cm sebesar 3,37%, untuk 7 cm sebesar 0,5%, untuk 3 cm sebesar 7,77%. Pada sensor ultrasonik 2 didapatkan rata-rata *error* untuk jarak 35 cm sebesar 1,28%, untuk 25 cm sebesar 0%, untuk 15 cm sebesar 0,93%, untuk 7 cm sebesar 0,52%, untuk 3 cm sebesar 6%. Pada sensor ultrasonik 3 didapatkan rata-rata *error* untuk jarak 35 cm sebesar 0,57%, untuk 25 cm sebesar 0,49%, untuk 15 cm 1,3%, untuk 7 cm sebesar 4,16%, untuk 3 cm sebesar 3,39%.
2. Pengendara dapat mengetahui batas jarak aman mobil dari objek/dinding melalui *buzzer* berupa suara ‘*beep*’, semakin dekat mobil dengan objek maka suara yang dikeluarkan akan semakin cepat. Disamping itu pada penelitian ini dapat memberikan informasi jika mobil sudah berada di batas jarak yang masih aman, dan pengendara dapat melihat angka jarak pada layar LCD.
3. Hasil pengujian pengiriman komunikasi MQTT menggunakan *software* IoT yaitu MQTT *Tool*, didapatkan nilai akurasi sebesar 93.33%. Pengendara dapat menyesuaikan jarak yang diinginkan melalui admin untuk mengatur jarak menggunakan *software* tersebut.

5.2 Saran

Saran untuk mengembangkan dan menyempurnakan dari alat Tugas Akhir ini adalah sebagai berikut:

1. Alat dipasang pada area yang memiliki koneksi jaringan *wireless* yang stabil.
2. Dapat mengimplimentasikan alat pada area parkir yang *real*.
3. Menambahkan *counter* untuk menghitung beberapa banyak jumlah mobil yang akan melakukan parkir.



UNIVERSITAS
Dinamika

DAFTAR PUSTAKA

- Alawiah, A., & Al Tahtawi, A. R. (2017). Sistem Kendali dan Pemantauan Ketinggian Air pada Tangki Berbasis Sensor Ultrasonik. *Jurnal Ilmiah Manajemen Informatika dan Komputer*, 25-30.
- Khotimah, N. (2021). Rancang Bangun Hardware Smart Dispenser Otomatis Berbasis IoT Pada PT. Plambo Pratama JS. *TUGAS AKHIR*.
- Maricar, M. (2019). Analisa Perbandingan Nilai Akurasi Moving Average dan Exponential Smoothing untuk Sistem Peramalan Pendapatan pada Perusahaan XYZ. *Jurnal Sistem dan Informatika*, 36-123.
- Naoval, D. A. (2022). Deteksi Jarak Aman Mobil Listrik Menggunakan Ultrasonic Sensor. *Kerja Praktik*.
- Purwanto, H., Riyadi, M., Astuti, D. W., & Kusuma, I. A. (2019). Komparasi Sensor Ultrasonik HC-SR04 dan JSN-SR04T Untuk Aplikasi Sistem Deteksi Ketinggian Air. *Jurnal SIMETRIS*, 1-8.
- Putra, A. W., Bhawiyuga, A., & Data, M. (2018). Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 584-593.
- Setiadi, D. H. (2021). Rancang Bangun Alat Desalinasi Air Laut Sebagai Sumber Air Minum Berbasis Wemos D1. *Tugas Akhir*.
- Shaputra, R., Gunoto, P., & Irsyam, M. (2019). Kran Air Otomatis Pada Tempat Berwudhu Menggunakan Sensor Ultrasonik Berbasis Arduino Uno. *Sigma Teknika*, 192-201.
- Sutono, & Nursoparisa, A. (2019). Perancangan Sistem Kendali Automatisasi Control Debit Air pada Pengisian Galon Menggunakan Modul Arduino. *Media Jurnal Informatika*, 33-42.
- Thayeb, M. G. (2018). Prototipe Rekayasa Lampu Traffic Berdasarkan Kepadatan Lalu Lintas. *Tugas Akhir*.
- Trisna, D. R., & Maharani, T. (2022). Penerapan Digital Moving Average Filter Pada Sensor Dissolved Oksigen Untuk Mengukur Kualitas Air. *Jurnal EEMISAS*, 32-40.

Widodo, T., Irawan, B., & Prastowo, A. T. (2020). Sistem Sirkulasi Air Pada Teknik Budidaya Bioflok Menggunakan Mikrokontroler Arduino Uno R3. *JTIKOM*, 34-39.

Zuhri, M., & Okselia, H. (2021). Rancang Bangun Sistem Monitoring Daya Listrik Berbasis IoT. *Tugas Akhir*.



UNIVERSITAS
Dinamika