

*MONITORING PERPARKIRAN MOBIL MENGGUNAKAN
SENSOR ULTRASONIK PING BERBASIS WIRELESS*



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan
Pendidikan diploma empat (D-4) Program Studi Teknik Komputer dan Jaringan
Jurusan Teknik Elektro
Politeknik Negeri Ujung Pandang

YORI PUTRA PRADANA
425 15 014

PROGRAM STUDI D-4 TEKNIK KOMPUTER DAN JARINGAN
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI UJUNG PANDANG
MAKASSAR
2019

HALAMAN PENGESAHAN

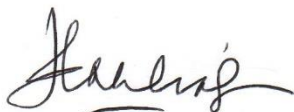
Skripsi dengan judul “*Monitoring Perparkiran Mobil Menggunakan Sensor Ultrasonik Ping Berbasis Wireless*” Oleh Yori Putra Pradana (425 15 014) telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Makassar, 27 Juni 2019

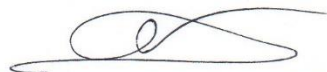
Mengesahkan,

Pembimbing I,

Pembimbing II,



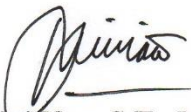
Ir. Dahlia, M.T.
NIP. 196412311991032003



Sahbuddin Abdul Kadir, S.T., M.T.
NIP. 197511302006041001

Mengetahui,

Ketua Program Studi,
Teknik Komputer dan Jaringan
Politeknik Negeri Ujung Pandang



Rini Nur, S.T., M.T.
NIP. 197307132009122001

HALAMAN PENERIMAAN

Pada hari ini, Kamis tanggal 27 Juni 2019, Tim Penguji Sidang Tugas Akhir, telah menerima dengan baik hasil skripsi oleh mahasiswa : **Yori Putra Pradana (425 15 014)** dengan judul ***"Monitoring Perparkiran Mobil Menggunakan Sensor Ultrasonik Ping Berbasis Wireless"***

Makassar, 27 Juni 2019

Tim Penguji Ujian Skripsi:


1 Muh. Fajri Raharjo, S.T., M.T.

Ketua

()

2 Meylanie Olivya, S.T., M.T.

Sekretaris

()

3 Drs. Kasim, M.T.

Anggota

()

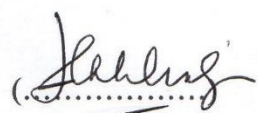
4 Irfan Syamsuddin, S.T., M.Com. ISM.Ph.D

Anggota

()

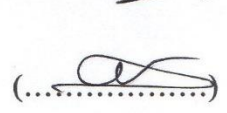
5 Ir.Dahlia, M.T.

Pembimbing I

()

6 Sahbuddin Abdul Kadir, S.T., M.T.

Pembimbing II

()

KATA PENGANTAR

Puji syukur senantiasa penulis panjatkan kehadirat Allah SWT atas Berkat dan Rahmat-Nya yang telah memberikan kesehatan dan keselamatan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat dan salam kepada baginda Rasul Muhammad SAW sebagai sebaik-baik panutan bagi seluruh manusia.

Skripsi ini disusun guna memenuhi salah satu syarat untuk menyelesaikan studi serta dalam rangka memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Penulis menyadari bahwa keberhasilan penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, dengan rendah hati penulis mengucapkan terima kasih kepada:

1. Orangtua penulis yakni Ayahanda Mursalim Baharuddin dan Ibunda Syamsiah Djamaluddin serta Kakak Kandung Penulis, Andini Firdha, Zohra Dwi Wahyuni dan Sry Defi yang sampai saat ini selalu memberikan semangat, motivasi, dukungan, bimbingan dan doa kepada penulis.
2. Bapak Prof. Ir. Muhammad Anshar, M. Si., Ph.D. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Ibu Dr. Ir. Hafsah Nirwana, M.T. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Ibu Rini Nur, S.T., M.T. selaku Ketua Program Studi Teknik Komputer dan

Jaringan.

5. Ibu Ir. Dahlia, M.T. selaku pembimbing I dan Bapak Sahbuddin Abdul Kadir, ST., M.T. selaku pembimbing II atas segala ilmu, motivasi, nasehat, arahan, bantuan dan kesedian waktu dan kesabarannya dalam membimbing penulis hingga terselesaikannya penelitian ini.
6. Seluruh dosen dan Staf Jurusan Teknik Elektro, Khususnya Prodi Teknik Komputer dan Jaringan.
7. Teman-teman seperjuangan di Program Studi TKJ angkatan 2015 yang selalu memberikan semangat selama menyelesaikan studi di Politeknik Negeri Ujung Pandang.
8. Semua pihak yang telah memberikan bantuan moril maupun materil yang tidak dapat disebutkan satu per satu

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, sehingga penulis mengharapkan kritik dan saran yang sifatnya membangun untuk perbaikan di masa mendatang. Semoga tulisan ini bermanfaat.

Makassar, 27 Juni 2019

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN	iii
KATA PENGANTAR	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL.....	x
DAFTAR LAMPIRAN	xi
SURAT PERNYATAAN.....	xii
RINGKASAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 <i>Monitoring</i>	5
2.2 Parkir	5
2.3 Mikrokontroler.....	6
2.3.1 NodeMCU ESP8266	6
2.4 Sensor Ultrasonik.....	8
2.4.1 Sensor Ultrasonik HC-SR04	10
2.4.2 Sensor Ultrasonik PING	11
2.5 Arduino IDE	13
2.6 <i>Database</i>	15
2.6.1 Pengertian <i>Database</i>	15
2.6.2 <i>Database Management System (DBMS)</i>	15

2.6.3	MySQL.....	16
2.8	Protokol MQTT.....	17
BAB III METODOLOGI PENELITIAN.....		24
3.1	Tempat Dan Waktu	24
3.2	Perangkat Penelitian	24
3.2.1	Perangkat Keras	25
3.2.2	Perangkat Lunak.....	25
3.3	Metode Penelitian.....	24
3.3.1	Studi Literatur	25
3.3.2	Analisis Sistem	25
3.3.3	Tahap Perancangan	27
3.3.4	Implementasi Sistem	32
3.3.5	Pengujian Sistem	33
3.3.6	Hasil Penelitian	33
3.3.7	Evaluasi	33
BAB IV HASIL DAN PEMBAHASAN		34
4.1	Implementasi	36
4.1.1	Halaman Utama.....	36
4.1.2	Halaman Pemasukan Nomor Plat Kendaraan	37
4.1.3	Tampilan Cetak Kartu Parkir	38
4.1.4	Prototipe Lot Parkir	38
4.2	Pengujian	39
4.2.1	Pengujian Fungsionalitas	39
4.2.2	Pengujian Kinerja	47
BAB V PENUTUP.....		53
5.1	Kesimpulan.....	53
5.2	Saran	53
DAFTAR PUSTAKA		54
LAMPIRAN		54

DAFTAR GAMBAR

Halaman

Gambar 2. 1 NodeMCU ESP8266	7
Gambar 2. 2 <i>Pinout</i> Diagram	8
Gambar 2. 3 Sensor Ultrasonik HC-SR04	11
Gambar 2. 3 Sensor Ultrasonik PING Parallax	12
Gambar 2. 4 Prinsip Kerja Sensor Ultrasonik PING	13
Gambar 2. 5 Tampilan Perangkat Lunak Arduino IDE	14
Gambar 2. 6 Alur pesan <i>CONNECT</i> dengan <i>clean session</i>	18
Gambar 2. 7 Alur pesan <i>CONNECT</i> tanpa <i>clean session</i>	19
Gambar 2. 8 Alur pesan <i>SUBSCRIBE</i> and <i>UNSUBSCRIBE</i> untuk sebuah topik ..	20
Gambar 2. 9 Alur pesan <i>PUBLISH</i> dengan QoS 0 (At most once).....	21
Gambar 2. 10 Alur pesan <i>PUBLISH</i> dengan QoS 1 (At least once)	21
Gambar 2. 12 Alur pesan ping untuk <i>keepalive</i>	22
Gambar 2. 13 Alur pesan <i>DISCONNECT</i> untuk memutuskan koneksi	23
Gambar 3. 1 Metode Penelitian.....	26
Gambar 3. 2 Arsitektur sistem	28
Gambar 3. 3 <i>Activity</i> Diagram.....	29
Gambar 3. 4 (a) <i>Flowchart</i> Sistem, (b) <i>Flowchart</i> Cek Lot Parkir.....	30
Gambar 3. 5 Rangkaian Mikrokontroller Dan Sensor Ultrasonik.....	31
Gambar 3. 6 <i>Flowchart</i> Rangkaian Mikrokontroller Dan Sensor Ultrasonik	32
Gambar 4.1 Ilustrasi Pub/Sub dalam MQTT	35
Gambar 4.2 Tampilan Halaman Utama	36
Gambar 4.3 Warna lot parkir	37
Gambar 4.4 Tampilan Pemasukan Nomor Plat Kendaraan	37
Gambar 4.5 Tampilan Cetak Kartu Parkir	38
Gambar 4.6 Prototipe Lot Parkir	38
Gambar 4.7 Monitoring Perparkiran (Lot berwarna hijau)	39
Gambar 4.8 Pemasukan Nomor Plat Kendaraan.....	40
Gambar 4.9 terisi_kosong ter-update “2”	40

Gambar 4.10 <i>Monitoring</i> Perparkiran (Lot berwarna kuning).....	41
Gambar 4.11 <i>Monitoring</i> Perparkiran (Lot berwarna merah).....	41
Gambar 4.12 terisi_kosong ter- <i>update</i> “1”	42
Gambar 4.13 Prototipe Lot Parkir.....	42
Gambar 4.14 Pemberian Kartu Parkir.....	43
Gambar 4.15 Kendaraan Yang Terparkir.....	43
Gambar 4.16 <i>Monitoring</i> Perparkiran.....	44
Gambar 4.17 terisi_kosong ter- <i>update</i>	44
Gambar 4.18 tabel_log ter- <i>update</i>	45
Gambar 4.19 <i>Monitoring</i> Perparkiran.....	45
Gambar 4.20 terisi_kosong ter- <i>update</i>	45
Gambar 4.21 tabel_log ter- <i>update</i>	46
Gambar 4.22 Grafik Waktu Respon Perubahan Warna Lot.....	49
Gambar 4.23 Grafik Kinerja Waktu Kendaraan ke Lot Parkir	50
Gambar 4.24 Pengujian Sensor Ultrasonik HC-SR04	51
Gambar 4.25 Pengujian Sensor Ultrasonik PING.....	52

DAFTAR TABEL

Halaman

Tabel 3. 1 Tabel Perangkat Keras	24
Tabel 3. 2 Tabel Perangkat Lunak	25
Tabel 3. 3 Tabel Daftar Kebutuhan Fungsional	27
Tabel 4. 1 Pengujian Fungsionalitas Lot Parkir	46
Tabel 4. 2 Pengujian Waktu Respon Perubahan Warna Lot	48
Tabel 4. 3 Pengujian Kinerja Waktu Kendaraan ke Lot Parkir.....	50

DAFTAR LAMPIRAN

Halaman

Lampiran 1. Pengujian jarak sensor ultrasonik PING.....	57
Lampiran 2. Pengujian Pub/Sub menggunakan MQTT Lens	57
Lampiran 3. Source code Mikrokontroller ESP8266.....	58
Lampiran 4. Source code update & insert database dan connect ke MQTT Broker.....	63
Lampiran 5. Konfigurasi Access Point Bridge	65

SURAT PERNYATAAN

Saya bertanda tangan dibawah ini :

Nama : Yori Putra Pradana

Nim : 425 15 014

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul “*Monitoring Perparkiran Mobil Menggunakan Sensor Ultrasonik Ping Berbasis Wireless*” merupakan gagasan dan hasil karya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manapun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut diatas tidak benar, saya siap menanggung risiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 27 Juni 2019



Yori Putra Pradana

425 15 014

MONITORING PERPARKIRAN MOBIL MENGGUNAKAN SENSOR ULTRASONIK PING BERBASIS WIRELESS

RINGKASAN

Pertambahan jumlah kendaraan bermotor di Indonesia setiap tahun meningkat pesat. Berdasarkan data Badan Pusat Statistik (BPS) tahun 2017, angka kendaraan bermotor dengan jenis kendaraan Mobil Penumpang mencapai 15.493.068 unit. Jumlah tersebut mengalami peningkatan dibandingkan dengan tahun 2016 sebesar 14.580.666 unit. Peningkatan jumlah kendaraan bermotor berdampak pada ketersediaan lahan parkir di sejumlah lokasi. Setiap pengendara yang akan memarkirkan kendaraanya menginginkan untuk mendapatkan lokasi/lot parkir yang kosong secepatnya. Namun tidak tersedianya informasi kapasitas parkir pada suatu area parkir menyebabkan pengguna harus berkeliling terlebih dahulu untuk mendapatkan tempat parkir yang kosong, apabila lahan parkir penuh pengguna bahkan harus keluar kembali karena tidak mendapatkan tempat parkir. Penelitian ini merancang dan membangun suatu alat mikrokontroller ESP8266 dan sensor ultrasonik PING yang memberikan informasi terhadap ketersediaan lot parkir yang masih kosong. Dengan adanya sistem monitoring ini diharapkan dapat meningkatkan efisiensi dari pengguna parkir yang mencari lot parkir yang kosong serta memudahkan admin parkir dalam manajemen *monitoring* perparkiran. Hasil penelitian menunjukkan bahwa sistem *monitoring* yang telah dibangun mampu memberikan informasi terhadap ketersediaan lot parkir yang kosong dan terintegrasi database untuk menyimpan informasi pemilik kendaraan berdasarkan kartu parkir. Informasi lot parkir di cantumkan pada kartu parkir yang diberikan kepada *user* yang memasuki area parkir.

Kata Kunci : *Monitoring, Perparkiran, Mikrokontroller ESP8266, Sensor Ultrasonik PING*

BAB I PENDAHULUAN

1.1 Latar Belakang

Pertambahan jumlah kendaraan bermotor di Indonesia setiap tahun meningkat pesat. Berdasarkan data Badan Pusat Statistik (BPS) tahun 2017, angka kendaraan bermotor dengan jenis kendaraan Mobil Penumpang mencapai 15.493.068 unit. Jumlah tersebut mengalami peningkatan dibandingkan dengan tahun 2016 sebesar 14.580.666 unit (Badan Pusat Statistik, 2017). Peningkatan jumlah kendaraan bermotor berdampak pada ketersediaan lahan parkir di sejumlah lokasi (Detik.com, 2013).

Kebijakan pemerintah untuk mengurangi adanya parkir di area jalan dilakukan dengan mengharuskan pusat-pusat kegiatan bisnis (*business district*), rumah sakit, perkantoran baik swasta maupun pemerintah menyediakan suatu ruang parkir yang cukup (memenuhi standar). Penyediaan ruang parkir yang cukup pada pusat-pusat kegiatan bisnis (*mall*) umumnya dalam bentuk lahan parkir bertingkat. (Limantara dkk, 2017).

Setiap pengendara yang akan memarkirkan kendaraanya menginginkan untuk mendapatkan lokasi/lot parkir yang kosong secepatnya. Namun tidak tersedianya informasi kapasitas parkir pada suatu area parkir menyebabkan pengguna harus berkeliling terlebih dahulu untuk mendapatkan tempat parkir yang kosong, apabila lahan parkir penuh pengguna bahkan harus keluar kembali karena tidak mendapatkan tempat parkir. Untuk itu perlu dilakukan upaya pengembangan sistem *monitoring* perparkiran yang menerapkan peralatan sensor sehingga informasi lot parkir yang kosong pada area parkir dapat diperoleh oleh pengguna

cepat dan akurat, hal ini akan sangat membantu pengguna dalam proses pencarian tempat parkir.

Pada penelitian sebelumnya oleh (Rudi dkk, 2017), dengan judul Rancang Bangun *Prototype Sistem Smart Parking* Berbasis Arduino Dan Pemantauan Melalui *Smartphone*. Telah merancang *Smart Parking* yang dapat menginformasikan dan mengarahkan pengendara mobil ke area parkir yang kosong. Lahan parkir yang dijadikan sebagai objek penelitiannya terdiri dari beberapa lokasi parkir dengan kapasitas beberapa kendaraan. Pada perancangan ini memiliki beberapa bagian umum yang digunakan, yaitu sensor ultrasonik, Arduino Mega, PC/laptop, LCD (*Liquid Cristal Display*) dan IOT (*Internet Of Things*). Penelitian menunjukkan bahwa LCD akan menampilkan beberapa slot parkir yang sudah terisi dan beberapa lagi slot parkir yang kosong dan pemantauan melalui *Smartphone* dengan menggunakan aplikasi Blynk.

Pada penelitian lainnya dilakukan oleh (Hakim, 2016), dengan judul Model *Smart Parking* Berbasis *Internet Of Things*. Merancang suatu alat mikrokontroler yang dapat menginformasikan lahan parkir kepada pengendara mobil di salah satu area parkir tertentu. Pada perancangan ini menggunakan sensor LDR, sensor ultrasonik, Arduino Uno, Module ESP8266, PC/laptop, web parkir. Sensor ultrasonik mendeteksi adanya kendaraan sedangkan sensor LDR akan mengirim data ke Module ESP8266, ada atau tidaknya mobil yang sedang parkir kemudian ESP8266 akan melanjutkannya ke web parkir. Mikrokontroler arduino uno sebagai tempat pemrosesan data dari semua sensor. Namun, dari rangkaian penelitian ini masih dalam bentuk model sehingga perlu diimplementasikan pada bidang

sesungguhnya selain itu skala yang masih berupa model terlihat lebih rumit dari pada rangkaian yang seungguhnya.

Berdasarkan kondisi diatas maka dibangun suatu alat mikrokontroller yang memberikan informasi terhadap ketersediaan lot parkir yang masih kosong. Ketersediaan lot parkir tersebut di *monitoring* oleh admin parkir melalui sistem *monitoring* perparkiran. Dengan adanya sistem *monitoring* ini diharapkan dapat meningkatkan efisiensi dari pengguna parkir yang mencari lot parkir yang kosong serta memudahkan admin parkir dalam manajemen *monitoring* perparkiran.

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang di atas, rumusan masalah yang menjadi fokus dalam penelitian ini adalah :

1. Bagaimana merancang sistem *monitoring* perparkiran menggunakan Mikrokontroller ESP8266 dan sensor ultrasonik?
2. Bagaimana perancangan sistem *monitoring* perparkiran yang dapat memberikan informasi terhadap ketersediaan lot parkir yang masih kosong ?
3. Bagaimana menguji fungsionalitas sistem *monitoring* perparkiran ?

1.3 Batasan Masalah

Batasan masalah dari pembuatan penelitian ini adalah :

1. Sistem *monitoring* perparkiran dibuat untuk kendaraan mobil.
2. Sensor yang digunakan berupa sensor ultrasonik.
3. Sistem *monitoring* perparkiran yang dibuat dapat dimonitoring melalui aplikasi *monitoring* perparkiran.

4. Sistem *monitoring* perparkiran yang diusulkan ditujukan pada sistem parkiran yang tidak membayar/gratis.

1.4 Tujuan Penelitian

Adapun penelitian ini bertujuan untuk :

1. Membuat/merancang sistem *monitoring* perparkiran.
2. Membangun sistem *monitoring* perparkiran untuk memudahkan menemukan lot parkir yang masih kosong.
3. Menguji fungsionalitas sistem *monitoring* perparkiran.

1.5 Manfaat Penelitian

Adapun manfaat dari penulisan penelitian ini adalah :

1. Dengan dikembangkan sistem *monitoring* perparkiran ini dapat memudahkan masyarakat untuk mendapatkan lot parkir yang kosong.
2. Bagi kalangan akademis dapat digunakan sebagai bahan informasi studi lanjutan yang di perlukan untuk pengembangan yang berkaitan dengan pembuatan *smart parking*.

BAB II TINJAUAN PUSTAKA

2.1 *Monitoring*

Monitoring adalah pemantauan yang dapat memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi atau kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan (Mardiani, 2013).

2.2 Parkir

Kendaraan tidak mungkin bergerak terus-menerus, akan ada waktunya kendaraan itu harus berhenti, baik itu bersifat sementara maupun bersifat lama atau biasa yang disebut parkir. Menurut Martadipura (2013) Parkir adalah suatu keadaan dimana suatu kendaraan yang bersifat sementara ditinggalkan oleh pengemudinya. Definisi parkir menurut undang- undang nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, Bab I Ketentuan Umum, pada Pasal 1 angka 15 dan 16 tertulis sebagai berikut:

Pasal 1 ayat 15 yang dimaksud dengan: Parkir adalah keadaan kendaraan berhenti atau tidak bergerak untuk beberapa saat dan ditinggalkan pengemudinya. Sedangkan ayat 16, berhenti adalah keadaan kendaraan tidak bergerak untuk sementara dan tidak ditinggalkan pengemudinya.

Terdapat dua jenis parkir yaitu parkir yang berada di badan jalan (*on street parking*) dan parkir yang berada luar badan jalan (*off street parking*). Pada *on street parking* membutuhkan badan jalan untuk digunakan sebagai tempat parkir, contoh adalah parkir di depan pertokoan dimana kendaraan berhenti di pinggir jalan. Sedangkan untuk *off street parking* tidak membutuhkan badan jalan untuk sebagai tempat parkir melainkan dipersiapkan sebuah lahan khusus untuk menampung kendaraan parkir (Martadipura, 2013).

Dari pengertian diatas dapat disimpulkan bahwa parkir adalah keadaan kendaraan diam atau ditinggalkan pengemudinya sementara.

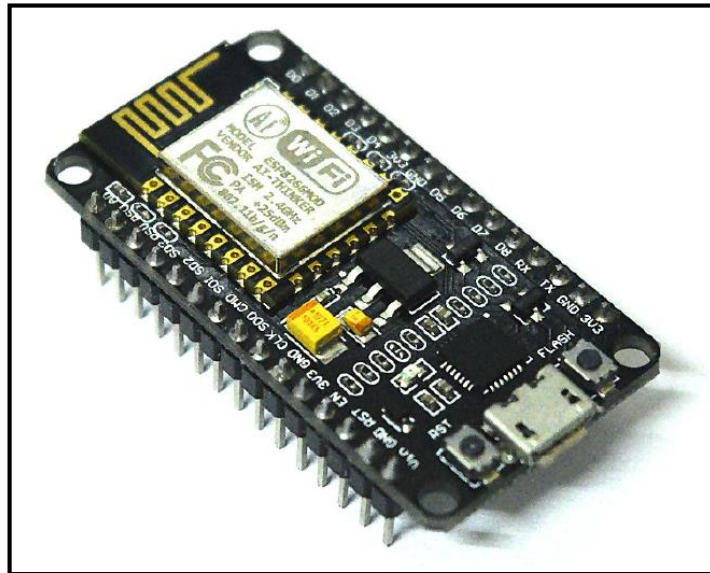
2.3 Mikrokontroller

Mikrokontroller merupakan sebuah sistem komputer fungsional yang di letakkan dalam papan elektronik yang berukuran mikro atau kecil. Di dalam mikrokontroller terdapat sebuah prosesor, memori, serta komponen *input* dan *output*. Dengan begitu, mikrokontroller adalah suatu alat elektronika digital yang memiliki *input* serta *output* yang kendalinya dapat diprogram ulang dengan suatu cara khusus. Ada banyak pilihan mikrokontroller yang telah terdapat modul jaringan agar dapat langsung terhubung ke jaringan internet sehingga dapat di implimentasikan untuk membuat peralatan berbasis IoT/WSN, contohnya NodeMCU ESP8266.

2.3.1 NodeMCU ESP8266

Merupakan *single board microcontroller* yang ditenagai memori 128kBytes, tempat penyimpanan 4Mbytes, dan bersumber daya dari USB.

Mikrokontroller ini memiliki modul WiFi dan *firmware* menggunakan bahasa pemrograman LUA atau Arduino IDE (Putra, Lhaksana, & Adytia, 2018).

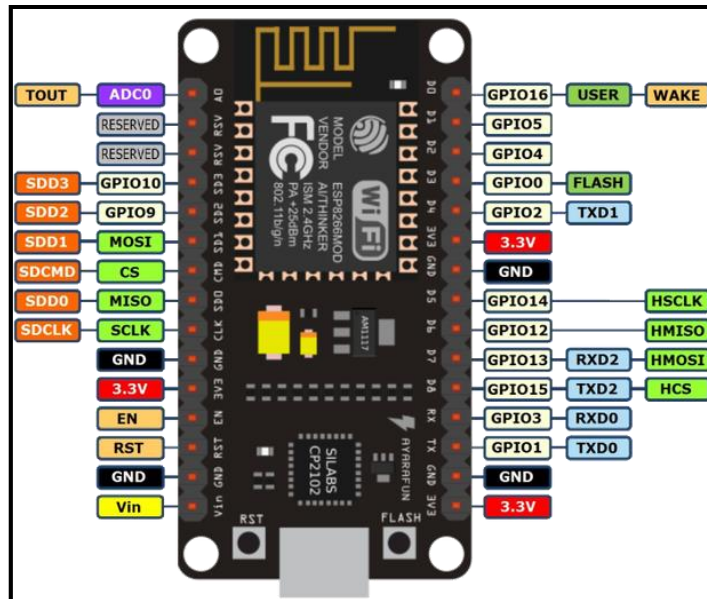


Gambar 2.1 NodeMCU ESP8266
Sumber: (EINSTRONIC, 2017)

1. Fitur-fitur:

- a) Versi: DevKit v1.0
- b) *Breadboard Friendly*
- c) Ringan dan ukuran kecil.
- d) 3.3V dioperasikan, dapat didukung USB.
- e) Menggunakan protokol nirkabel 802.11b / g / n.
- f) Kemampuan konektivitas nirkabel built-in.
- g) Built-in PCB antenna pada chip ESP-12E.
- h) Mampu PWM, I2C, SPI, UART, 1-kawat, 1 pin analog.
- i) Menggunakan modul antarmuka Komunikasi Serial CP2102 USB.
- j) Arduino IDE kompatibel (diperlukan manajer papan ekstensi).
- k) Mendukung Lua (node.js) dan bahasa pemrograman Arduino C.

2. Pinout Diagram (EINSTRONIC, 2017)



Gambar 2.2 Pinout Diagram
Sumber: (EINSTRONIC, 2017)

2.4 Sensor Ultrasonik

Sensor adalah peralatan yang digunakan untuk merubah suatu besaran fisik menjadi besaran listrik sehingga dapat dianalisa dengan rangkaian listrik tertentu. Hampir seluruh peralatan elektronik yang ada mempunyai sensor didalamnya. Pada saat ini, sensor tersebut telah dibuat dengan ukuran sangat kecil dengan orde nanometer. Ukuran yang sangat kecil ini sangat memudahkan pemakaian dan menghemat energi (Budiarso & Prihandono, 2015).

Sensor ultrasonik adalah sebuah sensor yang mengubah besaran fisis (bunyi) menjadi besaran listrik. Pada sensor ini gelombang ultrasonik dibangkitkan melalui sebuah benda yang disebut *piezoelektrik*. *Piezoelektrik* ini akan menghasilkan gelombang ultrasonik dengan frekuensi 40 kHz ketika sebuah osilator diterapkan pada benda tersebut. Sensor ultrasonik secara umum digunakan untuk suatu

pengungkapan tak sentuh yang beragam seperti aplikasi pengukuran jarak. Alat ini secara umum memancarkan gelombang suara ultrasonik menuju suatu target yang memantulkan balik gelombang kearah sensor. Kemudian sistem mengukur waktu yang diperlukan untuk pemancaran gelombang sampai kembali kesensor dan menghitung jarak target dengan menggunakan kecepatan suara dalam medium. Rangkaian penyusun sensor ultrasonik ini terdiri dari *transmitter*, *reiceiver*, dan komparator. Selain itu, gelombang ultrasonik dibangkitkan oleh sebuah kristal tipis bersifat *piezoelektrik* (Arief, 2011). Bagian-bagian dari sensor ultrasonik adalah sebagai berikut :

a. *Piezoelektrik*

Peralatan *piezoelektrik* secara langsung mengubah energi listrik menjadi energi mekanik. Tegangan *input* yang digunakan menyebabkan bagian keramik meregang dan memancarkan gelombang ultrasonik. Tipe operasi transmisi elemen *piezoelektrik* sekitar frekuensi 32 kHz. Efisiensi lebih baik, jika frekuensi osilator diatur pada frekuensi resonansi *piezoelektrik* dengan sensitifitas dan efisiensi paling baik. Jika rangkaian pengukur beroperasi pada mode pulsa elemen *piezoelektrik* yang sama dapat digunakan sebagai *transmitter* dan *reiceiver*. Frekuensi yang ditimbulkan tergantung pada osilatornya yang disesuaikan frekuensi kerja dari masing- masing transduser. Karena kelebihanannya inilah maka tranduser *piezoelektrik* lebih sesuai digunakan untuk sensor ultrasonik.

b. *Transmitter*

Transmitter adalah sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi sebesar 40 kHz yang dibangkitkan dari sebuah osilator.

Untuk menghasilkan frekuensi 40 kHz, harus di buat sebuah rangkaian osilator dan keluaran dari osilator dilanjutkan menuju penguat sinyal. Besarnya frekuensi ditentukan oleh komponen kalang RLC / kristal tergantung dari disain osilator yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik yang diumpankan ke *piezoelektrik* dan terjadi reaksi mekanik sehingga bergetar dan memancarkan gelombang yang sesuai dengan besar frekuensi pada osilator.

c. *Receiver*

Receiver terdiri dari transduser ultrasonik menggunakan bahan *piezoelektrik*, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakan pada permukaan suatu benda atau gelombang langsung LOS (*Line of Sight*) dari *transmitter*. Oleh karena bahan *piezoelektrik* memiliki reaksi yang *reversible*, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan menggetarkan bahan *piezoelektrik* tersebut (Arief, 2011).

2.4.1 Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 merupakan sensor jarak yang memanfaatkan gelombang suara ultrasonik didalam proses pengukuran jarak suatu objek. Modul HC-SR04 memiliki 4 pin yaitu pin VCC (5 volt), pin ground, pin echo serta pin trigger. Pin trigger berfungsi untuk memberikan pulsa trigger, sedangkan pin echo adalah pin yang berfungsi untuk memberitahukan rentang waktu pantulan suara (Permana, Triyanto, & Rismawan, 2015).



Gambar 2.3 Sensor Ultrasonik HC-SR04
Sumber: (Amazon.in)

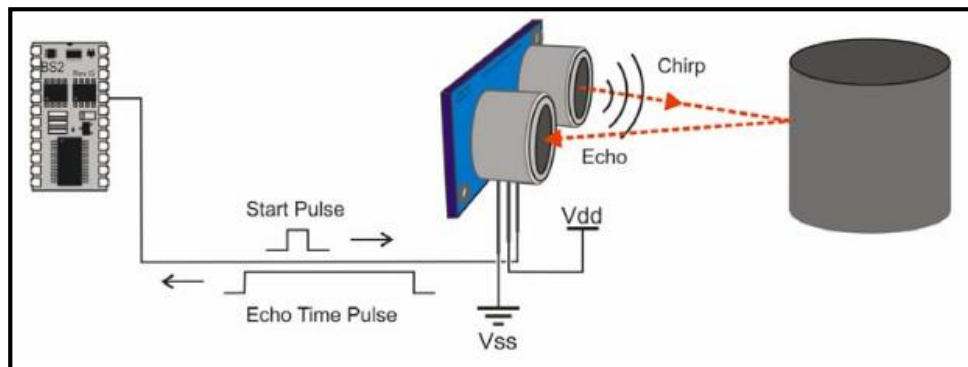
2.4.2 Sensor Ultrasonik PING Parallax

Modul sensor ultrasonik ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari modul sensor ultrasonik Ping ini berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya yang dihasilkan modul sensor ultrasonik ini bervariasi dari 115 μ S sampai 18,5 mS. Secara prinsip modul sensor ultrasonik ini terdiri dari sebuah chip pembangkit sinyal 40 kHz, sebuah *speaker* ultrasonik dan sebuah mikropon ultrasonik. *Speaker* ultrasonik mengubah sinyal 40 kHz menjadi suara sementara mikropon ultrasonik berfungsi untuk mendeteksi pantulan suaranya (Arief, 2011). Bentuk sensor ultrasonik diperlihatkan pada Gambar 2.5 berikut.



Gambar 2.4 Sensor Ultrasonik PING Parallax
Sumber: (Arief, 2011)

Sinyal *output* modul sensor ultrasonik dapat langsung dihubungkan dengan mikrokontroler tanpa tambahan komponen apapun. Modul sensor ultrasonik hanya akan mengirimkan suara ultrasonik ketika ada pulsa trigger dari mikrokontroler (Pulsa high selama 5 μ S). Suara ultrasonik dengan frekuensi sebesar 40 kHz akan dipancarkan selama 200 μ S oleh modul sensor ultrasonik ini. Suara ini akan merambat di udara dengan kecepatan 344.424 m/detik (atau 1cm setiap 29.034 μ S) yang kemudian mengenai objek dan dipantulkan kembali ke modul sensor ultrasonik tersebut. Selama menunggu pantulan sinyal ultrasonik dari bagian *transmitter*, modul sensor ultrasonik ini akan menghasilkan sebuah pulsa. Pulsa ini akan berhenti (low) ketika suara pantulan terdeteksi oleh modul sensor ultrasonik. Oleh karena itulah lebar pulsa tersebut dapat merepresentasikan jarak antara modul sensor ultrasonik dengan objek (Arief, 2011).

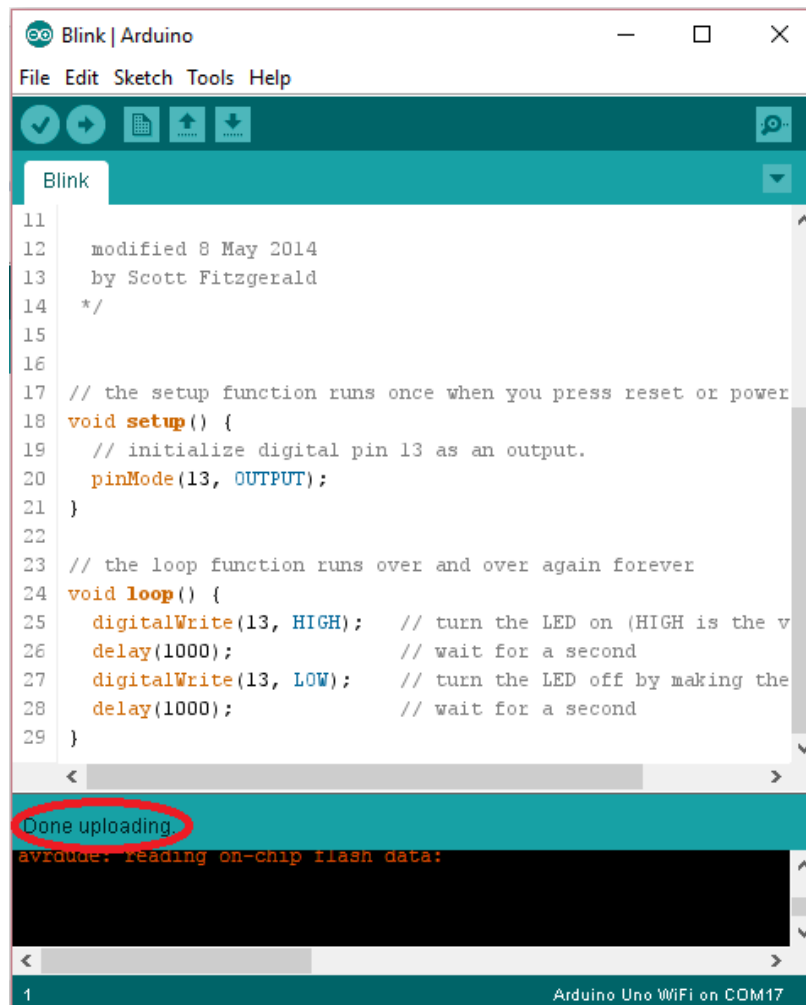


Gambar 2.5 Prinsip Kerja Sensor Ultrasonik PING
Sumber: (Budiarso, 2015)

Budiarso (2015) menuliskan dalam penelitiannya bahwa sensor ultrasonik parallax ping terdiri dari sensor, chip pembangkit gelombang, penerima gelombang dan pembangkit pulsa. Ketika rangkaian elektronik dari parallax ping mendapat catu daya, maka akan dihasilkan pulsa-pulsa yang akan dikirim oleh bagian transmitter. Sensor akan mendeteksi adanya sebuah objek yang berada di depan sensor, yang ditandai dengan adanya sinyal yang diterima oleh sensor penerima pulsa. Jarak tempuh pulsa dianggap sebagai dua kali jarak sensor dengan objek.

2.5 Arduino IDE

Yusuf (2017) dalam skripsinya menuliskan bahwa Arduino mempunyai sebuah *software* Arduino, dimana *software* tersebut digunakan untuk menuliskan program dan dimasukkan kedalam *memory* Arduino. Bahasa yang digunakan pada *software* Arduino adalah bahasa pemrograman C++. *Software* arduino dapat diinstal diberbagai operasi sistem, seperti Linux, Mac OS, dan Windows.



Gambar 2.6 Tampilan Perangkat Lunak Arduino IDE
Sumber: (Arduino, 2018)

Software IDE (*Integrated Development Environment*) terdiri dari tiga bagian, antara lain:

- b. *Editor program*: untuk mengedit dan menulis program dalam bahasa processing. *Listing* program pada arduino disebut *sketch*.
- c. *Compiler*: modul yang berfungsi mengubah bahasa *processing* (kode program) ke bentuk kode biner karena kode biner adalah satu-satunya bahasa yang dipahami mikrokontroler.

- d. *Uploader*: modul yang berfungsi memasukan kode biner kedalam memori mikrokontroller (Lumba, 2017).

2.6 Database

2.6.1 Pengertian Database

Database merupakan kumpulan data yang mempunyai relasi logika yang dapat dibagikan, mempunyai deskripsi pada data dan didesain untuk mendapatkan informasi yang dibutuhkan oleh organisasi. (Connolly & Begg, 2010).

Untuk mendapatkan informasi yang dibutuhkan oleh suatu organisasi, maka kita mencoba untuk mengidentifikasi entities, attributes dan relationships (Connolly et al. 2010).

1. *Entity* merupakan sebuah perbedaan objek seperti orang, tempat, benda, konsep atau acara pada suatu organisasi yang dapat merepresentasikan *database*.
2. *Attributes* merupakan sifat yang dideskripsikan pada beberapa pandangan objek yang kita harapkan untuk menyimpan.
3. *Relationship* merupakan sebuah asosiasi antara entities dan attributes.

2.6.2 Database Management System (DBMS)

DBMS adalah sebuah sistem *software* yang memungkinkan pengguna untuk mendefinisikan, membuat, memelihara dan mengontrol akses ke *database*. DBMS merupakan *software* yang berinteraksi dengan aplikasi program pengguna dan *database*. DBMS menyediakan fasilitas seperti:

1. Menyediakan akses kontrol ke *database* seperti sistem keamanan untuk mencegah pengguna yang tidak mempunyai hak akses untuk mengakses *database*, sistem integritas dimana untuk memelihara konsistensi data yang di simpan, sistem

kontrol konkurensi dimana untuk memperbolehkan pengguna yang bersangkutan untuk mengakses *database* pada waktu yang bersamaan, sistem kontrol pemulihan dimana digunakan untuk memulihkan *database* ke keadaan sebelumnya yang diakibatkan oleh kegagalan *software* atau *hardware*, katalog pengguna yang mengakses dimana menyimpan deskripsi data pada *database*

2. Memperbolehkan pengguna untuk *insert*, *update*, *delete* dan *retrieve* data dari sebuah *database* yang biasanya menggunakan *Data Manipulation Language* (DML). Bahasa Query merupakan data deskripsi yang memperbolehkan DML untuk menyediakan fasilitas permintaan umum pada data. *Structured Query Language* (SQL) merupakan bahasa Query yang paling umum digunakan dan juga merupakan standar bahasa untuk DBMS yang berelasi.

3. Memperbolehkan pengguna untuk mendefinisikan *database* yang biasanya menggunakan sebuah *Data Definition Language* (DDL). DDL memperbolehkan pengguna untuk mengelompokkan tipe data, struktur dan batasan data untuk disimpan dalam *database*. (Connolly & Begg, 2010).

2.6.3 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang cepat dan kokoh. Sebuah *database* memungkinkan untuk menyimpan, mencari, mengurutkan dan mendapatkan data secara efisien. Server MySQL mengontrol akses data untuk memastikan beberapa pengguna dapat bekerja dengan MySQL dengan menyediakan akses yang cepat dan memastikan hanya pengguna yang mempunyai hak yang dapat mengakses data. Oleh karena itu, server MySQL banyak digunakan dan multithreading. MySQL menggunakan *Structured Query*

Language (SQL) yang merupakan standar dari bahasa Query (Welling, Luke, 2009).

2.7 Protokol MQTT

Protokol MQTT merupakan sebuah protokol pertukaran pesan dengan model *publish/subscribe* yang sederhana dan ringan serta didesain untuk perangkat yang memiliki kemampuan terbatas dan bandwidth yang kecil, latency tinggi, atau jaringan yang tidak andal. Prinsip desain MQTT adalah untuk meminimalisasi bandwidth jaringan dan kebutuhan resource perangkat dan tetap menjamin keandalan dan beberapa tingkat jaminan tersampainya sebuah pesan. Prinsip inilah yang membuat protokol ini ideal untuk diaplikasikan pada komunikasi machine-to-machine (M2M) atau Internet of Things dan untuk aplikasi mobile dimana bandwidth dan kapasitas baterai terbatas (MQTT, 2019).

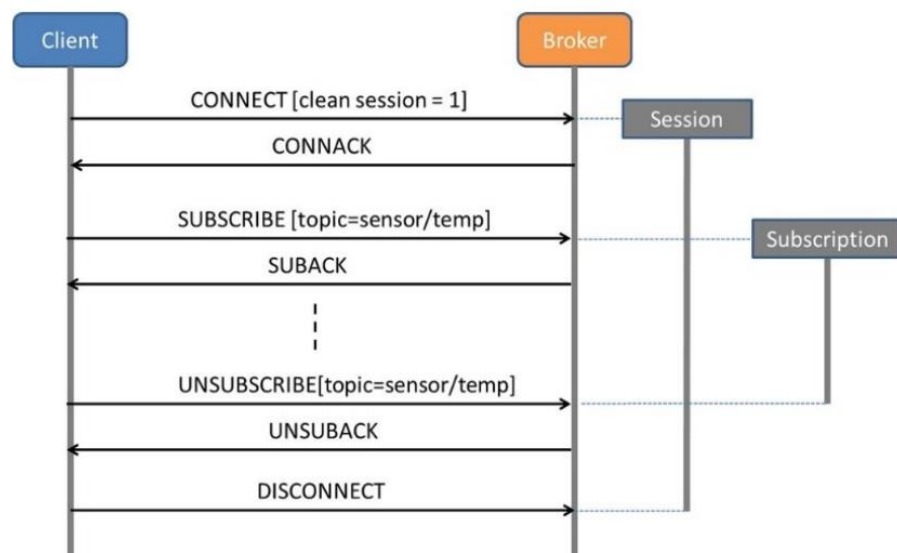
Pertukaran pesan dengan model *publish/subscribe* pada MQTT merupakan alternatif dari model *client-server*, dimana sebuah *client* (*publisher/subscriber*) berkomunikasi langsung dengan sebuah endpoint lainnya pada sebuah topik melalui sebuah broker yang bertugas melakukan penyaringan pesan dan mendistribusikannya.

Terdapat beberapa alur pesan umum pada MQTT:

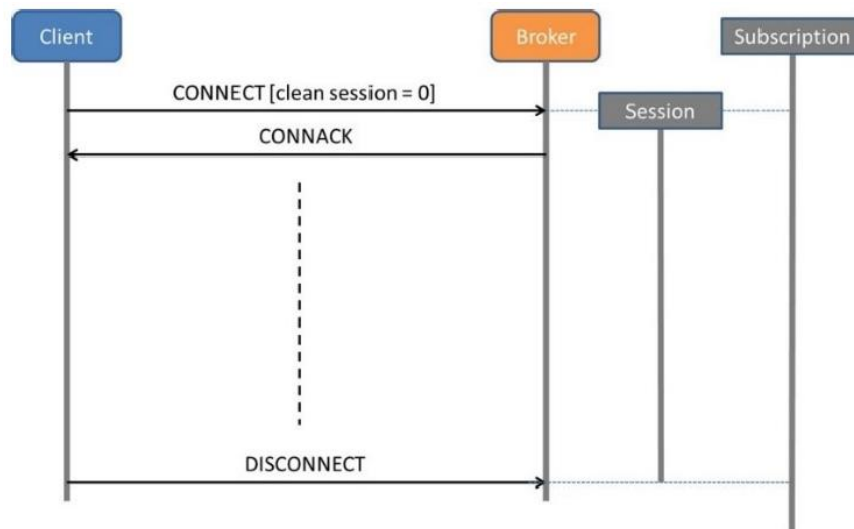
- 1) Alur pesan ketika *client* melakukan koneksi ke broker.

Sebuah *client* melakukan koneksi ke broker dengan mengirimkan sebuah pesan *CONNECT* dan broker akan merespon dengan sebuah pesan *CONNACK*. Ketika sebuah *client* terhubung akan terbentuk sebuah session dan dapat melakukan *subscribe* ke 1 atau lebih topik untuk menerima pesan. Pesan *CONNECT* memiliki

parameter clean session yang dapat bernilai 1 (*true*) atau 0 (*false*). Jika bernilai 1, maka ketika *client* terputus, semua *subscribe* yang dilakukan *client* akan di-*unsubscribe* oleh broker. Sementara jika bernilai 0, ketika *client* terputus dan terhubung kembali, semua *subscribe* pada *session* sebelumnya akan ter-*subscribe* secara otomatis (broker tidak melakukan *unsubscribe*). Alur dapat dilihat pada Gambar 2.7 dan 2.8 (Ppatierno, 2013)



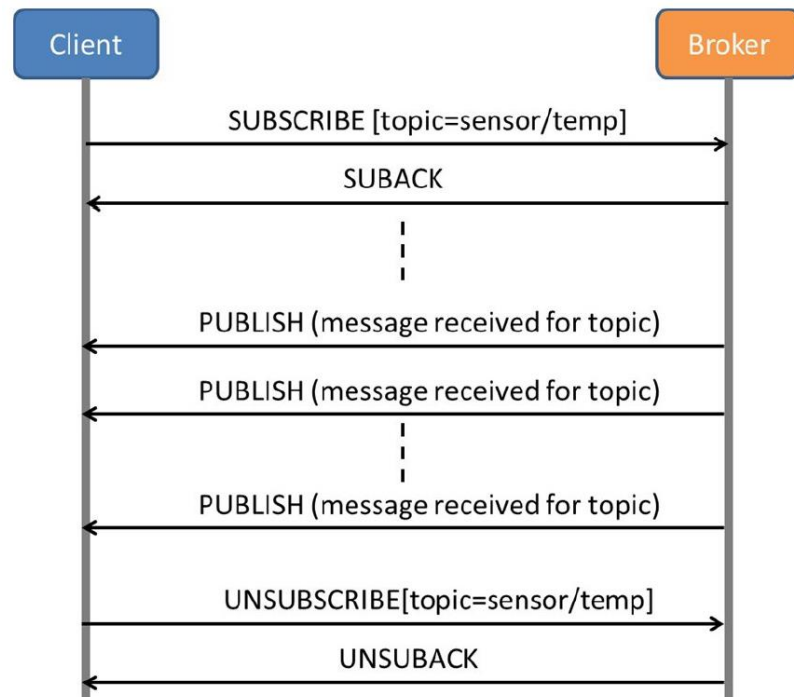
Gambar 2.7 Alur pesan *CONNECT* dengan *clean session*
Sumber: (Ppatierno, 2011)



Gambar 2.8 Alur pesan *CONNECT* tanpa *clean session*
 Sumber: (Ppatierno, 2011)

2) Alur pesan ketika *client* melakukan *subscribe* ke sebuah topik dan ketika sebuah pesan tiba.

Sebuah *client* melakukan subscription ke sebuah topik dengan mengirimkan sebuah pesan *SUBSCRIBE* dan broker akan merespon dengan sebuah pesan *SUBSCRIBEACK*. Ketika sebuah *client* sudah ter-*subscribe* pada sebuah topik, maka *client* dapat menerima semua pesan yang di-publish pada topik tersebut sampai *client* melakukan *unsubscribe* dengan mengirimkan pesan *UNSUBSCRIBE* ke broker dan menerima pesan *UNSUBACK*. Alur dapat dilihat pada Gambar 2.9

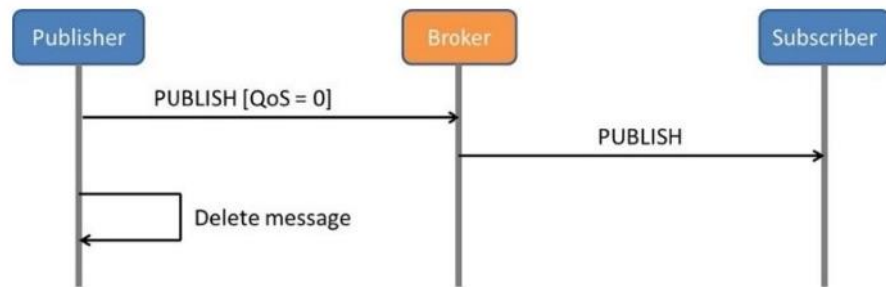


Gambar 2.9 Alur pesan *SUBSCRIBE* and *UNSUBSCRIBE* untuk sebuah topik

Sumber: (Ppatierno, 2011)

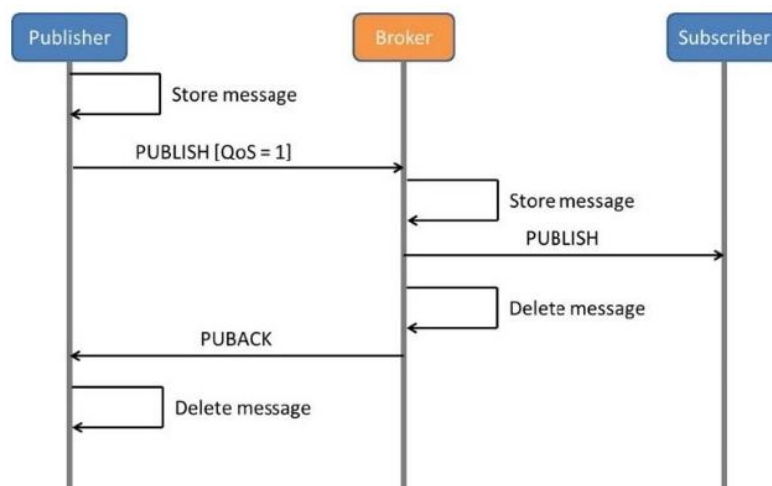
3) Alur pesan ketika *client* melakukan publish ke sebuah topik dengan 3 tingkat QoS.

Sebuah *client* dapat melakukan *publish* sebuah pesan pada sebuah topik dengan menggunakan salah satu dari 3 tingkat QoS. Pada QoS 0, pesan terkirim paling banyak sekali atau tidak terkirim sama sekali. Pesan dinyatakan telah terkirim jika pesan telah sampai pada broker dan broker akan meneruskannya ke semua *subscriber* pada topik tersebut. *Client* yang melakukan *publish* tidak menunggu *acknowledge* dari broker dan tidak menyimpan pesan. Alur untuk QoS 0 dapat dilihat pada Gambar 2.10.



Gambar 2.10 Alur pesan *PUBLISH* dengan QoS 0 (*At most once*)
 Sumber: (Ppatierno, 2011)

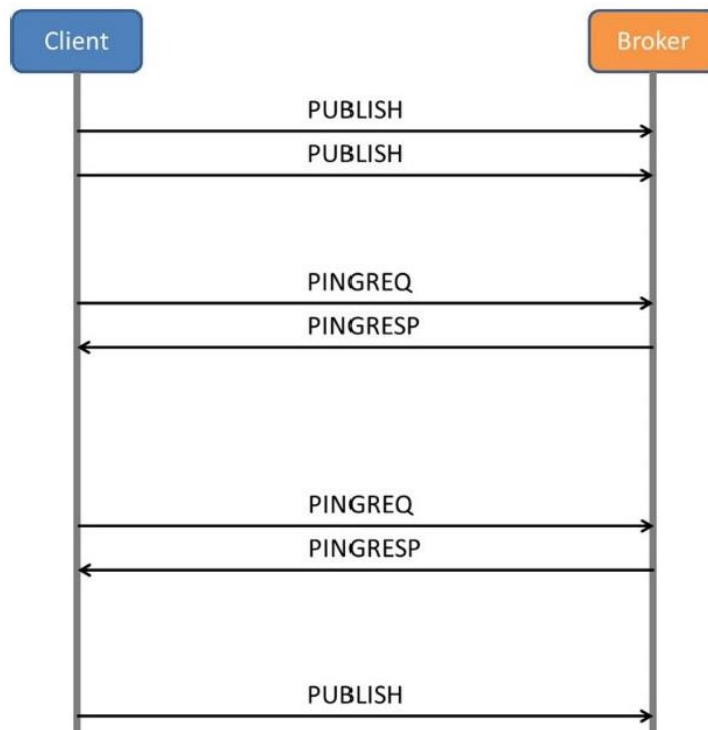
Pada QoS 1, pesan selalu terkirim paling sedikit sekali. Ketika *client* melakukan *publish* pesan, maka *client* dan broker menyimpannya secara lokal. Pesan yang tersimpan pada broker akan disimpan sampai selesai diproses dan broker akan mengirimkan pesan PUBACK setelahnya. Sementara, pesan yang tersimpan pada sisi *client* akan disimpan sampai menerima pesan PUBACK dari broker. Jika *client* tidak kunjung menerima pesan PUBACK, *client* akan mengirimkan kembali pesan tersebut, sehingga pesan yang sama dapat terkirim beberapa kali. Alur untuk QoS 1 dapat dilihat pada Gambar 2.11.



Gambar 2.11 Alur pesan *PUBLISH* dengan QoS 1 (*At least once*)
 Sumber: (Ppatierno, 2011)

4) Alur pesan untuk keepalive antara *client* dan broker.

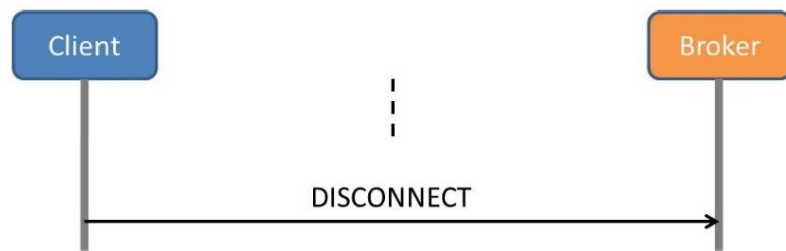
Ketika *client* tidak mengirimkan pesan dalam waktu tertentu, koneksi perlu dijaga agar tetap terhubung dengan *client* mengirimkan pesan PINGREQ ke broker setiap periode waktu tertentu sesuai dengan yang sudah ditentukan sebelumnya. Broker akan membalas pesan PINGREQ dengan pesan PINGRESP. Alur untuk *keepalive connection* dapat dilihat pada Gambar 2.13.



Gambar 2.12 Alur pesan ping untuk *keepalive*
Sumber: (Ppatierno, 2011)

5) Alur pesan untuk menutup koneksi.

Untuk menutup koneksi, *client* hanya perlu mengirimkan pesan *DISCONNECT* kepada broker tanpa pesan ada balasan. Alur untuk pemutusan koneksi dapat dilihat pada Gambar 2.14.



Gambar 2.13 Alur pesan *DISCONNECT* untuk memutuskan koneksi
Sumber: (Ppatierno, 2011)

BAB III METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

Penelitian dilaksanakan di Kampus Politeknik Negeri Ujung Pandang Jl. Perintis Kemerdekaan KM 10 Makassar, dimulai pada bulan November 2018 sampai bulan Juni 2019.

3.2 Perangkat Penelitian

Penelitian ini dimulai dengan melakukan identifikasi kebutuhan yang mencakup informasi tentang perangkat lunak, perangkat keras, sistem operasi dan aplikasi pendukung lainnya yang digunakan pada penelitian ini.

Adapun spesifikasi kebutuhan sebagai berikut:

3.2.1 Perangkat Keras

Pada tabel 3.1 ditunjukkan perangkat keras dan spesifikasinya yang digunakan.

Tabel 3.1 Tabel Perangkat Keras

No	Perangkat Keras	Deskripsi
1	1 Unit Laptop ASUS	Processor : Intel® Core™ i7-7700HQ, CPU @2.80GHz (8 CPUs). Memory : 8192 MB RAM. HDD : 1 TB.
2	4 Sensor Ultrasonik PING	Frekuensi kerja : 40 kHz untuk 200 us Antarmuka digital (1 pin I/O dengan level TTL) Jarak pengukuran : 2 - 300 cm
3	5 Mikrokontroler ESP8266	CPU: @ 80 MHz (default) or 160 MHz Memori: 32 KiB instruction, 80 KiB user data Wi-Fi : 80.11 b/g/n standard
4	Power Supply/Catu daya	

3.2.2 Perangkat Lunak

Pada tabel 3.2 ditunjukkan perangkat lunak dan spesifikasinya yang digunakan.

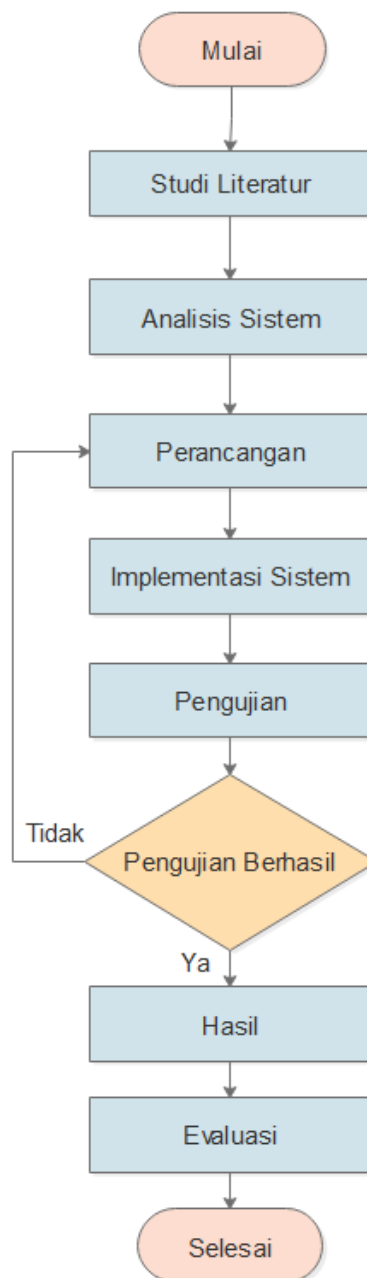
Table 3.2 Tabel Perangkat Lunak

No	Perangkat Lunak
1	<i>Sistem operasi Windows 10 64 bit</i>
2	<i>Arduino IDE</i>
3	<i>MySQL</i>
4	<i>PHP</i>
5	<i>Apache</i>
6	<i>Sublime</i>
7	<i>Xampp</i>

3.3 Metode Penelitian

Metode penelitian diperlukan agar penelitian dapat terstruktur sehingga hasil sesuai dengan tujuan penelitian, Adapun metode yang digunakan pada penelitian ini adalah Metode *Waterfall*, model ini berkembang secara sistematis dari satu tahap ke tahap selanjutnya.

Metode *waterfall* adalah sebuah pendekatan kepada pengembangan *software* yang sistematis dan sekuensial yang dimulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode program, pengujian dan pemeliharaan. Adapun metode pada penelitian ini dapat dilihat pada Gambar 3.1



Gambar 3.1 Metode Penelitian

3.3.1 Studi Literatur

Pada tahap ini dilakukan dengan mempelajari setiap teori yang akan diterapkan untuk mendukung penelitian. Teori-teori tersebut diperoleh dari buku, jurnal, internet dan karya ilmiah lainnya. Referensi yang akan dipelajari meliputi

cara membangun sebuah sistem *smart parking*, cara menggunakan dan memprogram mikrokontroller, dan prinsip kerja sensor ultrasonik.

3.3.2 Analisis Sistem

Analisis Sistem merupakan proses pengumpulan kebutuhan perangkat lunak dan perangkat keras. Untuk memahami dasar dari program yang akan dibuat, seorang analis harus mengetahui ruang lingkup informasi, fungsi-fungsi yang dibutuhkan, kemampuan kinerja yang ingin dihasilkan dan perancangan fungsi-fungsi utama dari sistem yang dapat dilakukan oleh aktor untuk memenuhi kebutuhannya terhadap sistem.

Aplikasi yang dibangun pada penelitian ini bernama *monitoring* perparkiran mobil menggunakan sensor ultrasonik PING berbasis *wireless*. Adapun kebutuhan fungsional penelitian ini terdapat pada tabel 3.3.

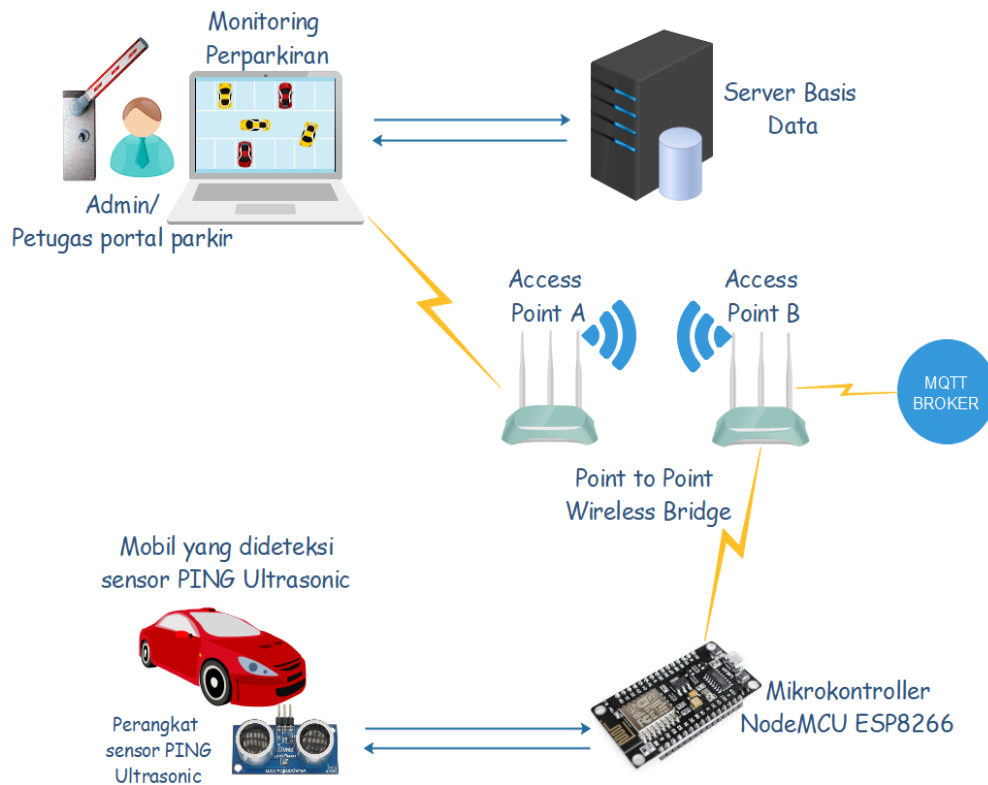
Tabel 3.3 Tabel Daftar Kebutuhan Fungsional

Kebutuhan	<i>Use case</i>	Jenis/Hak akses
Mampu mengetahui lot parkir yang kosong	Mengetahui lot parkir kosong	Admin
Mampu mengidentifikasi keberadaan mobil yang terparkir	Identifikasi lot parkir yang terisi	Sensor
Mampu mengidentifikasi mobil yang tidak terparkir	Identifikasi lot parkir yang kosong	Sensor

3.3.3 Tahap Perancangan

a. Arsitektur Sitem

Arsitektur sistem merupakan penggambaran umum untuk sistem yang akan dibuat. Pada Gambar 3.2 menunjukkan model perancangan arsitektur sistem *monitoring* perparkiran yang dibuat berdasarkan dari hasil analisa kebutuhan.

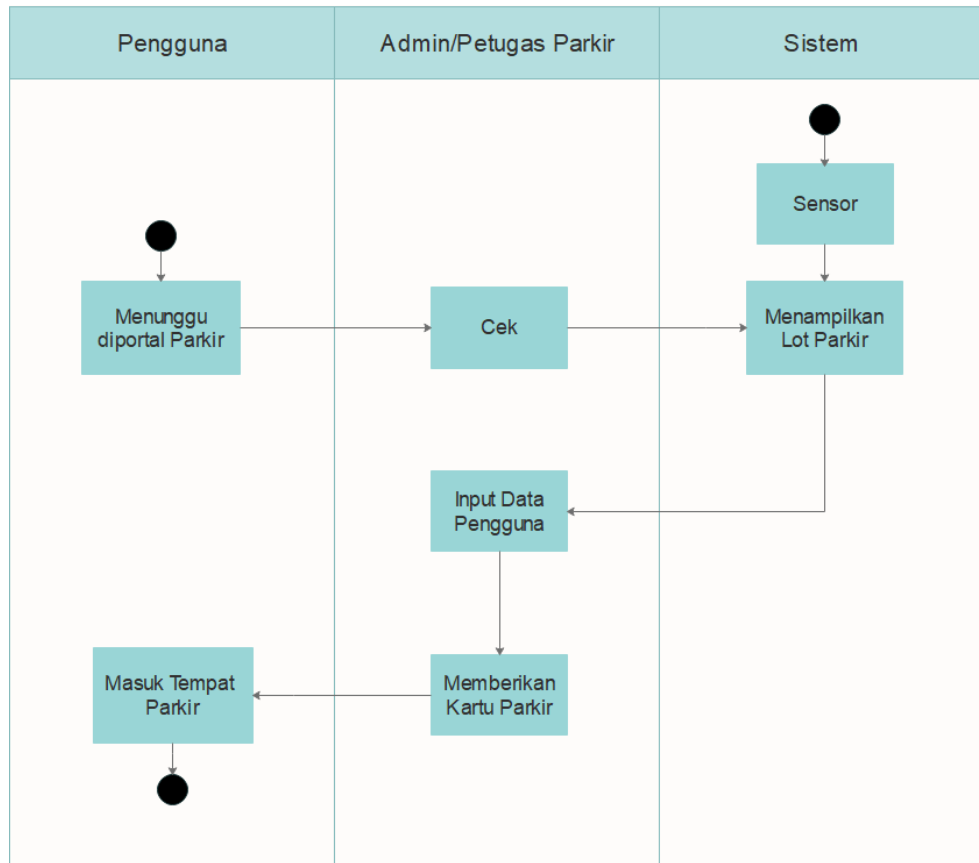


Gambar 3.2 Arsitektur sistem

Pada Gambar 3.2 menjelaskan bahwa setiap mobil yang parkir pada lot parkir akan dideteksi oleh sensor ultrasonik. Selanjutnya, sensor ultrasonik mengirim dan memproses data ke perangkat mikrokontroller Node MCU ESP8266, kemudian data-data yang diterima dari mikrokontroller akan dikirim ke komputer admin, dimana mikrokontroller tersebut terhubung ke *access point* B sedangkan komputer admin terhubung ke *access point* A, dan kedua *access point* tersebut telah dikonfigurasi *point to point wireless bridge* dan *client* antar kedua *access point* dapat berkomunikasi melalui MQTT Broker, sehingga admin/petugas portal parkir dapat melakukan *monitoring* parkir.

b. *Activity Diagram*

Activity Diagram merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.



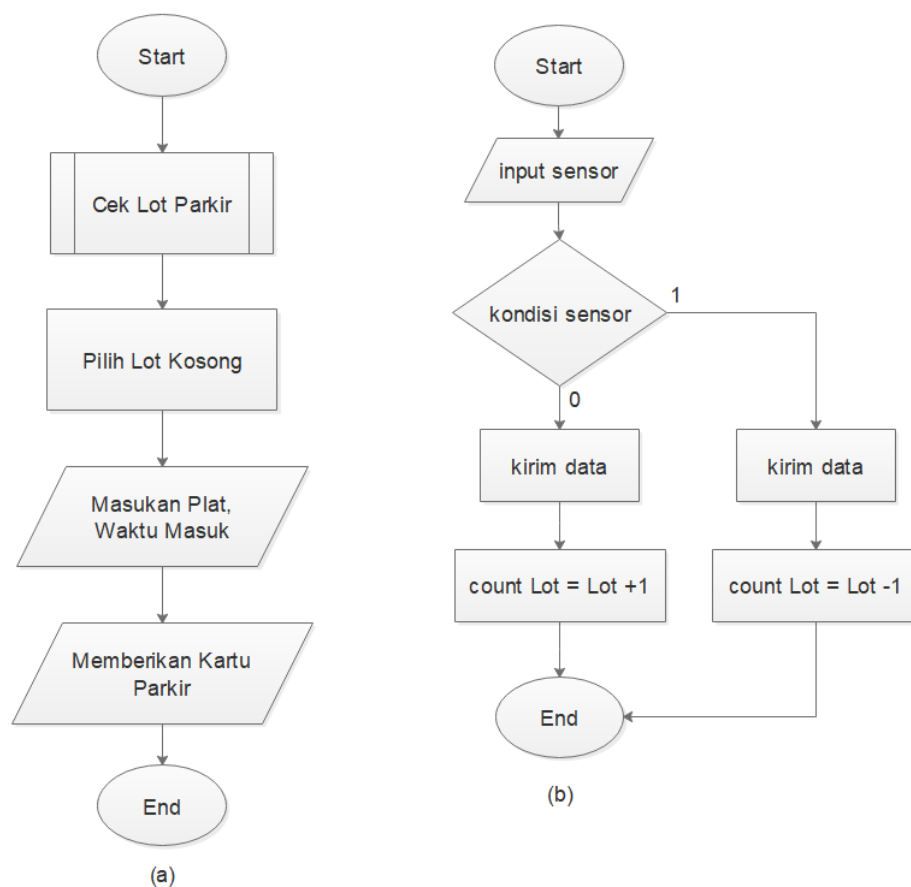
Gambar 3.3 *Activity Diagram*

Gambar 3.3 Menggambarkan *activity diagram* sistem pada penelitian ini. Dari sisi pengguna akan menunggu diportal parkir untuk mendapatkan kartu parkir, kemudian admin mengecek pada sistem untuk mengetahui lot parkir yang kosong berdasarkan inputan dari sensor, selanjutnya admin memasukkan nomor plat kendaraan pengguna lalu memberikan kartu parkir dan pengguna masuk ke tempat

parkir untuk memarkirkan kendaraanya di lot parkir sesuai pada kartu parkir yang diberikan oleh admin/petugas parkir.

c. *Flowchart*

Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses (intruksi) dengan proses lainnya dalam suatu program. *Flowchart* pada Sistem *Monitoring* Perparkiran terdapat pada Gambar 3.4



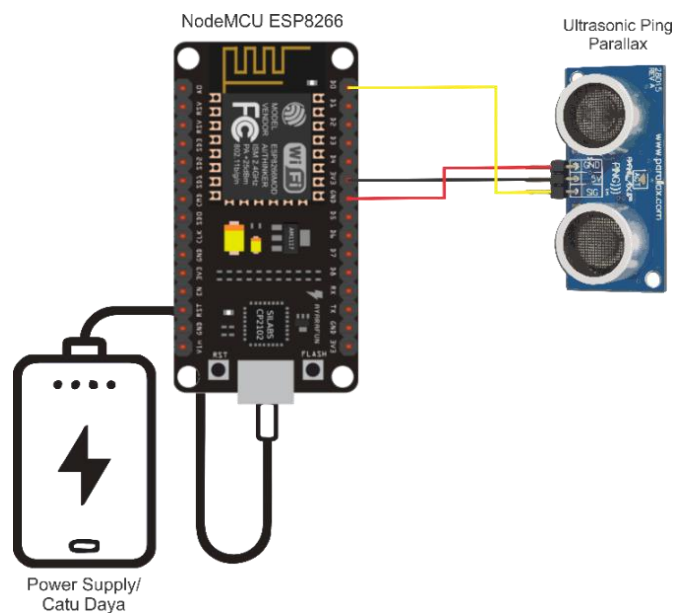
Gambar 3.4(a) *Flowchart* Sistem, (b) *Flowchart* Cek Lot Parkir

Gambar 3.4(a) merupakan gambaran dari alur sistem yang dibuat. Sistem melakukan pengecekan terhadap lot parkir yang kosong, kemudian admin parkir menentukan salah satu lot parkir yang kosong dan memasukkan nomor plat kendaraan pemilik kendaraan berupa nomor plat kendaraan & waktu masuk

kedalam sistem. Ketika pemasukan data selesai, selanjutnya admin memberikan kartu parkir kepada pemilik kendaraan.

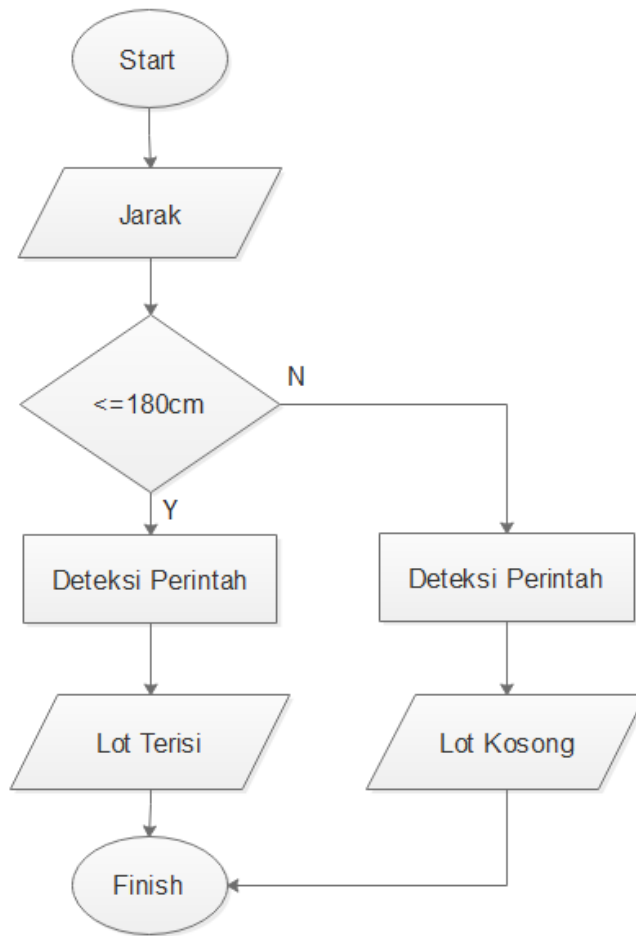
Proses cek lot parkir yang dapat dilihat pada Gambar 3.4b diawali dengan mengecek kondisi sensor, jika sensor bernilai 1 atau lot tersebut telah terisi dengan kendaraan maka data tersebut dikirim dari sensor ke sistem *monitoring* sehingga jumlah lot parkir yang tersedia berkurang sebanyak 1, sebaliknya jika sensor bernilai 0 atau lot tersebut tidak terisi oleh kendaraan maka data tersebut dikirim dari sensor ke sistem sehingga jumlah lot parkir yang tersedia bertambah 1.

d. Rangkaian Mikrokontroller Dan Sensor Ultrasonik



Gambar 3.5 Rangkaian Mikrokontroller Dan Sensor Ultrasonik

Rangkaian diatas terdiri dari *Power Supply*/Catu daya, NodeMCU ESP8266, dan satu buah sensor ultrasonik PING Parallax yang mendeteksi ada tidaknya kendaraan yang mengisi lot parkir.



Gambar 3.6 *Flowchart* Rangkaian Mikrokontroller dan Sensor Ultrasonik

Gambar 3.6 Merupakan proses pendeteksian mobil oleh sensor ultrasonik yang dimulai dari deteksi jarak, ketika jarak sensor kurang dari atau sama dengan 180 cm maka sensor mendeteksi perintah dan lot parkir tersebut telah terisi dengan kendaraan. Sebaliknya ketika jarak lebih dari 180 cm maka sensor mendeteksi perintah dan lot parkir tersebut tidak terisi dengan kendaraan.

3.3.4 Implementasi Sistem

Implementasi sistem terdiri dari implementasi sistem *monitoring* perparkiran, yang mengacu pada perancangan sistem yang dibuat. Untuk membuat

perintah pemograman yang akan dimasukan ke mikrokontroller menggunakan Arduino IDE.

3.3.5 Pengujian Sistem

Pengujian sistem disini bertujuan untuk memastikan apakah sistem bekerja dengan baik sesuai pengaturan dan perintah yang dibuat. Keberhasilan sistem yang dibuat ditentukan dari kesesuaian masing-masing *input* atau *output* yang diberikan. Pengujian sistem secara keseluruhan dilakukan dengan memberikan input yang sesuai dengan piranti input, untuk kemudian dilakukan pengujian kinerja sistem dalam menghasilkan *output* yang sesuai.

3.3.6 Hasil Penelitian

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian sistem telah selesai dilakukan. Pengambilan kesimpulan dapat dilakukan berdasarkan rumusan masalah terhadap hasil yang telah dicapai dari seluruh tahapan penelitian.

Hasil dari penelitian ini yaitu manajemen perparkiran dimana mobil/kendaraan yang akan parkir telah mendapatkan nomor lot parkir yang kosong yaitu berupa kartu parkir dari awal kedatangan dan admin akan memilih lot parkir kosong yang paling dekat.

3.3.7 Evaluasi

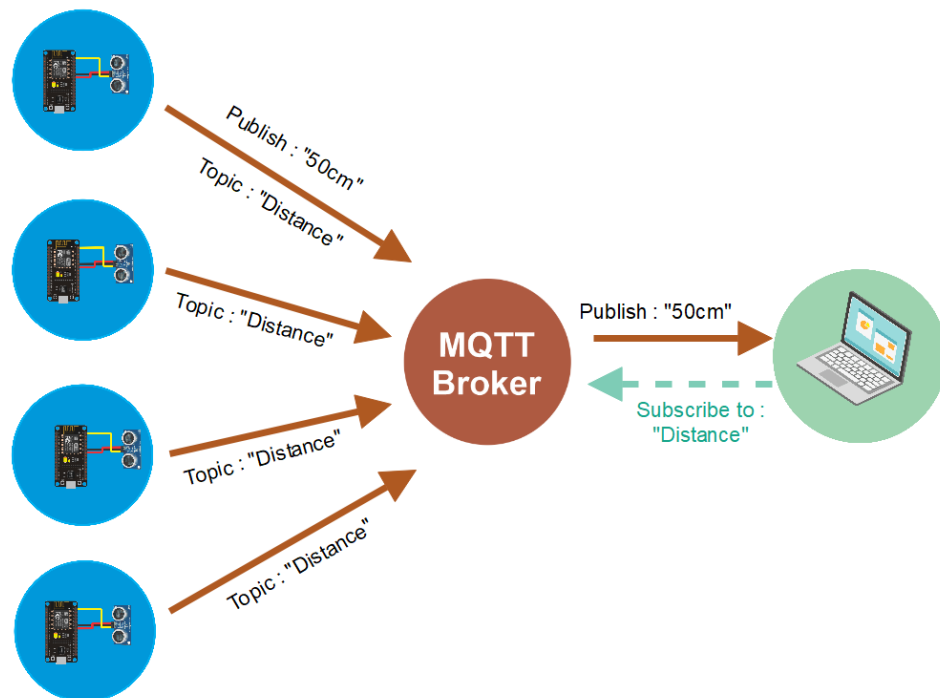
Pada tahap ini dilakukan perbaikan *error* yang terjadi pada saat proses pengujian, serta melakukan perbaikan jika hasil belum sesuai dengan kebutuhan

BAB IV HASIL DAN PEMBAHASAN

Hasil yang diperoleh dari penelitian ini yaitu *monitoring* perparkiran mobil menggunakan sensor ultrasonik berbasis *wireless*. Berdasarkan *flowchart* sistem pada Gambar 3.4a, sistem melakukan pengecekan terhadap lot parkir yang kosong, kemudian admin parkir memilih salah satu lot parkir yang kosong dan memasukkan nomor plat kendaraan pemilik kendaraan berupa nomor plat kendaraan & waktu masuk dalam sistem. Ketika pemasukan nomor plat kendaraan selesai, selanjutnya admin memberikan kartu parkir kepada pemilik kendaraan. Proses cek lot parkir yang dapat dilihat pada Gambar 3.4b diawali dengan mengecek kondisi sensor, jika sensor bernilai 1 atau lot tersebut telah terisi dengan kendaraan maka data tersebut dikirim dari sensor ke sistem *monitoring* perparkiran sehingga jumlah lot parkir yang tersedia berkurang sebanyak 1, sebaliknya jika sensor bernilai 0 atau lot tersebut tidak terisi oleh kendaraan maka data tersebut dikirim dari sensor ke sistem sehingga jumlah lot parkir yang tersedia bertambah 1.

Jarak deteksi sensor ultrasonik dengan mobil di area parkir yaitu 180 cm, ketika jarak sensor kurang dari atau sama dengan 180 cm maka sensor mendeteksi perintah dan lot parkir tersebut telah terisi dengan kendaraan. Sebaliknya ketika jarak lebih dari 180 cm maka sensor mendeteksi perintah dan lot parkir tersebut tidak terisi dengan kendaraan.

Sistem *monitoring* ini terintegrasi pada prototipe lot parkir melalui koneksi *wireless* dengan menggunakan metode *bridge* antar *access point* untuk memperluas sinyal, sehingga admin/pertugas portal parkir dapat melakukan *monitoring* perparkiran.



Gambar 4.1 Ilustrasi *Publish/Subscribe* dalam MQTT

Gambar 4.1 diatas menggambarkan Ilustrasi *Publisher* dan *Subscriber* dalam MQTT Protocol. Pada rancangan sistem tersebut terdiri dari Mikrokontroller ESP8266 dan sensor ultrasonik PING, kedua perangkat ini saling terhubung satu sama lain. Mikrokontroller ESP8266 terhubung dengan broker dan mengirimkan pesan berupa data jarak sensor ultrasonik ke broker. Kemudian broker meneruskan data dari mikrokontroller ke komputer/laptop *monitoring*. Pada komputer tersebut telah terhubung ke broker sebagai *subscriber*. Sehingga komputer tersebut dapat berfungsi untuk *monitoring* perparkiran.

Mikrokontroller ESP8266 mem-*publish* sebuah pesan dengan jarak sensor 50cm dan “*Distance*” sebagai *topic*. Broker menerima pesan yang di-*publish* oleh *publisher*, dan meneruskan *topic* “*Dinstance*” ke Laptop/computer *monitoring*.

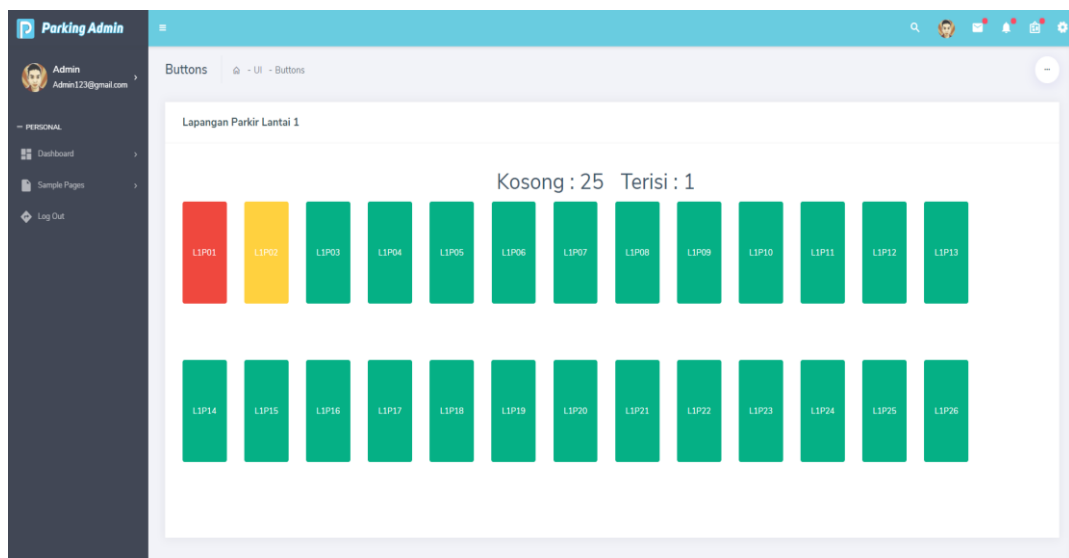
Berdasarkan perancangan tersebut, maka telah dilakukan implementasi

serta pengujian sistem yang meliputi pengujian fungsionalitas dan pengujian kinerja.

4.1 Implementasi

Implementasi pada *monitoring* perparkiran mobil menggunakan sensor ultrasonik berbasis *wireless* ini merupakan tampilan *monitoring* admin dan prototipe lot parkir, mulai dari pengecekan lot parkir yang kosong, pemasukan data pengguna, hingga *monitoring* perparkiran oleh admin.

4.1.1 Halaman Utama



Gambar 4.2 Tampilan Halaman Utama

Pada halaman utama admin *me-monitoring* dan pengecekan lot parkir pada perparkiran. Gambar 4.2 merupakan halaman utama yang menampilkan jumlah lot parkir yang kosong maupun lot parkir yang sedang terisi oleh kendaraan. Warna pada lot parkir terbagi menjadi 3 warna yaitu hijau, merah, dan kuning.



Gambar 4.3 Warna lot parkir

Warna pada lot parkir terbagi menjadi 3 warna yaitu hijau, kuning, dan merah. Pada saat kondisi lot parkir berwarna hijau berarti lot tersebut sedang kosong. Pada saat kondisi lot parkir berwarna kuning berarti pemilik kendaraan telah menerima kartu parkir dari admin dan segera menuju ke lot parkir, sehingga admin mengetahui lot parkir mana yang kartu parkirnya telah tercetak namun kendaraan belum terdeteksi oleh sensor, sedangkan jika kondisi lot parkir berwarna merah maka lot parkir tersebut telah terisi oleh kendaraan.

4.1.2 Halaman Pemasukan Plat Nomor Kendaraan

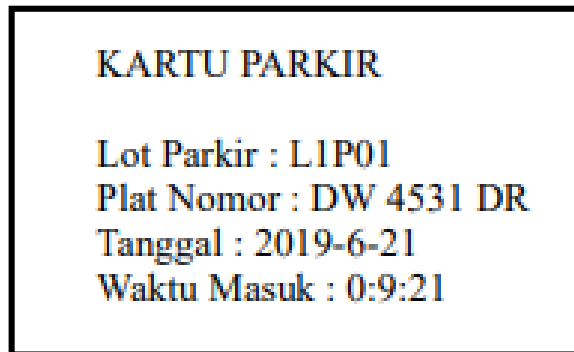
A screenshot of a web application window titled 'Lot Parking : L1P01'. Inside the window, there is a text input field labeled 'Plat Nomor' containing the text 'DT 1303 ME'. Below the input field, there are two buttons: a light gray button labeled 'CLOSE' on the left and a purple button labeled 'PRINT' on the right.

Gambar 4.4 Tampilan Pemasukan Plat Nomor Kendaraan

Halaman pemasukan plat nomor kendaraan berfungsi untuk memasukkan nomor plat kendaraan dan mencetak kartu parkir. Pada Gambar 4.4 dapat dilihat ketika admin memilih lot yang kosong sistem meminta input plat nomor kendaraan

dari pengguna perparkiran.

4.1.3 Tampilan Cetak Kartu Parkir



Gambar 4.5 Tampilan Cetak Kartu Parkir

Kartu parkir di berikan oleh admin parkir ke pengguna setelah admin mencetak kartu parkir. Pada Gambar 4.5 terlihat bahwa pada kartu parkir terdapat informasi berupa lot parkir, plat nomor, tanggal masuk, dan waktu masuk.

4.1.4 Prototipe Lot Parkir



Gambar 4.6 Prototipe Lot Parkir

Prototipe lot parkir terdiri atas sebuah Mikrokontroller ESP8266, *Powerbank* sebagai penghantar daya yang terhubung ke ESP8266, serta sensor

ultrasonik yang mendeteksi kendaraan pada lot perparkiran. Jika sensor mendeteksi kendaraan di lot parkir maka warna lot parkir pada halaman utama yang dapat dilihat pada Gambar 4.2 berwarna merah yang artinya lot terisi, sedangkan jika sensor tidak mendeteksi kendaraan maka warna pada lot parkir pada halaman utama yang dapat dilihat pada Gambar 4.2 berwarna hijau.

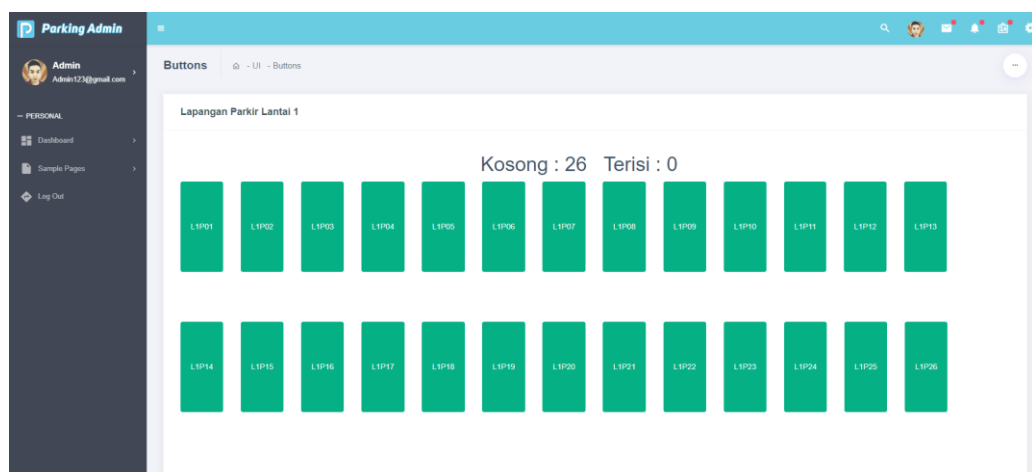
4.2 Pengujian

4.2.1 Pengujian Fungsionalitas

Pengujian fungsionalitas meliputi pengujian *monitoring* perparkiran, pengelolaan database, melihat data kendaraan masuk, serta pengujian prototipe lot parkir.

a. Pengujian Fungsionalitas *Monitoring* Perparkiran

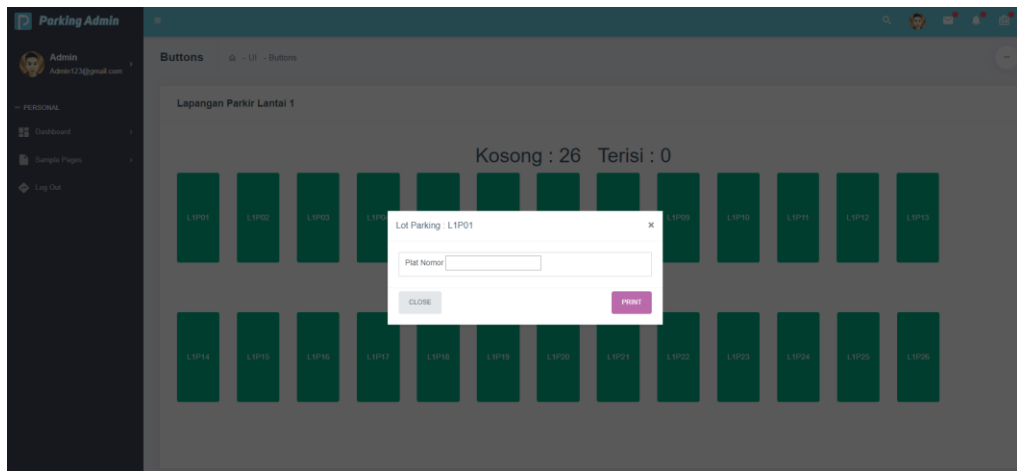
Pengujian fungsionalitas *monitoring* perparkiran dilakukan dengan melihat jumlah lot parkir yang terisi maupun yang kosong, dan memilih salah satu lot parkir yang kosong dan memasukkan nomor plat kendaraan pengguna.



Gambar 4.7 Tampilan *Monitoring* Perparkiran (lot berwarna hijau)

Pada Gambar 4.7 dapat dilihat tampilan jumlah lot parkir yang kosong

maupun lot parkir yang sedang terisi oleh kendaraan.

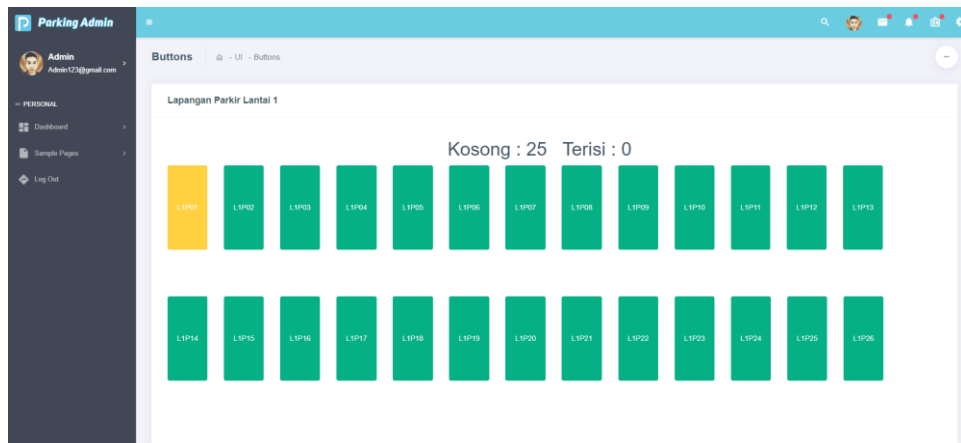


Gambar 4.8 Pemasukan Plat Nomor Kendaraan

Pada Gambar 4.8 menunjukkan bahwa admin telah memilih lot yang kosong dan memasukkan nomor plat kendaraan yang telah menunggu di portal parkir. Pada tabel_sensor kolom terisi_kosong di *database* telah *ter-update* dengan angka “2” berarti lot parkir L1P01 akan terisi mobil yang telah mendapatkan kartu parkir dapat dilihat pada Gambar 4.9.

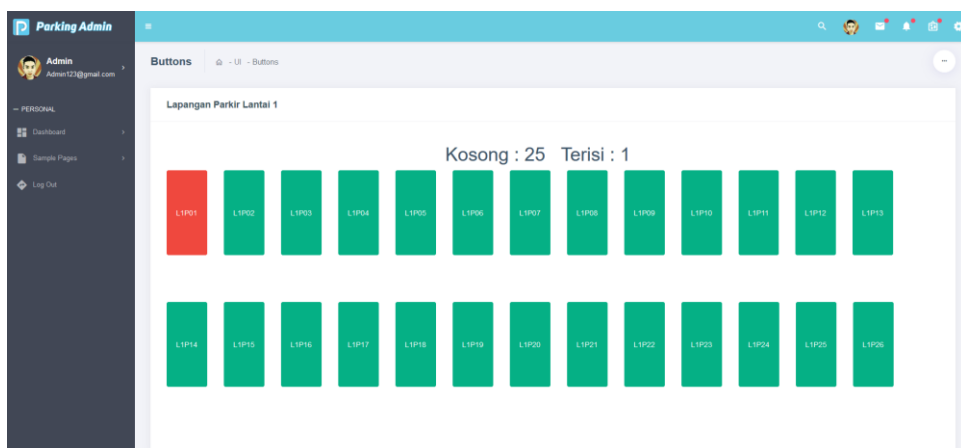
id	1	terisi_kosong	sensor_nama
1	2	L1P01	
2	0	L1P02	
3	0	L1P03	

Gambar 4.9 terisi_kosong *ter-update*



Gambar 4.10 Tampilan *Monitoring* Perparkiran (lot berwarna kuning)

Ketika admin telah mencetak kartu parkir pengguna pada lot parkir L1P01 maka warna pada lot parkir L1P01 berubah dari hijau menjadi kuning dan lot parkir yang kosong secara otomatis berkurang 1, yang sebelumnya jumlah lot parkir sebanyak 26 lot parkir menjadi 25 lot parkir yang kosong. Dapat dilihat pada Gambar 4.10.



Gambar 4.11 Tampilan *Monitoring* Perparkiran (lot berwarna merah)

Pada Gambar 4.11 menunjukkan bahwa warna pada lot parkir L1P01 berubah dari kuning menjadi hijau dan pada tabel_sensor kolom terisi_kosong di *database* telah ter-*update* dengan angka “1”, berarti lot parkir L1P01 telah terisi kendaraan yang dapat dilihat pada Gambar 4.12

id	terisi_kosong	sensor_nama
1	1	L1P01
2	0	L1P02
3	0	L1P03
...
25	0	L1P25
26	0	L1P26

Gambar 4.12 terisi_kosong ter-update

b. Pengujian Prototipe Lot Parkir

Pengujian prototipe lot parkir dilakukan untuk menguji keberhasilan prototipe mengirim dan menerima data dari sensor ke aplikasi *monitoring* perparkiran. Prototipe lot parkir di pasang di tiap-tiap lot perparkiran yang dapat di lihat pada Gambar 4.13.



Gambar 4.13 Prototipe Lot Parkir



Gambar 4.14 Pemberian Kartu Parkir

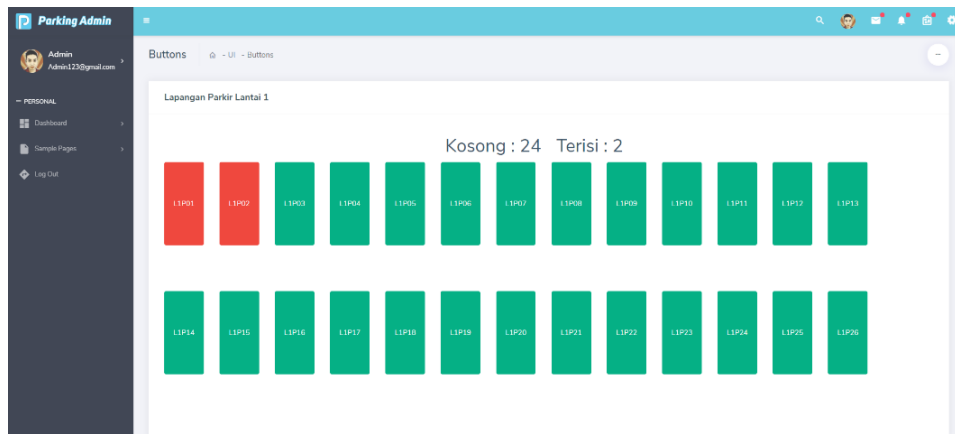
Pada Gambar 4.14 admin memberikan kartu parkir ke pemilik kendaraan dimana admin memilih lot parkir yang terdekat yaitu L1P01 dan memasukkan nomor plat kendaraan pengguna dan lot parkir pada tampilan *monitoring* perparkiran yang dapat dilihat pada Gambar 4.10 berubah menjadi warna kuning.



Gambar 4.15 Kendaraan yang Terparkir

Pada Gambar 4.15 terdapat kendaraan yang terparkir, kendaraan dengan nomor plat DT 1303 ME berada pada lot parkir L1P01, sedangkan kendaraan dengan nomor plat DN 611 AW memarkirkan kendaraanya pada lot parkir L1P02.

Pada tampilan *monitoring* perparkiran lot L1P01 dan lot L1P02 berwarna merah yang berarti lot terisi dapat dilihat pada Gambar 4.16.



Gambar 4.16 Tampilan *Monitoring* Perparkiran

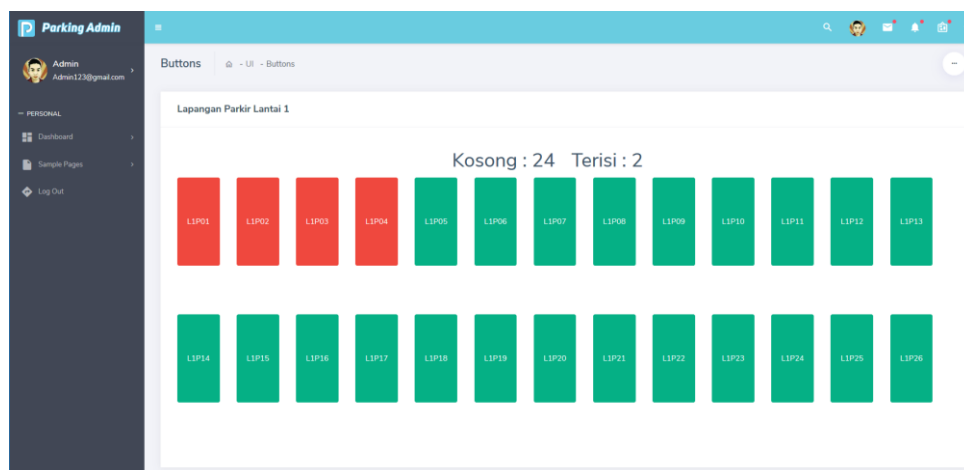
id	terisi_kosong	sensor_nama
1	1	L1P01
2	1	L1P02
3	0	L1P03
4	0	L1P04
...
25	0	L1P25
26	0	L1P26

Gambar 4.17 terisi_kosong ter-update

Ketika sensor mendeteksi kendaraan pada lot parkir maka pada tabel_sensor kolom terisi_kosong di *database* ter-update dengan angka “1” yang dapat dilihat pada Gambar 4.17 berarti lot parkir L1P01 dan L1P02 telah terisi kendaraan maka informasi pengguna perparkiran masuk ke *database* sistem seperti yang tampil pada Gambar 4.18.

id	lot	plat	datetime
1	L1P01	DD 1303 ME	2019-05-19 14:35:32
2	L1P02	DN 611 AW	2019-05-19 14:36:55

Gambar 4.18 tabel_log ter-update



Gambar 4.19 Tampilan *monitoring* perparkiran

Pada Gambar 4.19 tampilan *monitoring* perparkiran pada lot L1P01 lot L1P02, L1P03 dan L1P04 berwarna merah yang berarti lot terisi.

id	terisi_kosong	sensor_nama
1	1	L1P01
2	1	L1P02
3	1	L1P03
4	1	L1P04
...
25	0	L1P25
26	0	L1P26

Gambar 4.20 terisi_kosong ter-update

Pada tabel_sensor kolom terisi_kosong di *database* ter-update dengan

angka “1”. Pada Gambar 4.20 berarti lot parkir L1P01, L1P02, L1P03, dan L1P04 telah terisi kendaraan maka informasi pengguna perparkiran masuk ke *database* sistem seperti yang tampil pada Gambar 4.21.

id	lot	plat	datetime
1	L1P01	DD 1303 ME	2019-05-19 14:35:32
2	L1P02	DN 611 AW	2019-05-19 14:36:55
3	L1P03	DP 3691 ER	2019-05-19 15:10:32
4	L1P04	DW 4531 DR	2019-05-19 15:12:17

Gambar 4.21 tabel_log ter-update

c. Pengujian Fungsionalitas Lot Parkir

Jika dilihat secara keseluruhan pengujian fungsionalitas maka menghasilkan data pada tabel 4.1.

Tabel 4.1 Pengujian Fungsionalitas Lot Parkir

Skenario Pengujian	Lot parkir berwarna Hijau	Lot parkir berwarna kuning	Lot parkir berwarna merah	Ket.
Pengguna menunggu di portal	Ya	Tidak	Tidak	Berhasil
Pengguna Mendapatkan kartu parkir	Tidak	Ya	Tidak	Berhasil
Pengguna Memarkirkan kendaraan di lot	Tidak	Tidak	Ya	Berhasil
Pengguna meninggalkan lot parkir	Ya	Tidak	Tidak	Berhasil

Dari hasil pengujian *Monitoring* Lot Parkir pada tabel 4.1 diujikan dengan empat kondisi pengguna yang akan memarkirkan kendaraanya. Pengujian yang

pertama adalah pengguna yang menunggu di portal parkir dan lot parkir masih berwarna hijau. Pengujian kedua pengguna mendapatkan kartu parkir dan lot parkir berubah menjadi warna kuning. Pengujian yang ketiga pengguna memarkir kendaraan di lot parkir dan terdeteksi oleh sensor dan lot parkir berubah menjadi warna merah. Kondisi pengujian yang keempat pengguna meninggalkan lot parkir dan sensor tidak mendeteksi adanya kendaraan dan lot parkir berubah menjadi warna hijau.

4.2.2 Pengujian Kinerja

Pengujian kinerja dilakukan untuk menguji sistem *monitoring* perparkiran menggunakan sensor ultrasonik PING berbasis *wireless* yang meliputi pengujian waktu respon perubahan warna lot, pengujian kinerja waktu kendaraan ke lot parkir dan pengujian kinerja keakuratan sensor ultrasonik PING dan ultrasonik HC-SR04.

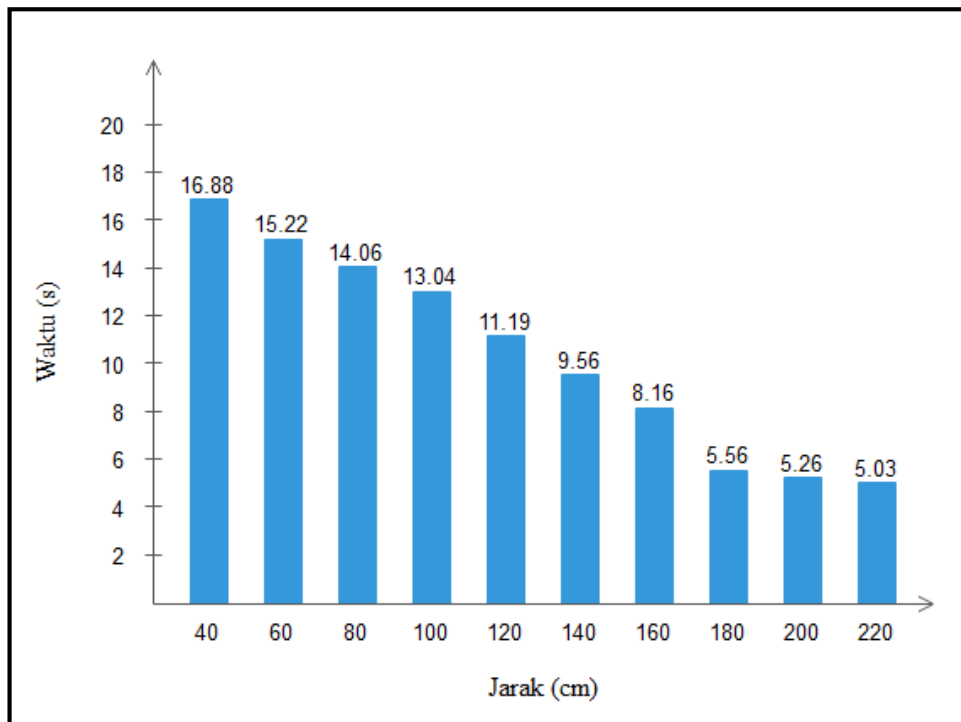
a. Pengujian Waktu Respon Perubahan Warna Lot

Pengujian ini dilakukan untuk menguji kecepatan respon sensor dan mengirim data ke *database*, pengujian dilakukan sebanyak 10 kali dengan jarak sensor yang berbeda-beda dan jarak mundur mobil ke lot parkir sama dengan 4 cm. Pengujian waktu dimulai pada saat mobil mundur ke lot parkir. Adapun hasil pengujian tersebut dapat dilihat pada tabel 4.2.

Tabel 4.2 Pengujian Waktu Respon Perubahan Warna Lot(detik)

No.Pengujian	Jarak Sensor(cm)	Kecepatan Waktu Respon(s)
1	40	16.88
2	60	15.20
3	80	14.06
4	100	13.04
5	120	11.19
6	140	09.56
7	160	08.16
8	180	05.56
9	200	05.26
10	220	05.03

Dari hasil pengujian waktu respon perubahan warna lot pada sensor, digambarkan dalam bentuk grafik seperti pada Gambar 4.22.



Gambar 4.22 Grafik Waktu Respon Perubahan Warna Lot

Dari grafik pada Gambar 4.22 menunjukkan bahwa jarak sensor terhadap kendaraan berpengaruh pada waktu respon. Dari hasil pengujian pada Gambar 4.22 juga terlihat bahwa semakin jauh jarak deteksi sensor maka semakin cepat waktu perubahan warna lot pada tampilan *monitoring* perparkiran. Dari hasil pengujian tersebut dapat dikatakan bahwa jarak sensor dengan 180 cm dengan kecepatan waktu respon 5.56 detik, 200 cm dengan kecepatan waktu respon 5.26 detik, dan 220 cm dengan kecepatan waktu respon 5.03 detik. Jika ketiga hasil tersebut di rata-ratakan maka di peroleh angka 5.28 detik waktu respon, nilai waktu respon tersebut merupakan nilai terbaik sensor ultrasonik untuk mendeteksi kendaraan.

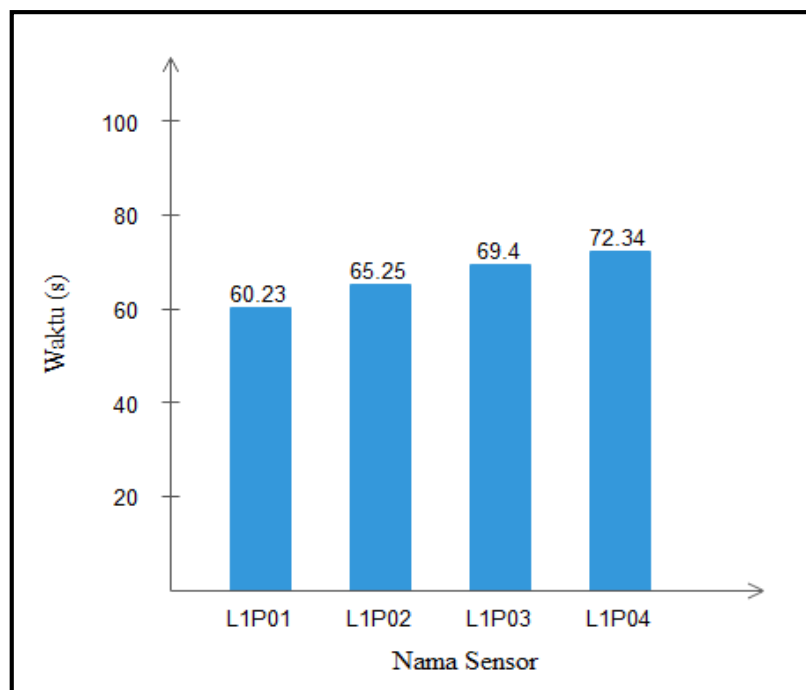
b. Pengujian kinerja waktu kendaraan ke lot parkir

Pengujian ini dilakukan untuk menguji waktu yang di butuhkan pengguna saat perjalanan dari portal parkir menuju ke lot parkir yang berbeda dengan kecepatan sama dengan 15 km/jam.

Tabel 4.3 Pengujian Kinerja Waktu Kendaraan ke Lot Parkir

No.Pengujian	Kecepatan Mobil(km/jam)	Nama lot Parkir	Waktu (s)
1	15	L1P01	60.23
2	15	L1P02	65.25
3	15	L1P03	69.40
4	15	L1P04	72.34

Dari hasil pengujian kinerja waktu kendaraan ke lot parkir, digambarkan dalam bentuk grafik seperti pada Gambar 4.22.

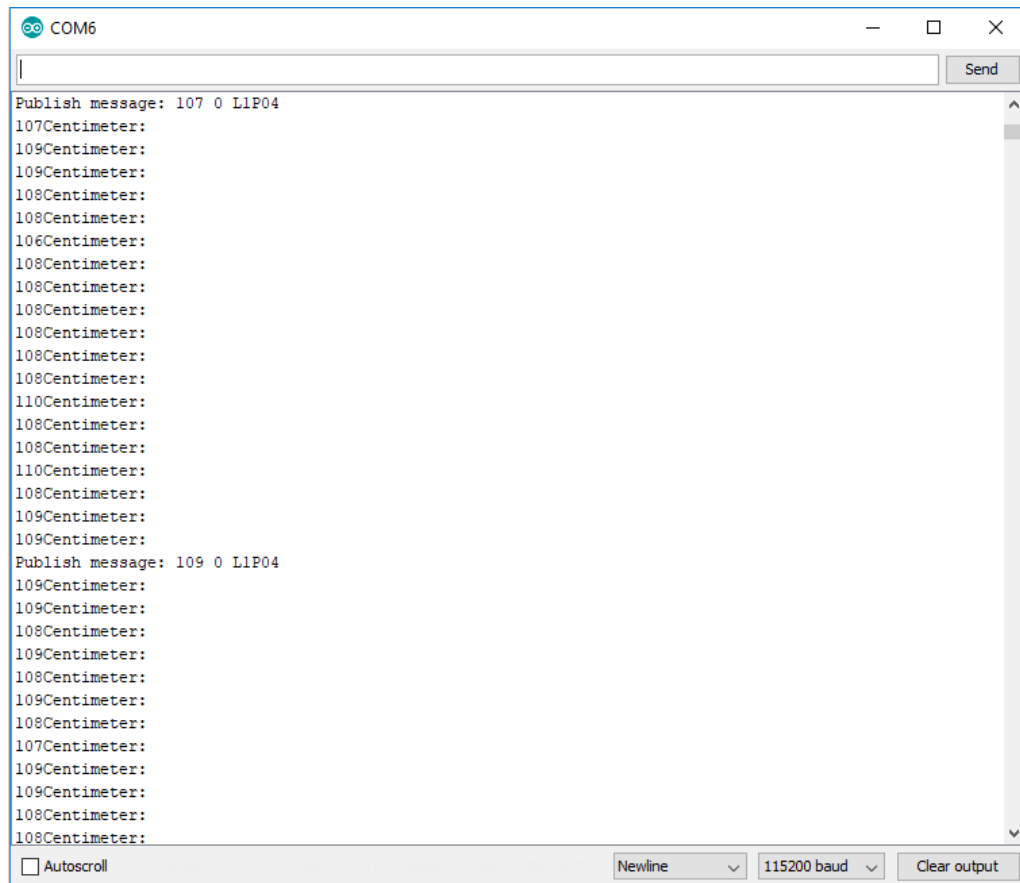


Gambar 4.23 Grafik Kinerja Waktu Kendaraan ke Lot Parkir

Pengujian ini dilakukan untuk menguji waktu yang di butuhkan pengguna saat perjalanan dari portal parkir menuju ke lot parkir. Dari hasil pengujian kinerja waktu kendaraan ke lot parkir menunjukkan bahwa semakin dekat lot parkir yang di dapatkan maka semakin cepat waktu yang di butuhkan pengguna ke lot parkir.

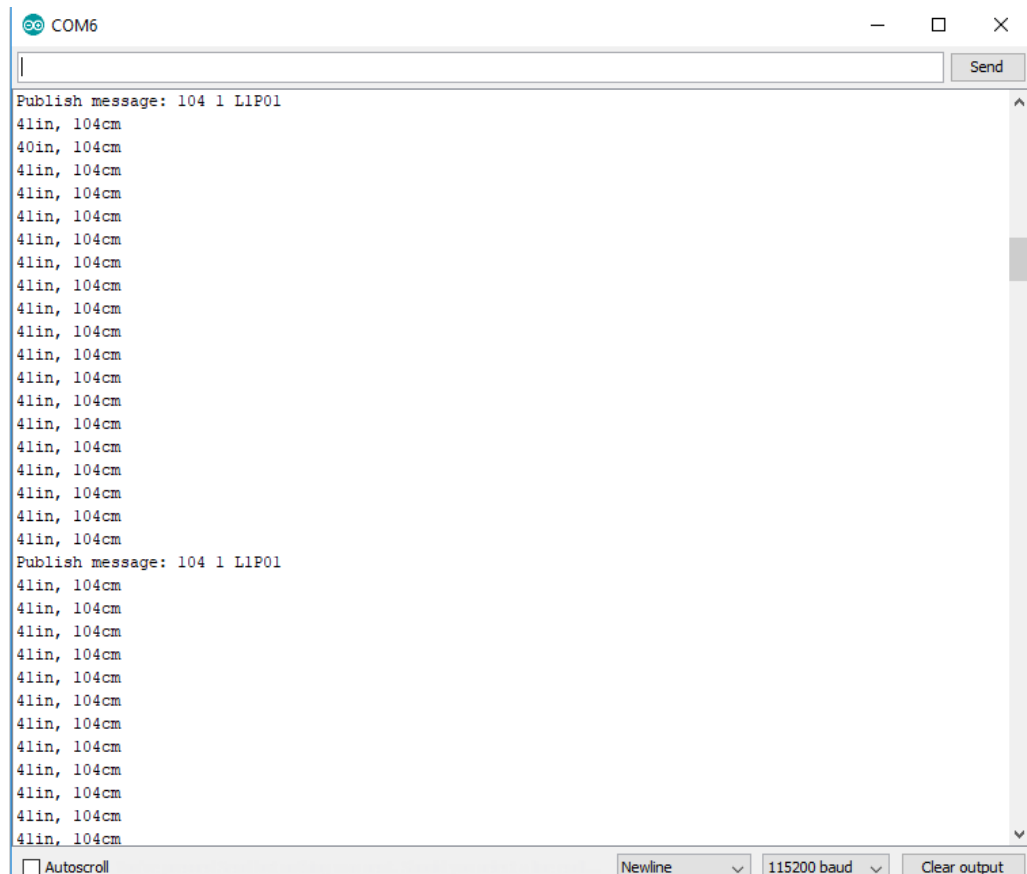
c. Pengujian keakuratan sensor ultrasonik PING dan ultrasonik HC-SR04

Pengujian ini dilakukan untuk menguji keakuratan kedua sensor ultrasonik dalam mendeteksi sebuah objek. Pengujian pada objek tersebut dalam kondisi tidak bergerak.



Gambar 4.24 Pengujian Sensor Ultrasonik HC-SR04

Dari hasil pengujian diatas menunjukkan bahwa sensor ultrasonik HC-SR04 mendeteksi sebuah objek dengan nilai jarak sensor yang berubah-ubah yaitu 106 cm sampai 110 cm.



Gambar 4.25 Pengujian Sensor Ultrasonik PING

Dari hasil pengujian diatas menunjukkan bahwa sensor ultrasonik PING mendeteksi sebuah objek dengan nilai jarak sensor tidak berubah yaitu 104 cm. Dari hasil pengujian kedua sensor tersebut dapat disimpulkan bahwa sensor ultrasonik PING lebih akurat dibandingkan sensor ultrasonik HC-SR04

BAB V PENUTUP

5.1 Kesimpulan

Dari proses perancangan, implementasi, serta pengujian sistem dapat ditarik kesimpulan sebagai berikut.

1. Sistem monitoring yang dibangun dapat melakukan monitoring perparkiran dan mengetahui jumlah lot parkir pada area parkir yang kosong maupun lot parkir yang sedang terisi oleh kendaraan
2. Sistem monitoring yang telah dibangun mampu memberikan informasi terhadap ketersediaan lot parkir yang kosong dan terintegrasi database untuk menyimpan informasi pemilik kendaraan berdasarkan kartu parkir. Informasi pada kartu parkir yaitu lot parkir, plat nomor, tanggal masuk, dan waktu masuk.
3. Hasil pengujian menunjukkan semakin jauh jarak deteksi sensor maka semakin cepat waktu respon perubahan warna lot pada tampilan monitoring perparkiran. Sensor dengan jarak 180 cm dengan kecepatan waktu respon 5.56 detik merupakan nilai terbaik sensor ultrasonik PING untuk mendeteksi kendaraan.

5.2 Saran

Untuk pengembangan selanjutnya dapat dilakukan pengembangan aplikasi yang dapat pendeteksian pada pengendara yang salah memarkirkan kendaraannya pada lot parkir. Sistem *monitoring* ini memiliki fitur-fitur yang belum sempurna. Oleh karena itu, hal ini dapat menjadi acuan untuk dikembangkan kedepannya agar lebih bermanfaat.

DAFTAR PUSTAKA

- Arief, U. M. (2011). Pengujian Sensor Ultrasonik PING untuk Pengukuran Level Ketinggian dan Volume Air. *Jurnal Ilmiah "Elektrikal Enjiniring" UNHAS*, 09(02), 72–77.
- Badan Pusat Statistik. (2016). *Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis 1949-2017*. Tersedia pada: <https://www.bps.go.id/linkTableDinamis/view/id/1133> (Diakses: 6 November 2018)
- Budiarso, Z., & Prihandono, A. (2015). Implementasi Sensor Ultrasonik Untuk Mengukur Panjang Gelombang Suara Berbasis Mikrokontroler. *Jurnal Teknologi Informasi DINAMIK*, 20(2), 171–177.
- Connolly, T. M., & Begg, C. E. (2010). Database Systems : A Practical Approach to design, Implementation, and Managemnet(5th ed.). *Boston: Pearson*.
- Detiknews. (2013). *Warga Keluhkan Sulitnya Cari Parkiran Mobil di Mal-Mal Jakarta*. Tersedia pada: <https://news.detik.com/berita/2358138/warga-keluhkan-sulitnya-cari-parkiran-mobil-di-mal-mal-jakarta> (Diakses: 6 November 2018).
- EINSTRONIC. (2017). *INTRODUCTION to NodeMCU ESP8266*. Tersedia pada: <https://einstronic.com/wp-content/uploads/2017/06/NodeMCU-ESP8266-ESP-12E-Catalogue.pdf> (Diakses: 21 Mei 2019).
- Hakim, M. L. (2016). *Model Smart Parking Berbasis Internet of Things*. Bogor: Perpustakaan FMIPA UNPAK.
- Limantara, A. D., Cahyo Setianto Purnomo, Y., & Mudjanarko, S. W. (2017). *Pemodelan Sistem Pelacakan Lot Parkir Kosong Berbasis Sensor Ultrasonic Dan Internet Of Things (IoT) Pada Lahan Parkir Diluar Jalan*. Makalah yang disajikan dalam Seminar Nasional Sains dan Teknologi. Fakultas Teknik Universitas Muhammadiyah Jakarta. Jakarta: 1 November 2017.
- Lumba, A. Y. (2017). *Sistem Kontrol Pengisian Air Pada Tandon Menggunakan SMS Gateway Berbasis Arduino*. Skripsi. Kendari: Universitas Halu Oleo.
- Mardiani, G. T. (2013). Sistem Monitoring Data Aset dan Inventaris PT Telkom Cianjur Berbasis Web. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 2(1), 35–40.

- Martadipura, J. (2013). *Perancangan Sistem Infomasi Perparkiran pada Universitas Komputer Indonesia*. Artikel. Bandung: Universitas Komputer Indonesia.
- MQTT. (2019). *MQTT Frequently Asked Questions*. Tersedia pada: <https://mqtt.org/faq> (Diakses: 21 Mei 2019).
- Permana, A., Triyanto, D., & Rismawan, T. (2015). Rancang Bangun Sistem Monitoring Volume dan Pengisian Air Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler AVR ATMEGA8. *Jurnal Coding, Sistem Komputer Untan*, 03(2), 76–87.
- Ppatierno. (2013). *Message Flow*. Tersedia pada: <http://www.embedded101.com/Develop-M2M-IoT-Devices-Ebook/DevelopM2MIoTDevicesContent/smid/946/ArticleID/224> (Diakses: 21 Mei 2019)
- Putra, R. F. H., Lhaksmana, K. M., & Adytia, D. (2018). Aplikasi IoT untuk Rumah Pintar dengan Fitur Prediksi Cuaca. *E-Proceeding of Engineering*, 5(1), 1746–1760.
- Rudi, Dinata, I., & Kurniawan, R. (2017). Rancang Bangun Protortype Sistem Smart Parking Berbasis Arduino dan Pemantauan Melalui Smartphone. *Jurnal Ecotipe*, 4(2), 14–20.
- Welling, Luke, & L. T. (2009). *PHP dan MySQL Web Development (4th edition)*. Addison Wesley Longman.

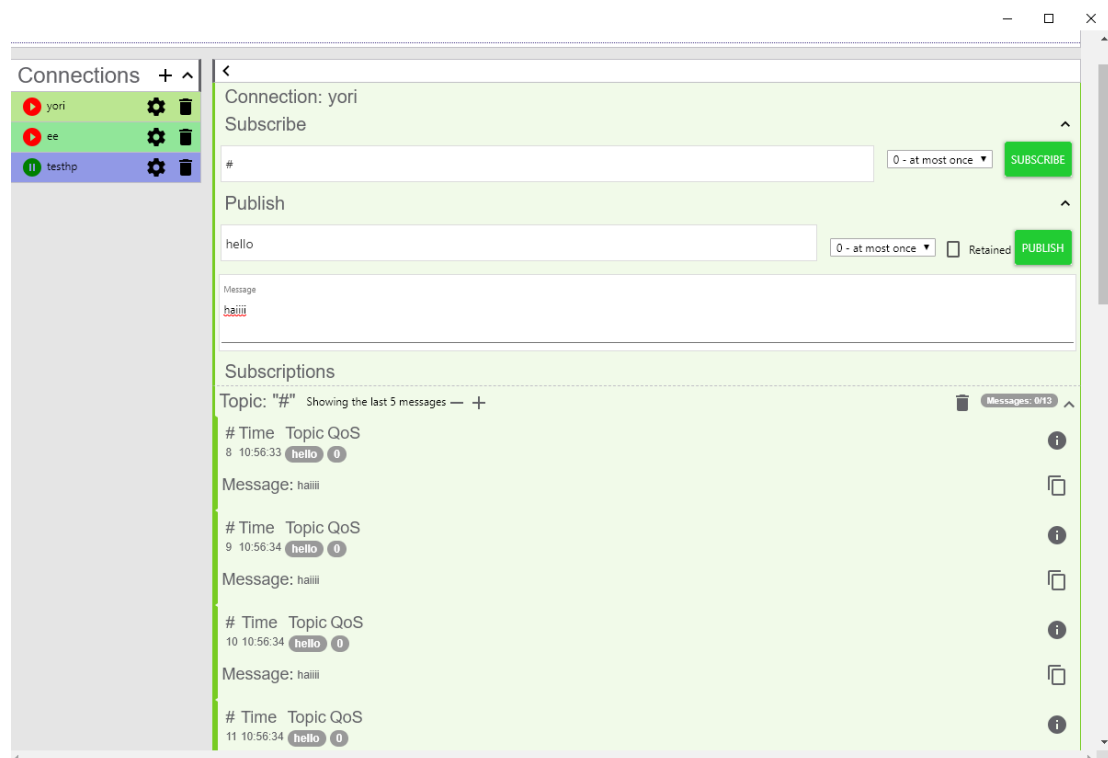
L
A
M
P
I
R
A
N

Lampiran 1. Pengujian jarak sensor ultrasonik PING

```
Administrator: Command Prompt
B23 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B25 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B24 0 L1P03
B25 0 L1P03
Traceback (most recent call last):
  File "teq.py", line 63, in <module>
    client.loop_forever()
  File "C:\Python27\lib\site-packages\paho\mqtt\client.py", line 1578, in loop_forever
    rc = self.loop(timeout, max_packets)
  File "C:\Python27\lib\site-packages\paho\mqtt\client.py", line 1057, in loop
    socklist = select.select(rlist, wlist, [], timeout)
KeyboardInterrupt

C:\Users\YORI\Desktop>python teq.py
```

Lampiran 2. Pengujian Pub/Sub menggunakan MQTT Lens



Lampiran 3. Source code Mikrokontroler ESP8266

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.
const int pingPin = 16;

const char* ssid = "TEST";
const char* password = "test1234";
const char* mqtt_server = "192.168.0.101";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup_wifi() {

  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

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

  randomSeed(micros());

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int
length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
```

```

    Serial.print((char)payload[i]);
}
Serial.println();

// Switch on the LED if an 1 was received as first
character
if ((char)payload[0] == '1') {
    digitalWrite(BUILTIN_LED, LOW);    // Turn the LED on
    (Note that LOW is the voltage level
    // but actually the LED is on; this is because
    // it is active low on the ESP-01)
} else {
    digitalWrite(BUILTIN_LED, HIGH);    // Turn the LED off by
    making the voltage HIGH
}

}

void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Create a random client ID
        String clientId = "ESP8266Client-";
        clientId += String(random(0xffff), HEX);
        // Attempt to connect
        if (client.connect(clientId.c_str())) {
            Serial.println("connected");
            // Once connected, publish an announcement...
            client.publish("outTopic", "hello world");
            // ... and resubscribe
            client.subscribe("inTopic");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

void setup() {
    pinMode(BUILTIN_LED, OUTPUT);    // Initialize the
    BUILTIN_LED pin as an output
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

```



```

void loop() {
    long duration, inches, cm, id;

    // The PING))) is triggered by a HIGH pulse of 2 or more
    microseconds.
    // Give a short LOW pulse beforehand to ensure a clean
    HIGH pulse:
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);

    // The same pin is used to read the signal from the
    PING))) : a HIGH pulse
    // whose duration is the time (in microseconds) from the
    sending of the ping
    // to the reception of its echo off of an object.
    pinMode(pingPin, INPUT);
    duration = pulseIn(pingPin, HIGH);

    // convert the time into a distance
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);

    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();

    delay(100);

    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    long now = millis();
    if (now - lastMsg > 2000) {
        lastMsg = now;
        ++value;
        if (cm > 180) {
            id = 0;
        }
        else {
            id = 1;
        }
    }
}

```

```

void loop() {
    long duration, inches, cm, id;

    // The PING))) is triggered by a HIGH pulse of 2 or more
    microseconds.
    // Give a short LOW pulse beforehand to ensure a clean
    HIGH pulse:
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);

    // The same pin is used to read the signal from the
    PING))) : a HIGH pulse
    // whose duration is the time (in microseconds) from the
    sending of the ping
    // to the reception of its echo off of an object.
    pinMode(pingPin, INPUT);
    duration = pulseIn(pingPin, HIGH);

    // convert the time into a distance
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);

    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();

    delay(100);

    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    long now = millis();
    if (now - lastMsg > 2000) {
        lastMsg = now;
        ++value;
        if (cm > 180) {
            id = 0;
        }
        else {
            id = 1;
        }
    }
}

```

```

        snprintf (msg, 50, "%ld %i L1P01", cm, id);
        Serial.print("Publish message: ");
        Serial.println(msg);
        client.publish("outTopic", msg);
    }
}

long microsecondsToInches(long microseconds) {
    // According to Parallax's datasheet for the PING))) ,
    there are 73.746
    // microseconds per inch (i.e. sound travels at 1130 feet
    per second).
    // This gives the distance travelled by the ping, outbound
    and return,
    // so we divide by 2 to get the distance of the obstacle.
    // See: http://www.parallax.com/dl/docs/prod/acc/28015-
    PING-v1.3.pdf
    return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
    // The speed of sound is 340 m/s or 29 microseconds per
    centimeter.
    // The ping travels out and back, so to find the distance
    of the object we
    // take half of the distance travelled.
    return microseconds / 29 / 2;
}

```

Lampiran 4. Source code update & insert database dan *connect* ke MQTT Broker

```
import paho.mqtt.client as mqtt
import mysql.connector

# This is the Subscriber

# set db configuration -----
dbhost = "localhost"
dbuser = "root"
dbpassword = ""
dbname = "parking"
#-----

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("#")

def on_message(client, userdata, msg):
    # if msg.payload.decode() == "Hello world!":
    print(msg.payload.decode())

    sensor_data = msg.payload.decode()
    data = sensor_data.split()

    if data[2] in sensor_nama:
        mydb =
mysql.connector.connect(host=dbhost,user=dbuser,passwd=dbpas
sword,database=dbname)
        mycursor = mydb.cursor()

        mycursor.execute("SELECT * FROM sensor")
        myresult = mycursor.fetchall()

        isi = myresult[0][1]

        if isi == 2:
            if data[1] == '1':
                sql = "UPDATE sensor SET terisi_kosong = %s WHERE
sensor_nama = %s"
                val = (data[1], data[2])
                mycursor.execute(sql,val)
                mydb.commit()
```

```

        else:
            sql = "UPDATE sensor SET terisi_kosong = %s WHERE
sensor_nama = %s"
            val = (data[1], data[2])
            mycursor.execute(sql, val)
            mydb.commit()
        else :
            #print('data tdak ada')
            sensor_nama.append(data[2])
            sql = "INSERT INTO sensor (terisi_kosong, sensor_nama)
VALUES (%s, %s)"
            # value
            val = (data[1], data[2])

            mycursor.execute(sql, val)
            mydb.commit()

    # client.disconnect()

sensor_nama = ['s0']
# mysqlconnect
mydb =
mysql.connector.connect(host=dbhost,user=dbuser,passwd=dbpas
sword,database=dbname)
mycursor = mydb.cursor()

mycursor.execute("SELECT * FROM sensor")
myresult = mycursor.fetchall()
isi = []
for x in myresult:
    sensor_nama.append(x[2])
    isi.append(x[1])

perubahan = False
client = mqtt.Client()
client.connect("192.168.0.101",1883,60)

client.on_connect = on_connect
client.on_message = on_message

client.loop_forever()

```

Lampiran 5. Konfigurasi Access Point Bridge

Wireless Settings

Wireless Network Name:

TEST1

(Also called the SSID)

Region:

United States

Warning:

Ensure you select a correct country to conform local law.
Incorrect settings may cause interference.

Mode:

11bgn mixed

Channel Width:

Auto

Channel:

6

Max Tx Rate:

300Mbps

☒ Enable Wireless Router Radio

☒ Enable SSID Broadcast

☒ Enable WDS Bridging

SSID (to be bridged):

TEST

BSSID (to be bridged):

A0-F3-C1-A5-7F-CC

Example:00-1D-0F-11-22-33

Survey

Key type:

WPA-PSK/WPA2-PSK

WEP Index:

1

Auth type:

open

Password:

test1234

Save