



**RANCANG BANGUN REMOTE SISTEM ROBOT PEMBAWA BARANG
MENGUNAKAN ANDROID BERBASIS ARDUINO IOT
(Internet Of Thing)**

**Disusun dan Diajukan untuk Memenuhi Persyaratan Mengikuti Ujian Skripsi
Memperoleh Gelar Sarjana Teknik dari Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan**

SKRIPSI

OLEH

**NAMA : SANDI ANUGRAHA
NPM : 1724210090
PROGRAM STUDI : TEKNIK ELEKTRO
PEMINATAN : TEKNIK ENERGI LISTRIK**

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2019**

RANCANG BANGUN REMOTE SISTEM ROBOT PEMBAWA BARANG MENGGUNAKAN ANDROID BERBASIS ARDUINO IOT (Internet Of Thing)

Sandi Anugraha*

Solly Aryza Lubis**

Muhammad Rizky Syahputra**

Teknik Elektro Universitas Pembangunan Pancabudi

ABSTRAK

Banyaknya porter (jasa pembawa barang) di bandara Indonesia mengakibatkan sering terjadinya kasus seperti koper hilang atau tertukar. Akibat dari masalah tersebut pada tanggal 1 september 2017 pihak manajemen bandara Soekarno Hatta memberikan kebijakan untuk mentiadakan jasa porter di bandara. Dari masalah yang ada, dirancanglah sebuah sistem kontrol troli pembawa barang dengan menggunakan *smartphone* berbasis Arduino - MQTT Hive MQ dengan tujuan mengubah sistem troli (manual) di bandara menjadi sistem troli yang dapat dikendalikan dengan menggunakan *smartphone* serta mampu memberikan informasi letak keberadaan troli (*location tracking*) secara *realtime* kepada pemiliknya melalui internet. Perancangan sistem kontrol ini menggunakan Mikrokontroler Arduino Pro Mini yang terinterkoneksi terhadap GPS, sensor Ultrasonik, ESP 8266, serta motor DC. Sistem troli ini mempunyai tingkat penyimpangan rata-rata pembacaan koordinat posisi troli sebesar 0.13%, penyimpangan rata-rata pembacaan jarak terhadap halangan sebesar 6% , dan delay pelaksanaan perintah oleh troli terhadap Smartphone sebesar 1.16 detik.

Kata kunci: Porter, Arduino Pro Mini, *GPS*, *Smartphone*, ESP 8266, Sensor Ultrasonik, MQTT Hive MQ

* Mahasiswa Program Studi Teknik Elektro : sandi.pln28@gmail.com

** Dosen Program Studi Teknik Elektro

DAFTAR ISI

DAFTAR ISI	i
DAFTAR TABEL	iv
DAFTAR GAMBAR	v

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Metodologi Pengumpulan Data	3
1.7 Sistematika Penulisan	4

BAB 2 DASAR TEORI

2.1 Mikrokontroler Arduino Pro Mini	6
2.1.1 Spesifikasi Arduino Pro Mini	7
2.1.2 Input dan Output (I/O)	8
2.1.3 Komunikasi	10
2.1.4 Pemrograman	11
2.1.5 Bahasa Pemrograman Arduino	12
2.2 Wifi ESP 8266	18
2.2.1 Spesifikasi Umum ESP 8266	19
2.2.2 Pemrograman ESP 8266	20
2.2.3 Parameter	20
2.2.4 Aplikasi Utama	21
2.3 <i>Message Queue Telemetry Transport</i> (MQTT)	22
2.3.1 Arsitektur Komunikasi <i>Publish/Subscribe</i>	22

2.3.2	Hive MQ Broker	23
2.4	<i>Wireless Fidelity (WiFi)</i>	24
2.4.1	IP Adress	25
2.5	Sensor Ultrasonik.....	27
2.5.1	Prinsip Kerja Sensor Ultrasonik.....	20
2.5.2	Rangkaian Sensor Ultrasonik.....	29
2.6	GPS U-Blox Neo 6M.....	33
2.6.1	Fitur dan Spesifikasi.....	34
2.7	Driver Motor DC (L298N)	35
2.7.1	Spesifikasi	35
2.8	Motor DC	36
2.8.1	Komponen Utama	37
2.8.2	Kelebihan Motor DC.....	38
2.9	Buzzer	39
2.9.1	Prinsip Kerja PiezoBuzzer	40
2.10	Lithium Polimer (LiPo) Baterai	41
2.10.1	Tegangan (Voltage).....	42
2.11	Android	42
2.11.1	Arsitektur Android	43

BAB 3 PERANCANGAN ALAT

3.1	<i>Design Prototype</i> Alat	45
3.2	Diagram Alur <i>Prototype</i> Alat	47
3.3	Diagram Blok Sistem.....	48
3.4	Perancangan <i>Hardware</i>	51
3.4.1	Interkoneksi Mikrokontroler terhadap GPS.....	52
3.4.2	Interkoneksi Mikrokontroler terhadap Sensor Ultrasonik.....	52
3.4.3	Interkoneksi Mikrokontroler terhadap Modul WiFi ESP 8266.....	54

3.4.4	Interkoneksi Mikrokontroler terhadap Driver Motor L298N & Baterai LiPo.....	55
3.4.5	Rangkaian Keseluruhan	57
3.5	Perancangan Software.....	58
 BAB 4 PENGUJIAN DAN PEMBAHASAN		
4.1	Pengujian	65
4.1.1	Pengujian Akurasi Pembacaan GPS	65
4.1.2	Pengujian Akurasi Pembacaan Jarak Dengan Sensor Ultrasonik.....	68
4.1.3	Pengujian Delay Sistem	72
 BAB 5 PENUTUP		
5.1	Kesimpulan	75
5.2	Saran	75
DAFTAR PUSTAKA		76
LAMPIRAN.....		77

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Ketika berada di bandara, banyak penumpang yang menggunakan jasa porter untuk membantu mengangkut atau membawakan barang bawaan bagasi penumpang. Terlebih bagi penumpang penerbangan yang memiliki bagasi yang terbilang banyak. Hal tersebut membuat bandara di Indonesia dipenuhi oleh porter. Budaya porter atau tenaga pengangkut ini memang jarang sekali ditemui di bandara luar negeri. Tidak jarang jasa porter disewa oleh penumpang untuk membantu mereka membawa barang bawaannya. Tetapi pada akhir-akhir ini kasus porter nakal pun sudah jamak terjadi di bandara, seperti koper hilang atau tertukar. Akibat dari masalah tersebut pada tanggal 1 September 2017 bandara Soekarno Hatta memberikan kebijakan untuk mentiadakan jasa porter di Indonesia [1], tetapi kebijakan tersebut bukan merupakan solusi yang baik karena para porter akan kehilangan pekerjaannya. Dari sisi penumpang masih membutuhkan solusi sebagai pengganti akibat ketiadaan porter di bandara. Sama halnya pihak bandara pun membutuhkan solusi agar peristiwa kehilangan barang di bandara bisa diminimalkan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan tersebut di atas, tampak bahwa inti permasalahannya adalah sebagai berikut:

- a. Bagaimana membuat system troli yang dapat terhubung dengan internet dan dapat dimonitor letak keberadaannya?

- b. Bagaimana mengontrol pergerakan trolley secara otomatis melalui *smartphone* ?

1.3 Batasan Masalah

Pada tugas akhir ini, perancangan sistem dibatasi dengan spesifikasi teknis sebagai berikut:

- a. Menggunakan mikrokontroler Arduino Pro Mini sebagai pengendali.
- b. Menggunakan GPS untuk memberikan informasi letak keberadaan trolley.
- c. Menggunakan *smartphone* berbasis Android untuk mengendalikan gerak trolley.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian dalam tugas akhir ini adalah untuk memecahkan masalah tersebut di atas dengan cara:

- a. Membuat trolley yang mampu memberikan informasi letak keberadaan trolley (*location tracking*) berupa peta lokasi secara *real time* kepada pemiliknya melalui internet.
- b. Mengubah sistem trolley (manual) di bandara menjadi sistem trolley yang dapat di kendalikan dengan menggunakan *smartphone*.

1.5 Manfaat Penelitian

Adapun Manfaat dari penelitian dalam tugas akhir ini adalah :

- a. Dapat diaplikasikan di semua bandara dan mempermudah pengguna dalam memonitoring barang bawaan.
- b. Optimalisasi teknologi dan data yang sama yang meningkatkan pengalaman mahasiswa untuk meningkatkan penggunaan perangkat, dan membantu

perbaikan teknologi yang lebih maju. IoT (Internet of Thing) membuka dunia data fungsional dan lapangan kritis

1.6 Metodologi Pengumpulan Data

Dalam penelitian dan pembuatan skripsi ini, dilakukan metode – metode pengerjaan sebagai berikut :

a. Studi Lapangan

Studi ini dilaksanakan dengan cara mengumpulkan data dan hal-hal yang berhubungan dengan alat tersebut secara langsung ke lapangan.

b. Studi Literatur

Studi ini dilaksanakan dengan cara mempelajari dan mengumpulkan literatur yang ada, untuk memperoleh data yang berhubungan dengan aplikasi android, sensor (alarm) keamanan, pengendalian jarak jauh dan hal-hal yang berhubungan dengan alat tersebut.

c. Perancangan sistem

Dalam tahap ini dilakukan perancangan diagram blok sistem, dan prinsip kerja, serta penentuan spesifikasi alat yang akan dibuat.

d. Perancangan hardware dan software

e. Integrasi software ke dalam hardware

f. Pengujian sistem

Dalam tahap ini, dilakukan pengujian fungsi masing-masing bagian dan fungsi sistem secara keseluruhan, serta pengukuran besarnya penyimpangan pembacaan sistem yang dibuat dengan mengacu kepada alat ukur standar.

g. Troubleshooting dan penyempurnaan

Jika kinerja sistem belum sesuai dengan yang diharapkan, maka dilakukan *troubleshooting* untuk menemukan penyebabnya. Selanjutnya dilakukan penyempurnaan hingga mencapai kinerja yang diharapkan.

1.7 Sistematika Penulisan

Adapun yang akan menjadi sistematika penulisan yang akan digunakan dalam penulisan skripsi ini adalah sebagai berikut :

BAB I : PENDAHULUAN

Pada bab ini berisi latar belakang masalah, tujuan, batasan masalah, manfaat, metodologi pengumpulan data dan sistematika penulisan.

BAB II : DASAR TEORI

Pada bab ini dibahas mengenai dasar teori serta penjelasan mengenai peralatan yang ada dalam sistem yang diangkat dalam penelitian. Literatur – literatur yang mendukung akan disertakan untuk pemahaman yang baik.

BAB III : PERANCANGAN ALAT

Dalam bab ini membahas tentang perencanaan dan pembuatan rangkaian , pembuatan *software* dan perakitan *hardware*.

BAB IV : PENGUJIAN ALAT

Pada bab ini menjelaskan tentang mekanisme pengujian, data hasil pengujian, analisa terhadap data hasil pengujian tersebut.

BAB V : PENUTUP

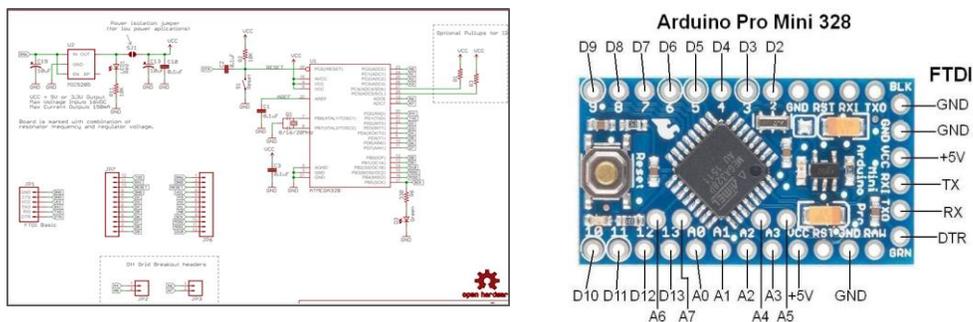
Pada bab ini akan dibahas tentang inti kesimpulan dari pengujian keseluruhan sistem dan disertai saran-saran untuk pengembangan lebih lanjut.

BAB 2

DASAR TEORI

2.1 Mikrokontroler Arduino Pro Mini

Arduino Pro Mini adalah papan pengembangan (development board) mikrokontroler yang berbasis chip ATmega328P dengan bentuk yang sangat mungil dan paling minimalis. Secara fungsi tidak ada bedanya dengan Arduino Uno, dan sangat mirip dengan Arduino Nano. Perbedaan utama terletak pada ketiadaan jack power DC dan konektor Mini-B USB, sehingga harus menggunakan modul FTDI atau USB to TTL untuk menghubungkan ke komputer. Disebut sebagai papan pengembangan karena board ini memang berfungsi sebagai arena prototyping sirkuit mikrokontroller. Dengan menggunakan papan pengembangan, anda akan lebih mudah merangkai rangkaian elektronika mikrokontroller dibanding jika anda memulai merakit ATmega328 dari awal di breadboard. Terdapat dua versi Arduino Pro Mini. Versi 3.3 Volt dan versi 5 Volt, yang dipilih menurut kebutuhan rangkaian mikrokontroller yang anda gunakan.



Gambar 2.1 Schematic Board Arduino Pro Mini [2]

Sumber : arduino, 2018

2.1.1 Spesifikasi Arduino Pro Mini

Adapun spesifikasi detail dari Arduino Pro Mini itu sendiri, sebagai berikut:

Spesifikasi

Chip mikrokontroler	ATmega328P
Tegangan operasi	5V atau 3.3V (tergantung model)
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	6 buah
Arus DC per pin I/O	40 mA
Memori Flash	32 KB, 0.5 KB telah digunakan untuk bootloader
SRAM	2 KB
EEPROM	1 KB
Clock speed	8 Mhz (model 3.3V) atau 16 Mhz (Model 5V)
Dimensi	33 mm x 18 mm
Berat	5 g

Gambar 2.2 Spesifikasi Arduino Pro Mini [3]

Sumber : arduino, 2018

a. 14 pin Input/Output digital (0-13)

Berfungsi sebagai input atau output dan dapat di atur oleh program.

b. 6 Pin Input Analog

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

c. Arus DC – Sumber Daya

Berfungsi untuk memberikan daya dengan sumber 40 mA.

d. Memori Flash

Komponen utama dari papan arduino yang didalamnya terdapat CPU, ROM dan RAM

e. Tombol RESET

Untuk me-reset papan sehingga program akan mulai lagi dari awal, namun tombol reset ini tidak dapat digunakan untuk fungsi menghapus program atau mengosongkan mikrokontroler.

f. 14 pin Input/Output digital (0-13)

Berfungsi sebagai input atau output dan dapat di atur oleh program.

g. 6 Pin Input Analog

Pin ini sangat berguna untuk membaca tegangan yang dihasilkan oleh sensor analog, seperti sensor suhu. Program dapat membaca nilai sebuah pin input antara 0 – 1023, dimana hal itu mewakili nilai tegangan 0 – 5V.

h. Arus DC – Sumber Daya

Berfungsi untuk memberikan daya dengan sumber 40 mA.

i. Memori Flash

Komponen utama dari papan arduino yang didalamnya terdapat CPU, ROM dan RAM

j. Tombol RESET

Untuk me-reset papan sehingga program akan mulai lagi dari awal, namun tombol reset ini tidak dapat digunakan untuk fungsi menghapus program atau mengosongkan mikrokontroler.

2.1.2 Input dan Output (I/O)

Arduino Pro Mini memiliki 14 buah digital pin yang dapat digunakan sebagai input atau output, dengan menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Pin-pin tersebut bekerja pada tegangan 5V, dan setiap pin dapat

menyediakan atau menerima arus 20mA, dan memiliki tahanan pull-up sekitar 20-50k ohm (secara default dalam posisi disconnect). Nilai maximum adalah 40mA, yang sebisa mungkin dihindari untuk menghindari kerusakan chip mikrokontroler

Beberapa pin memiliki fungsi khusus :

1. **Serial**, terdiri dari 2 pin : pin 0 (RX) dan pin 1 (TX) yang digunakan untuk menerima (RX) dan mengirim (TX) data serial.
2. **External Interrupts**, yaitu pin 2 dan pin 3. Kedua pin tersebut dapat digunakan untuk mengaktifkan *interrupts*. Gunakan fungsi *attachInterrupt()*
3. **PWM**: Pin 3, 5, 6, 9, 10, dan 11 menyediakan output PWM 8-bit dengan menggunakan fungsi *analogWrite()*
4. **SPI** : Pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) mendukung komunikasi SPI dengan menggunakan *SPI Library*
5. **LED** : Pin 13. Pada pin 13 terhubung *built-in* led yang dikendalikan oleh digital pin no 13.

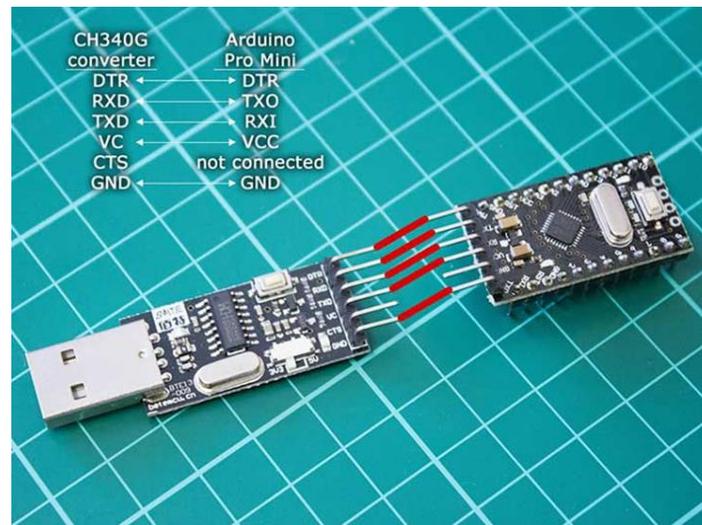
Arduino Pro Mini memiliki 8 buah input analog, yang diberi tanda dengan A0 hingga A7. Masing-masing pin analog tersebut memiliki resolusi 10 bits (jadi bisa memiliki 1024 nilai). Secara default, pin-pin tersebut diukur dari ground ke 5V, namun bisa juga menggunakan pin REF dengan menggunakan fungsi *analogReference()*. Pin Analog A6 dan A7 tidak bisa dijadikan sebagai pin digital, hanya sebagai analog. Beberapa pin lainnya pada board ini adalah :

1. **I2C** : Pin A4 (SDA) dan A5 (SCL). Pin ini mendukung komunikasi I2C (TWI) dengan menggunakan *Wire Library*.

2. **Reset.** Hubungkan ke LOW untuk melakukan reset terhadap mikrokontroller. Biasanya digunakan untuk dihubungkan dengan *switch* yang dijadikan tombol *reset*.

2.1.3 Komunikasi

Arduino Nano 3.0 memiliki beberapa fasilitas untuk berkomunikasi dengan komputer, dan berkomunikasi dengan Arduino lainnya, atau dengan mikrokontroller lainnya. Chip Atmega328 menyediakan komunikasi serial UART TTL (5V) yang tersedia di pin 0 (RX) dan pin 1 (TX). Sebuah chip FTDI yang terdapat pada *board* berfungsi menterjemahkan bentuk komunikasi ini melalui USB dan akan tampil sebagai Virtual Port di komputer. Pada Arduino Software (IDE) terdapat monitor serial yang memudahkan data textual untuk dikirim menuju Arduino atau keluar dari Arduino. Lampu led TX dan RX akan menyala berkedip-kedip ketika ada data yang ditransmisikan melalui chip FTDI USB to Serial via kabel USB ke komputer. Untuk menggunakan komunikasi serial dari digital pin, gunakan Software Serial *library*. Chip ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Di dalam Arduino Software (IDE) sudah termasuk *Wire Library* untuk memudahkan anda menggunakan bus I2C. Untuk menggunakan komunikasi SPI, gunakan SPI *library*. Berikut gambar Arduino Pro Mini dengan USB to TTL (*Converter*).



Gambar 2.3 Arduino Pro Mini dengan USB to TTL [4]
 Sumber : imam solahudin, jurnal arduino, 2017

2.1.4 Pemrograman

Pemrograman *board* Arduino Pro Mini dilakukan dengan menggunakan Arduino Software (IDE). Chip ATmega328 yang terdapat pada Arduino Nano telah diisi program awal yang sering disebut *bootloader*. *Bootloader* tersebut yang bertugas untuk memudahkan dalam melakukan pemrograman lebih sederhana menggunakan Arduino Software, tanpa harus menggunakan tambahan *hardware* lain. Yang dibutuhkan hanyalah *board* FTDI atau USB to Serial, yang dihubungkan ke PC, Mac, atau Linux. Adapun penampakan software Arduino (IDE) sebagai berikut. IDE Arduino adalah software yang sangat canggih dan ditulis dengan menggunakan bahasa “java” dan bahasa “C”. IDE arduino terdiri dari:

1. ***Editor Program***

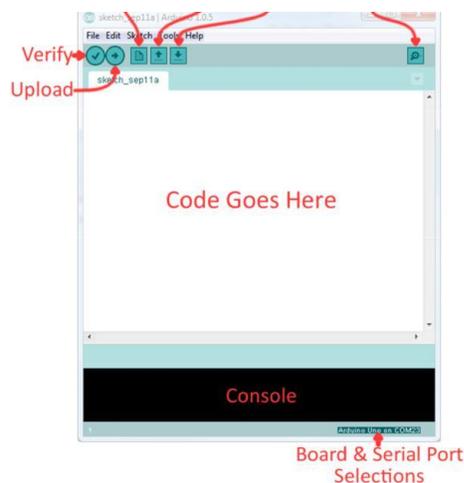
Sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.

2. *Compiler*

Sebuah modul yang mengubah kode program (bahasa *processing*) menjadi kode biner karena kode biner adalah satu – satunya bahasa program yang dipahami oleh *mikrokontroller*.

3. *Uploader*

Sebuah modul yang memuat kode biner dari komputer kedalam memori *mikrokontroller* didalam papan arduino.



Gambar 2.4 Tampilan Software Arduino IDE [5]

Sumber : penulis, 2018

Untuk pengguna mikrokontroller yang sudah lebih mahir, dapat tidak menggunakan *bootloader* dan melakukan pemrograman langsung via *header ICSP (In Circuit Serial Programming)* dengan menggunakan Arduino ISP.

2.1.5 Bahasa Pemrograman Arduino

Arduino menggunakan bahasa program dengan bahasa C. Bahasa ini sudah dipermudah dengan menggunakan fungsi-fungsi yang sederhana sehingga pemula pun dapat mempelajarinya dengan cukup mudah. Berikut adalah penjelasan singkat mengenai karakter bahasa C.

1. Struktur

Setiap pemrograman Arduino (*sketch*) mempunyai dua buah fungsi yang harus terdapat dalam setiap bahasa pemrograman pada Arduino.

a. `void setup () { }`

Semua kode didalam kurung kurawal akan dijalankan hanya satukali ketika program Arduino dijalankan untuk pertama kalinya.

b. `void loop () { }`

Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan secara terus menerus sampe catudaya dilepas.

2. Syntax

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan

a. `//` (komentar satu baris)

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti kode-kode yang dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

b. `/* */` (komentar banyak baris)

Digunakan untuk membuat banyak catatan yang dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

- c. **{ }** (kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai berakhir (digunakan juga pada fungsi dan pengulangan).

- d. **;** (titik koma)

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

3. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara cerdas. Variabel inilah yang digunakan untuk memindahkannya.

- a. **int (*integer*)**

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32.768 dan 32.767.

- b. **long (*long*)**

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2.147.483 dan 2.147.646.

- c. **boolean (*boolean*)**

Variabel sederhana yang digunakan untuk menyimpan nilai TRUE (benar) atau *ELSE* (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

d. **float** (*float*)

Digunakan untuk angka desimal (*floating point*). Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -3,4028235E+38 dan 3,4028235E+38.

e. **Char** (*character*)

Menyimpan 1 karakter menggunakan kode ASCII (misalnya, A" = 65).
Hanya memakai 1 byte (8 bit) dari memori (RAM)

4. Operator Matematika

Operator matematika digunakan untuk memanipulasi angka (bekerja seperti matematika sederhana).

- a. **=** Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: $x=10*2$, x sekarang sama dengan 20).
- b. **%** Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya: $12\% 10$, ini aka menghasilkan angka 2).
- c. **+** Penjumlahan
- d. **-** Pengurangan
- e. ***** Perkalian
- f. **/** Pembagian

5. Operator pembandingan

Digunakan untuk membandingkan nilai logika.

- a. **==** Sama dengan (misalnya: $12 == 10$ adalah *FALSE* (salaa) atau $12 == 12$ adalah *TRUE*(benar))

- b. \neq Sama dengan (misalnya: $12 \neq 10$ adalah *TRUE* atau $12 \neq 12$ adalah *FALSE*).
- c. $<$ Lebih kecil dari (misalnya: $12 < 10$ adalah *FALSE* atau $12 < 12$ adalah *FALSE* atau $12 < 14$ adalah *TRUE*).
- d. $>$ Lebih besar dari (misalnya: $12 > 10$ adalah *TRUE* atau $12 > 12$ adalah *FALSE* atau $12 > 14$ adalah *FALSE*).

6. Struktur pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya. Berikut ini adalah elemen dasar pengaturan.

if....else, dengan format seperti berikut ini: if (kondisi) { }

else if (kondisi) { } else { }

Dengan struktur seperti diatas program akan menjalankan kode yang ada didalam kurung kurawal jika kondisinya *TRUE*, dan jika tidak *FALSE* maka akan diperiksa apakah kondisi pada else if dan jika kondisinya *FALSE* maka kode else yang akan dijalankan.

for, dengan format seperti berikut: for(int i = 0; i < #pengulangan; i++) { }

Digunakan bila ingin melakukan pengulangan kode didalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan perhitungan ke atas dengan i++ atau ke bawah dengan i--.

7. Digital

- a. **pinMode (pin,mode)**

Digunakan untuk menetapkan mode dari suatu pin, pin adalah nomor pin yang akan digunakan dari 0 – 19 (pin analog 0 – 5 adalah 14 – 19). Mode yang bisa digunakan adalah *input* atau *output*.

b. **digitalWrite (pin, value)**

Ketika pin ditetapkan sebagai *output*, pin tersebut dapat dijadikan *HIGH* (ditarik menjadi 5 volt) atau *LOW* (diturunkan menjadi *ground*).

c. **digitalRead (pin)**

Ketika pin ditetapkan sebagai input, maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (ditarik 5 volt) atau *LOW* (diturunkan menjadi *ground*).

8. Analog

a. **analogWrite (pin, value)**

Beberapa pin pada arduino mendukung PWM (*pulse width modulation*) yaitu pin 3, 5, 6, 9, 10, 11. Pin ini dapat merubah pin hidup (*on*) atau mati (*off*) dengan sangat cepat sehingga membuatnya dapat berfungsi seperti keluaran analog. Nilai pada format kode tersebut adalah angka antara 0 (0% *duty cycle* ~ 0V) dan 255 (100% *duty cycle* ~ 5V).

b. **analogRead (pin)**

ketika pin analog ditetapkan sebagai input maka pin tersebut dapat membaca keluaran voltasenya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1.024 (untuk 5 volts).

2.2 Modul WiFi ESP 8266

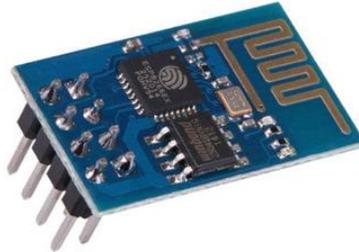
ESP 8266 merupakan modul wifi yang berfungsi sebagai perangkat tambahan mikrokontroler seperti **Arduino** agar dapat terhubung langsung dengan wifi dan membuat koneksi TCP/IP. Modul ini membutuhkan daya sekitar 3.3v dengan memiliki tiga mode wifi yaitu Station, Access Point dan Both (Keduanya). Modul ini juga dilengkapi dengan prosesor, memori dan *General-purpose input/output (GPIO)* dimana jumlah pin bergantung dengan jenis **ESP8266** yang digunakan. Sehingga modul ini bisa berdiri sendiri tanpa menggunakan mikrokontroler apapun karena sudah memiliki perlengkapan layaknya mikrokontroler.

Firmware default yang digunakan oleh perangkat ini menggunakan *attention (AT) Command*, selain itu ada beberapa *Firmware Standard Development Kit (SDK)* yang digunakan oleh perangkat ini berbasis *opensource* yang diantaranya adalah sebagai berikut :

1. **NodeMCU** dengan menggunakan basic programming lua
2. **MicroPython** dengan menggunakan basic programming python
3. **AT Command** dengan menggunakan perintah perintah AT command

Untuk pemrogramannya sendiri digunakan **ESPlorer** untuk Firmware berbasis **NodeMCU** dan menggunakan putty sebagai terminal control untuk AT Command. Selain itu bisa memprogram perangkat ini menggunakan **Arduino IDE**. Dengan menambahkan **library ESP8266** pada board manager dapat dengan mudah memprogram dengan basic program arduino. Ditambah lagi dengan harga yang cukup

terjangkau, dapat membuat berbagai proyek dengan modul ini. Maka dari itu banyak orang yang menggunakannya modul ini untuk membuat proyek *Internet of Think (IoT)*.



Gambar 2.5 Modul WiFi ESP 8266

Sumber : myrobotic, 2019

2.2.1 Spesifikasi Umum ESP 8266

Adapun spesifikasi umum ESP 8266 itu sendiri, sebagai berikut :

1. 802.11 b/g/n
2. Integrated low power 32-bit MCU
3. Integrated 10-bit ADC
4. Integrated TCP/IP protocol stack
5. Integrated TR switch, balun, LNA, power amplifier and matching network
6. Integrated PLL, regulators, and power management units
7. Supports antenna diversity
8. WiFi 2.4 GHz, support WPA/WPA2
9. Support STA/AP/STA+AP operation modes
10. Support Smart Link Function for both Android and iOS devices
11. SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO

12. STBC, 1×1 MIMO, 2×1 MIMO
13. A-MPDU & A-MSDU aggregation & 0.4s guard interval
14. Deep sleep power <10uA, Power down leakage current < 5uA
15. Wake up and transmit packets in < 2ms
16. Standby power consumption of < 1.0mW (DTIM3)
17. +20 dBm output power in 802.11b mode
18. Operating temperature range -40C ~ 125C
19. FCC, CE, TELEC, WiFi Alliance, and SRRC certified

2.2.2 Pemrograman ESP 8266

Pada umumnya, ESP8266 dapat diprogram dengan:

1. AT command via serial komunikasi UART
2. Pemrograman ke mikrokontroler yang ada di ESP8266 menggunakan Arduino IDE dengan Core yang sudah terinstall ESP8266.

Kelebihan lain ESP8266 adalah memiliki *deep sleep mode*, sehingga penggunaan daya akan relatif jauh lebih efisien dibandingkan dengan modul WiFi. Catatan penting yang harus di garis bawahi ialah, ESP8266 beroperasi pada tegangan 3.3V.

2.2.3 Parameter

Adapun parameter ESP 8266 itu sendiri, sebagai berikut :

Tabel 2.1 Parameter ESP 8266

Categories	Items	Values
WiFi Paramters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocles	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)

	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
Types of Antenna	PCB Trace, External, IPEX Connector,	
	Ceramic Chip	
Hardware Parameters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A
Software Parameters	WiFi mode	Station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP
	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App

Sumber : *cnxsoftware, Specification, 2017*

2.2.4 Aplikasi Utama

Adapun Aplikasi Utama Esp 8266 yang bisa diterapkan ke *Intenet of Things* (IoT) adalah sebagai berikut :

1. Wifi Location Aware Device
2. Otomasi Rumah

3. Otomasi Lampu Pintar
4. Pembuatan Jaringan
5. Industrial Wireless Control
6. Monitoring Bayi
7. IP Camera
8. Security ID Tags

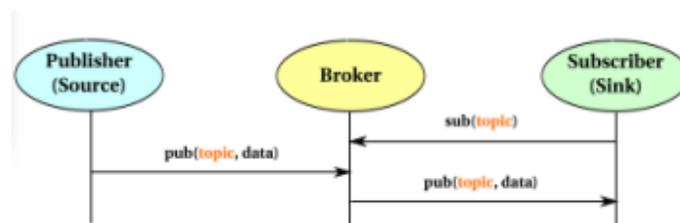
2.3 Message Queue Telemetry Transport (MQTT)

MQTT merupakan singkatan dari *Message Queue Telemetry Transport*. Ini merupakan protokol komunikasi *publish/subscribe topic based* yang sangat sederhana dan ringan, yang didesain untuk alat yang memiliki kemampuan terbatas, bandwidth yang rendah, latency yang tinggi atau jaringan yang kurang dapat diandalkan. Prinsip dari desain ini adalah untuk meminimalkan penggunaan bandwidth jaringan dan kebutuhan sumber daya pada perangkat serta pada waktu yang sama juga berusaha untuk memastikan keandalan dan kepastian dari pengiriman data. Prinsip yang ada ini juga memunculkan beberapa ide protokol mengenai “*machine-to-machine*” (M2M) atau IoT yang menginginkan perangkat di dunia untuk saling terhubung, dan untuk aplikasi *mobile* dimana *bandwidth* dan daya baterai pada keadaan yang cukup (MQTT, 2016).

2.3.1 Arsitektur Komunikasi Publish/Subscribe

Prinsip dari model komunikasi *publish / subscribe* yaitu beberapa komponen yang menginginkan sejumlah informasi yang sesuai dengan topik mereka inginkan dengan cara mendaftarkan topik apa yang diinginkan. Proses mendaftarkan topik ini

disebut dengan *subscribe*. Sedangkan kumpulan komponen yang menginginkan informasi tersebut disebut sebagai *subscriber*. Komponen lain membuat atau memberikan informasi terkait dengan yang diinginkan oleh *subscriber* dengan cara melakukan *publish* informasi. Komponen itu disebut sebagai *publisher*. Sedangkan entitas yang bertugas untuk memastikan bahwa suatu paket informasi dapat terkirim dari *publisher* menuju *subscriber* disebut *broker*. *Broker* bertugas untuk mengkoordinasi keinginan akan informasi dari para *subscriber*, dan *subscriber* biasanya secara eksplisit akan melakukan kontak dengan *broker* untuk melakukan *subscribe*. (Hunkeler, Truong, & Clark, 2015).

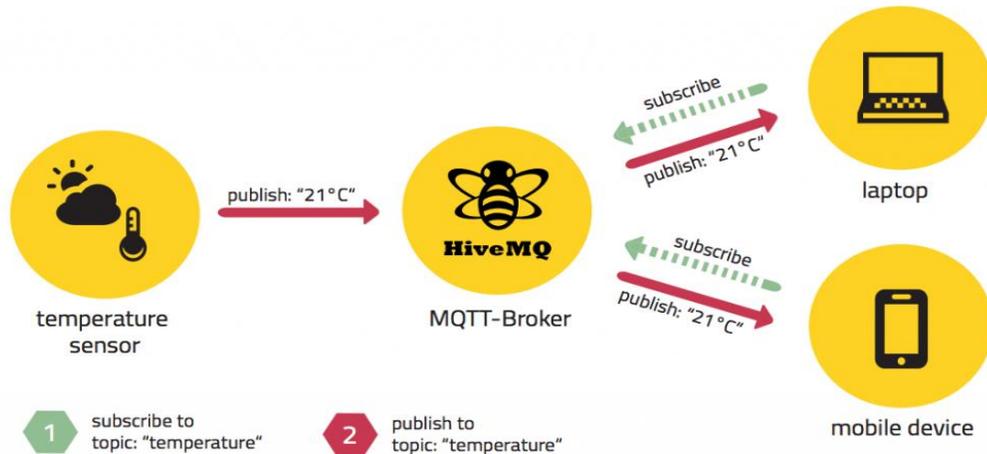


Gambar 2.6 Model Komunikasi *Topic-Based Publish/Subscribe* [7]

Sumber : *MQTT client user guide book, 2017*

2.3.2 Hive MQ Broker

Hive MQ adalah broker untuk menghubungkan semua perangkat dan layanan dengan menggunakan protocol IoT MQTT. Hive MQ sistem yang memasikan semua perangkat dan layanan dapat berbagi informasi secara *instant*, pesan langsung dua arah dan dapat mengirim data ke sistem *back end secara real time*. Penggunaan broker Hive MQ biasa di terapkan di Industri seperti bidang kesehatan, komunikasi *mobile, retail, smart home*. Karakteristik Hive MQ sendiri yaitu skabilitas, keamanan, simple, handal.



Gambar 2.7 Model Komunikasi menggunakan MQTT Broker [8]
 Sumber : *Mosquitto MQTT user guide book, 2018*

2.4 Wireless Fidelity (WiFi)

Wireless Fidelity (WiFi) adalah sebuah teknologi terkenal yang memanfaatkan peralatan elektronik untuk bertukar data secara nirkabel (menggunakan gelombang radio) melalui sebuah jaringan komputer.

Untuk penggunaan internet, wifi memerlukan sebuah titik akses yang biasa disebut dengan hotspot untuk menghubungkan dan mengontrol antara pengguna wifi dengan jaringan internet pusat. Sebuah hotspot pada umumnya dilengkapi dengan password yang bisa meminimalisasi siapa saja yang bisa menggunakan fasilitas tersebut. Ini sering digunakan oleh pengguna rumahan, restoran, swalayan, café dan hotel. Namun ada juga hotspot yang tidak diberi password, sehingga siapa saja boleh menggunakan fasilitas tersebut. Misalnya taman hiburan dan stadion.

Kelebihan dari menggunakan wifi yaitu lebih portable, artinya pengguna tidak repot dengan memikirkan kabel penghubung ke hotspot. Dan juga, akses transfer data

lebih cepat (ini bisa diperoleh jika peralatan yang di gunakan masih dalam jangkauan pusat hotspot. Namun ada juga kelemahan dari wi-fi ini, yaitu sering di hack (dibobol) oleh orang lain. Saat ini WiFi dapat dinikmati dan diakses melalui berbagai jenis alat elektronik yang semakin canggih, seperti laptop, dan smartphone.

2.4.1 IP Adress

Untuk bisa saling berkomunikasi komputer memerlukan IP Address, IP address merupakan alamat pengenalan standar untuk semua perangkat di jaringan yang menggunakan standar protocol internet. IP address inilah yang menjadi label kemana paket data akan dikirim dan diterima. jadi semua perangkat yang terhubung ke jaringan harus mempunyai IP Address yang menjadi alamat standar pengiriman dan penerimaan data. Pada umumnya IP address yang sering digunakan adalah IPV4 seperti contoh : 192.168.100.102 dan IPV6 untuk saat ini hanya digunakan untuk server yang berada di jaringan internet. Terdapat dua jenis IP yaitu IP *Private* dan IP *Public*.

1. IP Private (IP Local)

Disebut private karena IP ini hanya bisa diakses dari jaringan local saja dan tidak bisa diakses melalui jaringan internet secara langsung tanpa bantuan router(NAT). IP private digunakan untuk jaringan local (LAN) agar sesama komputer dapat saling berkomunikasi, misalnya digunakan di jaringan sekolah,kantor,toko,dll. perangkat kantor /gadget seperti printer,komputer,laptop,smartdevice menggunakan ip private.

2. IP Publik

IP publik adalah IP yang digunakan dalam jaringan global Internet, karena kelas IP ini digunakan di dalam jaringan internet maka IP ini bisa diakses

melalui jaringan internet secara langsung. Perangkat yang menggunakan IP public biasanya bertindak sebagai SERVER atau ROUTER , seperti web server,mailserver,DNS server,Game Server dan server yang lain dengan tujuan agar server bisa diakses dari jaringan manapun di dunia ini yang terkoneksi ke internet. Untuk dapat menggunakan IP public, suatu organisasi harus mendaftarkan diri agar mendapat alokasi IP public yang akan digunakan.IP public bisa didapat dari ISP (internet service provider).

Ada 2 jenis IP berdasar persistensi ip tersebut menempel di perangkat komputer, yaitu IP Static dan IP Dynamic. (tidak peduli apakah IP Public atau IP private), diantaranya yaitu :

1. Dynamic IP Address

Disebut IP Dynamic / dinamis karena IP yang didapat perangkat komputer akan berganti setiap kalo perangkat komputer direstart, seperti IP Address yang didapat modem speedy, akan berubah setiap kali modem direstart. dan IP yang didapat pada saat konek ke hotspot. IP dinamik tidak memerlukan settingan manual karena IP akan otomatis dialokasikan oleh Router(DHCP Server) ke perangkat yang terhubung ke perangkat komputer yang terhubung ke jaringan

2. Static IP Address

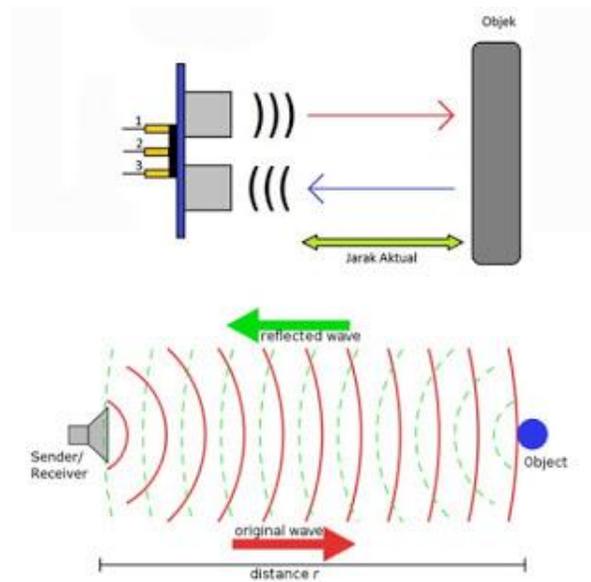
Static IP atau ip statis adalah ip yang dipasang secara manual atau IP yang dialokasikan khusus oleh router untuk suatu perangkat komputer (router,tablet,smartphone,smart device dll),sehingga ketika perangkat direstart IP tidak berubah.

2.5 Sensor Ultrasonik

Sensor ultrasonik adalah sensor yang berfungsi mengubah besaran fisis (bunyi) menjadi besaran listrik atau sebaliknya. Cara kerja sensor ini didasarkan pada prinsip pemantulan gelombang. Hasil pantulan gelombang suara digunakan untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sensor ultrasonic karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik). *Gelombang ultrasonik* adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu kisaran 20.000 Hz. Bunyi ultrasonik tidak dapat didengar oleh telinga manusia. akan tetapi, bunyi ultrasonik dapat didengar oleh hewan seperti : anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik bisa merambat melalui medium zat padat, cair maupun gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair.

2.5.1 Prinsip Kerja Sensor Ultrasonik

Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.



Gambar 2.8 cara kerja sensor ultrasonik dengan transmitter dan receiver (atas), sensor ultrasonik dengan single sensor yang berfungsi sebagai transmitter dan receiver [9]

Sumber : elangsakti, sensor ultrasonic, Jurnal Teknik, 2018

1. Sinyal dipancarkan oleh pemancar ultrasonik. Sinyal tersebut berfrekuensi diatas 20kHz, biasanya yang digunakan untuk mengukur jarak benda adalah 40kHz. Sinyal tersebut dibangkitkan oleh rangkaian pemancar ultrasonik.
2. Sinyal yang dipancarkan tersebut kemudian akan merambat sebagai sinyal / gelombang bunyi dengan kecepatan bunyi yang berkisar 340 m/s. Sinyal tersebut kemudian akan dipantulkan dan akan diterima kembali oleh bagian penerima Ultrasonik.
3. Setelah sinyal tersebut sampai di penerima ultrasonik, kemudian sinyal tersebut akan diproses untuk menghitung jaraknya. Jarak dihitung berdasarkan rumus :

$$S = 340.t/2 \quad (2.1)$$

“ dimana S adalah jarak antara sensor ultrasonik dengan bidang pantul, dan t adalah selisih waktu antara pemancaran gelombang ultrasonik sampai diterima kembali oleh bagian penerima ultrasonik”.

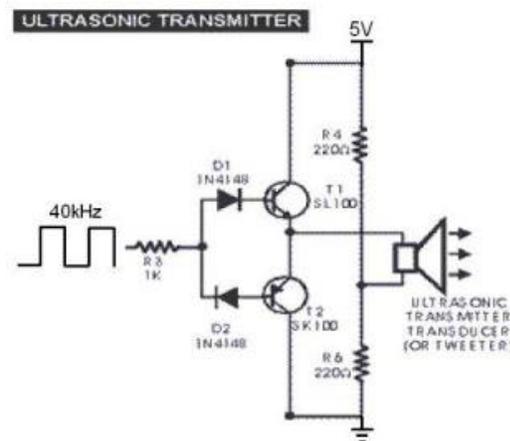
Besar amplitudo sinyal elektrik yang dihasilkan unit sensor penerima tergantung dari jauh dekatnya objek yang dideteksi serta kualitas dari sensor pemancar dan sensor penerima. Proses sensing yang dilakukan pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran. Jarak antara sensor tersebut dihitung dengan cara mengalikan setengah waktu yang digunakan oleh sinyal ultrasonik dalam perjalanannya dari rangkaian Tx sampai diterima oleh rangkaian Rx, dengan kecepatan rambat dari sinyal ultrasonik tersebut pada media rambat yang digunakannya, yaitu udara.

2.5.2 Rangkaian Sensor Ultrasonik

Piezoelektrik berfungsi untuk mengubah energi listrik menjadi energi mekanik. Bahan *piezoelektrik* adalah material yang dapat memproduksi medan listrik ketika dikenai regangan atau tekanan mekanis. Sebaliknya, jika medan listrik diterapkan, maka material tersebut akan mengalami regangan atau tekanan mekanis. Jika rangkaian pengukur beroperasi pada mode pulsa elemen piezoelektrik yang sama, maka 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.

1. Pemancar Ultrasonik (*Transmitter*)

Pemancar Ultrasonik ini berupa rangkaian yang memancarkan sinyal sinusoidal berfrekuensi di atas 20 KHz menggunakan sebuah *transducer transmitter* ultrasonik



Gambar 2.9 Rangkaian Pemancar Gelombang Ultrasonik [10]

Sumber : elangsakti, sensor ultrasonic, Jurnal Teknik 2015

Prinsip kerja dari rangkaian pemancar gelombang ultrasonik tersebut adalah sebagai berikut :

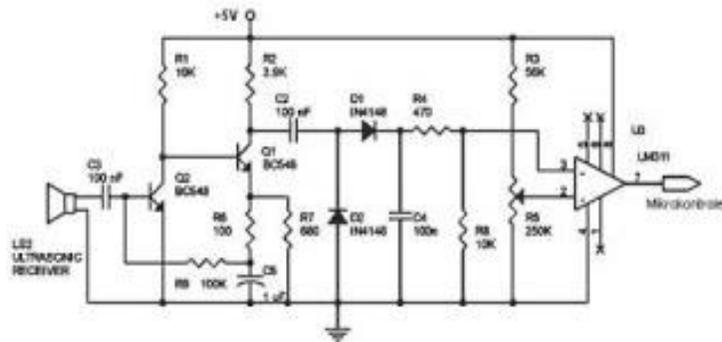
- Sinyal 40 kHz dibangkitkan melalui mikrokontroler.
- Sinyal tersebut dilewatkan pada sebuah resistor sebesar 3kOhm untuk pengaman ketika sinyal tersebut membias maju rangkaian dioda dan transistor.
- Kemudian sinyal tersebut dimasukkan ke rangkaian penguat arus yang merupakan kombinasi dari 2 buah dioda dan 2 buah transistor.
- Ketika sinyal dari masukan berlogika tinggi (+5V) maka arus akan melewati dioda D1 (D1 on), kemudian arus tersebut akan membias

transistor T1, sehingga arus yang akan mengalir pada kolektor T1 akan besar sesuai dari penguatan dari transistor.

- e. Ketika sinyal dari masukan berlogika tinggi (0V) maka arus akan melewati dioda D2 (D2 on), kemudian arus tersebut akan membias transistor T2, sehingga arus yang akan mengalir pada kolektor T2 akan besar sesuai dari penguatan dari transistor.
- f. Resistor R4 dan R6 berfungsi untuk membagi tegangan menjadi 2,5 V. Sehingga pemancar ultrasonik akan menerima tegangan bolak – balik dengan $V_{peak-peak}$ adalah 5V (+2,5 V s.d -2,5 V).

2. Penerima Ultrasonik (*Receiver*)

Penerima Ultrasonik ini akan menerima sinyal ultrasonik yang dipancarkan oleh pemancar ultrasonik dengan karakteristik frekuensi yang sesuai. Sinyal yang diterima tersebut akan melalui proses filterisasi frekuensi dengan menggunakan rangkaian *band pass filter* (penyaring pelewat pita), dengan nilai frekuensi yang dilewatkan telah ditentukan. Kemudian sinyal keluarannya akan dikuatkan dan dilewatkan ke rangkaian komparator (pembanding) dengan tegangan referensi ditentukan berdasarkan tegangan keluaran penguat pada saat jarak antara sensor kendaraan mini dengan sekat/dinding pembatas mencapai jarak minimum untuk berbelok arah. Dapat dianggap keluaran komparator pada kondisi ini adalah *high* (logika '1') sedangkan jarak yang lebih jauh adalah *low* (logika '0'). Logika-logika biner ini kemudian diteruskan ke rangkaian pengendali (mikrokontroler).



Gambar 2.10 Rangkaian Penerima Gelombang Ultrasonik [11]
Sumber : elangsakti, sensor ultrasonic, Jurnal Teknik, 2015

Prinsip kerja dari rangkaian pemancar gelombang ultrasonik tersebut adalah sebagai berikut :

- a. Pertama – tama sinyal yang diterima akan dikuatkan terlebih dahulu oleh rangkaian transistor penguat Q2.
- b. Kemudian sinyal tersebut akan di filter menggunakan High pass filter pada frekuensi $> 40\text{kHz}$ oleh rangkaian transistor Q1.
- c. Setelah sinyal tersebut dikuatkan dan di filter, kemudian sinyal tersebut akan disearahkan oleh rangkaian dioda D1 dan D2.
- d. Kemudian sinyal tersebut melalui rangkaian filter low pass filter pada frekuensi $< 40\text{kHz}$ melalui rangkaian filter C4 dan R4.
- e. Setelah itu sinyal akan melalui komparator Op-Amp pada U3.

Jadi ketika ada sinyal ultrasonik yang masuk ke rangkaian, maka pada komparator akan mengeluarkan logika rendah (0V) yang kemudian akan diproses oleh mikrokontroler untuk menghitung jaraknya.

2.6 Modul GPS U-Blox Neo 6M

Global Positioning System (GPS) adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (synchronization) sinyal satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu.



Gambar 2.11 Modul GPS U Blox Neo 6M [12]
Sumber : penulis, 2019

Modul GPS ini adalah keluarga dari *stand-alone GPS receivers* yang memiliki fitur dengan performa tinggi sebagai mesin penentu posisi. Modul *flexible* dan murah ini menawarkan beberapa pilihan koneksi dengan ukuran 16 x 12.2 x 2.4 mm. Dengan arsitektur, power, dan memory yang optimal modul ini sangat cocok untuk device yang menggunakan battery sebagai sumber daya dengan biaya dan *space* yang terbatas sehingga sangat cocok untuk digunakan pada CanSat. Dengan memiliki 50 kanal *positioning engine* akan mempercepat Time-To-First-Fix (TTFF) kurang dari 1 detik.

2.6.1 Fitur & Spesifikasi

1. U-blox NEO-6M GPS module

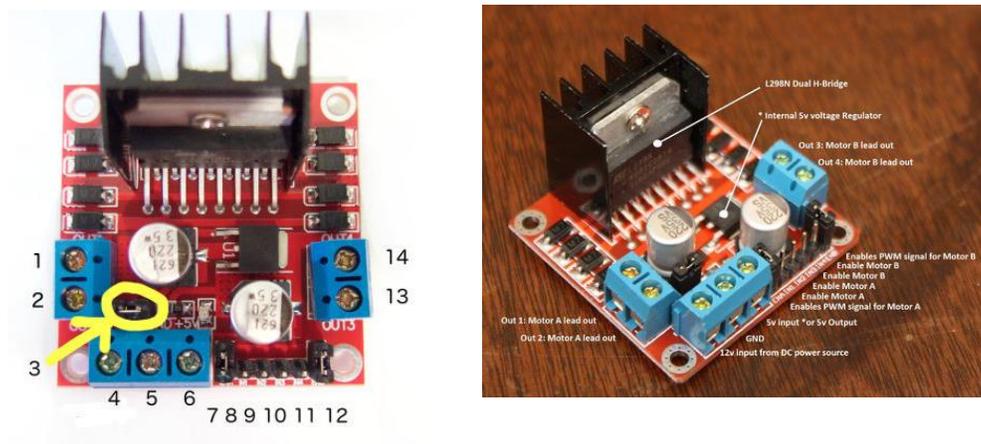
2. Under 1 second time-to-first-fix for hot and aided starts
3. SuperSense ® Indoor GPS: -162 dBm tracking sensitivity
4. Anti-jamming technology
5. Support SBAS (WAAS, EGNOS, MSAS, GAGAN)
6. u-blox 6 50 channel positioning engine with over 2 million effective correlators
7. Timepulse
8. 5Hz position update rate
9. Operating temperature range: -40 TO 85°C
10. UART TTL socket
11. EEprom to store settings
12. Rechargeable battery for Backup
13. Build in 18X18mm GPS antenna
14. RoHS compliant
15. Dimension : 22mmX30mm
16. Height : 13mm
17. Hole dia : 3mm
18. Weight : 12g
19. Antenna Socket : IPEX
20. Output Protocol : NMEA

2.7 Driver Motor DC (L298N)

Driver motor L298N merupakan driver motor yang paling populer digunakan untuk mengontrol atau mengendalikan kecepatan dan arah pergerakan motor DC. IC L298 merupakan sebuah IC tipe H- bridge yang mampu mengendalikan beban induktif seperti relay, solenoid, motor DC dan motor stepper.

2.7.1 Spesifikasi

Adapun spesifikasi dari driver motor DC adalah sebagai berikut :



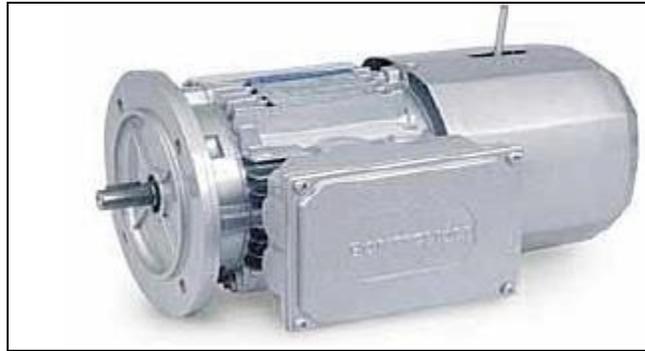
Gambar 2.12 Rangkaian Driver Motor DC [13]
Sumber :makasar robotika, motor driver, 2016

1. DC motor 1 “+” atau stepper motor A+
2. DC motor 1 “-” atau stepper motor A-
3. 12V jumper – lepaskan jumper ini jika menggunakan sumber lebih dari 12 V DC. Ini memungkinkan sumber dari regulator pada Arduino 5V.
4. Hubungkan sumber tegangan motor disini, maksimum 35 V DC. Lepaskan 12V jumperV jika >12V.

5. GND
6. 5V output jika 12V jumper digunakan, ideal untuk mensuplai Arduino Anda(etc).
7. DC motor 1 enable jumper. Lepaskan ini ketika menggunakan stepper motor. Hubungkan ke keluaran PWM untuk menatur kecepatan motor.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Lepaskan ini ketika menggunakan stepper motor. Hubungkan ke keluaran PWM untuk menatur kecepatan motor.
13. DC motor 2 “+” or stepper motor B+
14. DC motor 2 “-” or stepper motor B-

2.8 Motor DC

Motor DC adalah jenis motor listrik yang bekerja menggunakan sumber tegangan DC. Motor DC atau motor arus searah sebagaimana namanya, menggunakan arus langsung dan tidak langsung/direct-unidirectional. Motor DC digunakan pada penggunaan khusus dimana diperlukan penyalaan torque yang tinggi atau percepatan yang tetap untuk kisaran kecepatan yang luas.



Gambar 2.13 Motor DC [14]
Sumber: flipkart, motor DC, 2017

2.8.1 Komponen Utama

Gambar diatas memperlihatkan sebuah motor DC yang memiliki tiga komponen utama :

1. Kutub Medan Magnet

Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan kumparan motor DC yang menggerakkan bearing pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.

2. Kumparan Motor DC

Bila arus masuk menuju kumparan motor DC, maka arus ini akan menjadi elektromagnet. kumparan motor DC yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil,

kumparan motor DC berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan kumparan motor DC.

3. Commutator Motor DC

Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam kumparan motor DC. Commutator juga membantu dalam transmisi arus antara kumparan motor DC dan sumber daya.

2.8.2 Kelebihan Motor DC

Keuntungan utama motor DC adalah dalam hal pengendalian kecepatan motor DC tersebut, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur :

1. Tegangan kumparan motor DC – meningkatkan tegangan kumparan motor DC akan meningkatkan kecepatan
2. Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Motor DC tersedia dalam banyak ukuran, namun penggunaannya pada umumnya dibatasi untuk beberapa penggunaan berkecepatan rendah, penggunaan daya rendah hingga sedang seperti peralatan mesin dan rolling mills, sebab sering terjadi masalah dengan perubahan arah arus listrik mekanis pada ukuran yang lebih besar. Juga, motor tersebut dibatasi hanya untuk penggunaan di area yang bersih dan tidak berbahaya sebab resiko percikan api pada sikatnya.

Motor DC juga relatif mahal dibanding motor AC. Hubungan antara kecepatan, flux medan dan tegangan kumparan motor DC ditunjukkan dalam persamaan berikut :

$$N = \frac{V_{TM} - I_A R_A}{K\Phi} \quad (2.2)$$

dimana:

V_{TM} = Tegangan Terminal

Φ = Flux magnet yang terbentuk pada motor

I_A = Arus kumparan motor DC

K = Konstanta Motor

R_A = Hambatan kumparan motor DC

2.9 Buzzer

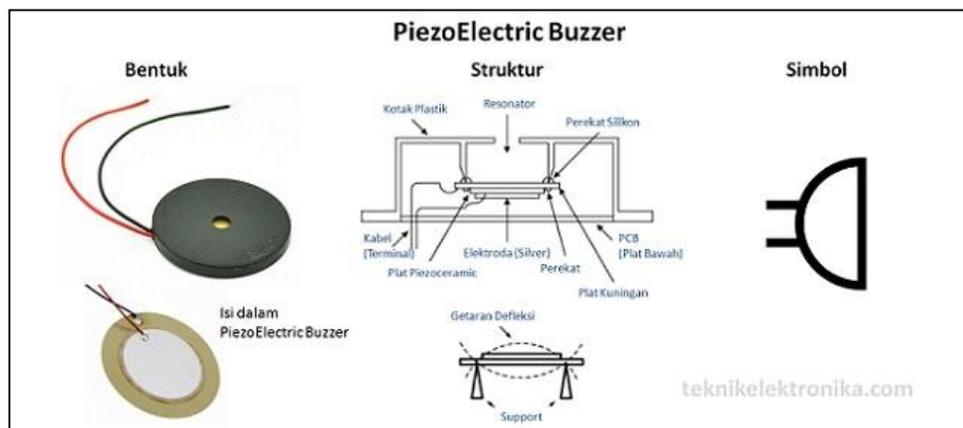
Buzzer Listrik adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Pada umumnya, Buzzer yang merupakan sebuah perangkat audio ini sering digunakan pada rangkaian anti-maling, Alarm pada Jam Tangan, Bel Rumah, peringatan mundur pada Truk dan perangkat peringatan bahaya lainnya. Jenis Buzzer yang sering ditemukan dan digunakan adalah Buzzer yang berjenis Piezoelectric, hal ini dikarenakan Buzzer Piezoelectric memiliki berbagai kelebihan seperti lebih murah, relatif lebih ringan dan lebih mudah dalam menggabungkannya ke Rangkaian Elektronika lainnya. Buzzer yang termasuk dalam keluarga Transduser ini juga sering disebut dengan Beeper.

Piezoelectric Buzzer adalah jenis Buzzer yang menggunakan efek Piezoelectric untuk menghasilkan suara atau bunyinya. Tegangan listrik yang diberikan ke bahan Piezoelectric akan menyebabkan gerakan mekanis, gerakan tersebut kemudian diubah

menjadi suara atau bunyi yang dapat didengar oleh telinga manusia dengan menggunakan diafragma dan resonator.

2.9.1 Prinsip Kerja *PiezoBuzzer*

Berikut ini adalah gambar bentuk dan struktur dasar dari sebuah Piezoelectric Buzzer.



Gambar 2.14 Piezoelectric Buzzer [15]

Sumber : Dickson Kho, Jurnal Teknik Elektronika, 2018

Piezo Buzzer dapat bekerja dengan baik dalam menghasilkan frekuensi di kisaran 1 – 5 kHz hingga 100 kHz untuk aplikasi Ultrasound. Tegangan Operasional Piezoelectric Buzzer yang umum biasanya berkisar diantara 3Volt hingga 12 Volt.

2.10 Lithium Polimer (LiPo) Baterai



Gambar 2.15 Lipo Battery [15]

Sumber : amazon, battery Lipo, 2018

Baterai Lithium Polimer atau biasa disebut dengan LiPo merupakan salah satu jenis baterai yang sering digunakan dalam dunia RC. Utamanya untuk RC tipe pesawat dan helikopter. Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada charging dan discharging rate. Masalah ini sebenarnya bisa diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini dianggap tidak dapat untuk diaplikasikan pada keadaan sehari-hari. Seandainya para ilmuwan dapat memecahkan masalah ini maka risiko keamanan pada batera jenis lithium akan sangat berkurang.

2.10.1 Tegangan Baterai Lipo

Baterai Lipo memiliki rating 3.7 Volt per sel. Keuntungannya adalah tegangan baterai yang tinggi dapat dicapai dengan menggunakan jumlah sel yang lebih sedikit. Pada setiap paket baterai LiPo selain tegangan ada label yang disimbolkan dengan “S”. Disini “S” berarti sel yang dimiliki sebuah paket baterai (battery pack). Sementara bilangan yang berada didepan simbol menandakan jumlah sel dan biasanya berkisar antar 2-6S (meskipun kadang ada yang mencapai 10S). Berikut adalah beberapa contoh notasi baterai LiPo.

1. 3.7 volt battery = 1 cell x 3.7 volts
2. 7.4 volt battery = 2 cells x 3.7 volts (2S)
3. 11.1 volt battery = 3 cells x 3.7 volts (3S)
4. 14.8 volt battery = 4 cells x 3.7 volts (4S)
5. 18.5 volt battery = 5 cells x 3.7 volts (5S)
6. 22.2 volt battery = 6 cells x 3.7 volts (6S)

2.11 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android dibentuklah Open Handset Alliance, konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, T-Mobile dan Nvidia. Pada saat perilis perdana Android, 5

November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Sedangkan Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

2.11.1 Arsitektur Android

Secara garis besar arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

1. Applications dan Widgets

Merupakan layer dimana kita dapat berhubungan dengan aplikasi apa saja, dimana biasanya kita men-download suatu aplikasi kemudian dilakukan instalasi dan menjalankan aplikasi tersebut. Di-layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak dan lain-lain. Semua aplikasi tersebut ditulis menggunakan bahasa pemrograman Java.

2. Applications Frameworks

Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resource, menjalankan service background, mengatur alarm, dan menambah status notification, dan sebagainya. Pengembang memiliki akses penuh menuju Application Programming Interface (API) framework seperti yang dilakukan oleh aplikasi yang berkategori inti, sehingga dapat disimpulkan Application Framework merupakan layer dimana para pembuat aplikasi dapat melakukan

pengembangan/pembuatan aplikasi yang akan dijalankan pada sistem aplikasi Android.

3. *Libraries*

Libraries merupakan layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libs dan SSL.

4. *Android Run Time*

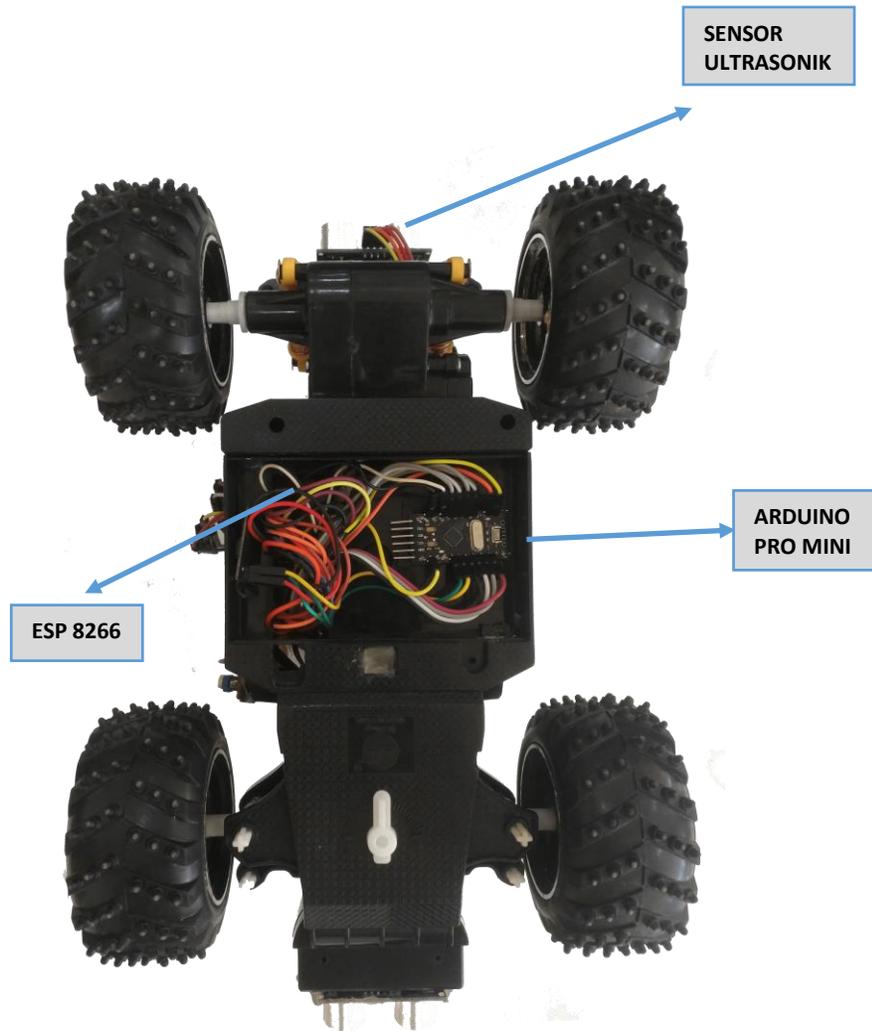
Merupakan *layer* yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux.

BAB 3

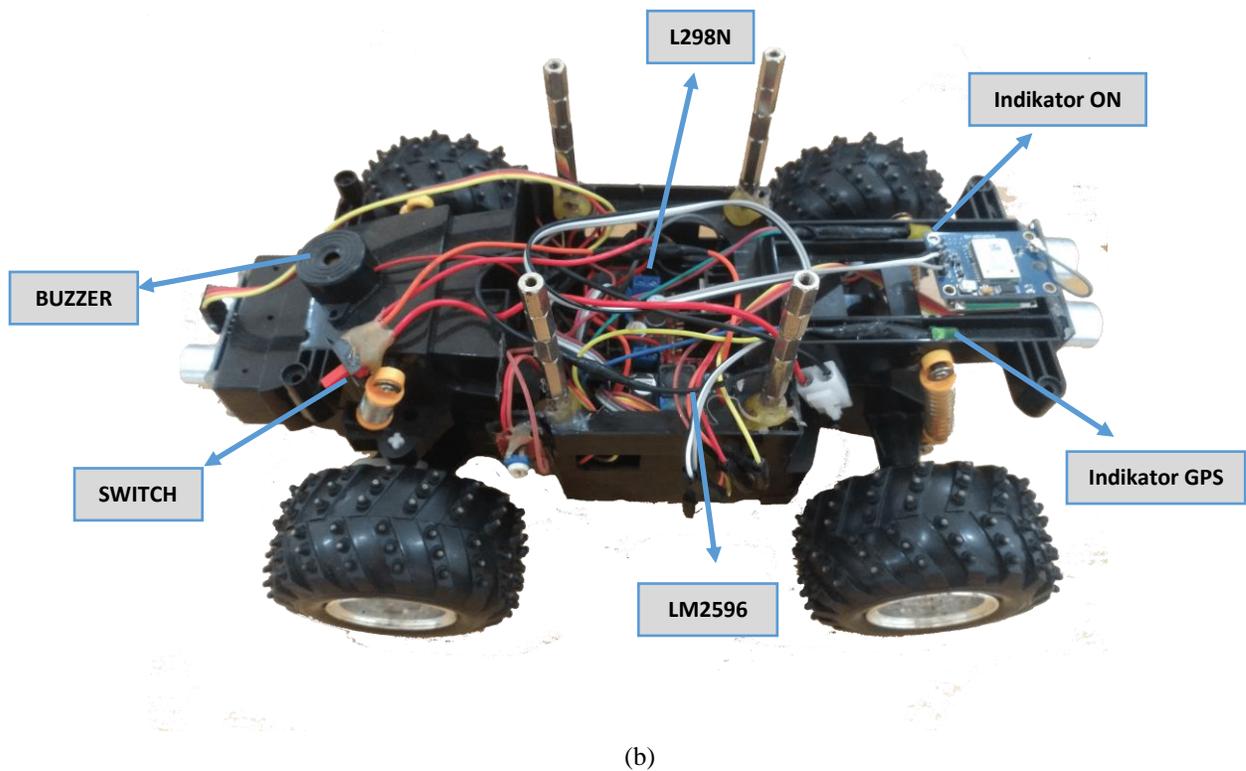
PERANCANGAN ALAT

3.1 *Design Prototype Alat*

Berikut ini adalah implementasi rancangan yang sudah dibuat :



(a)

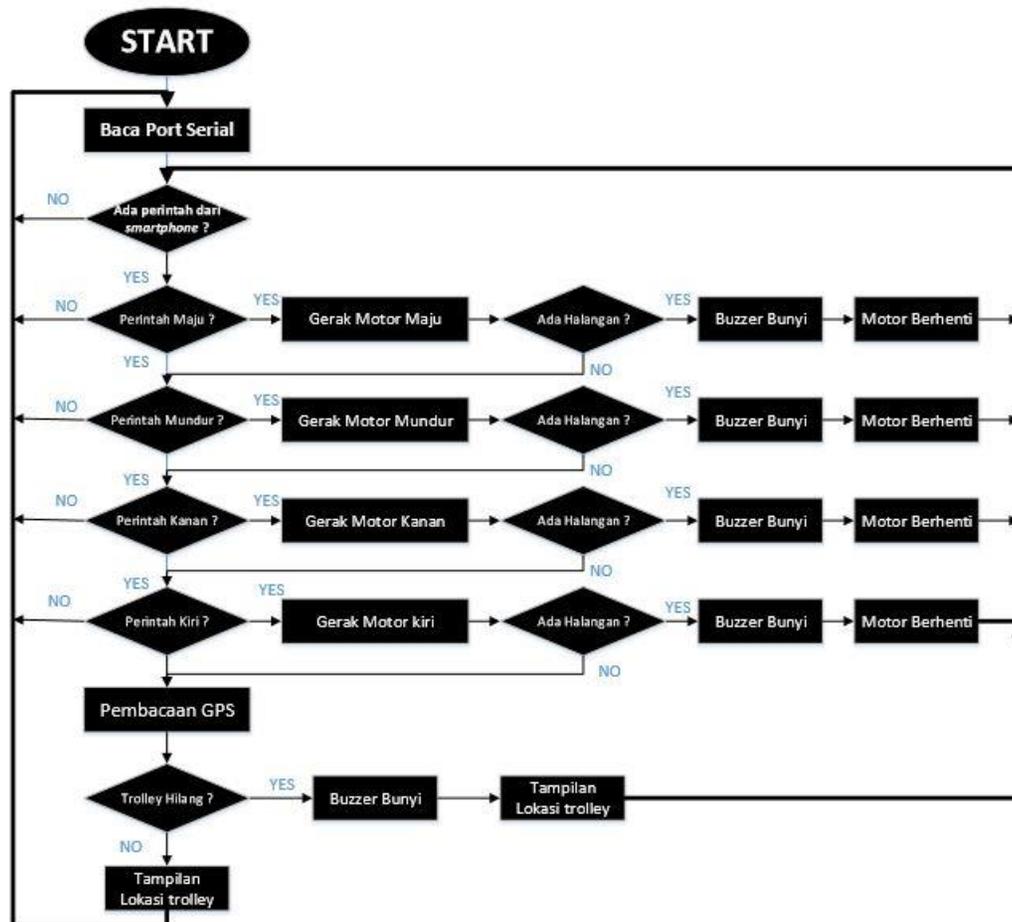


Gambar 3.1 Prototype *Robot* Pembawa Barang Yang Dikontrol Oleh *Smartphone*
 (a) Penempatan ESP dan Mikrokontroler (b) Penempatan semua *sensor*
 Sumber : *Penulis*, 2019

Pada Gambar 3.1 merupakan *prototype robot* yang dapat dikontrol melalui *Smartphone Android*. Selanjutnya setelah berhasil melakukan implementasi, perlu dilakukan pengujian untuk melihat *performance* sistem. Pengujian tersebut dilakukan terhadap fungsi GPS, sensor ultrasonik pendeteksi halangan, dan delay pelaksanaan perintah *Smartphone* oleh *prototype robot* pembawa barang.

3.2 Diagram Alur *Prototype* Alat

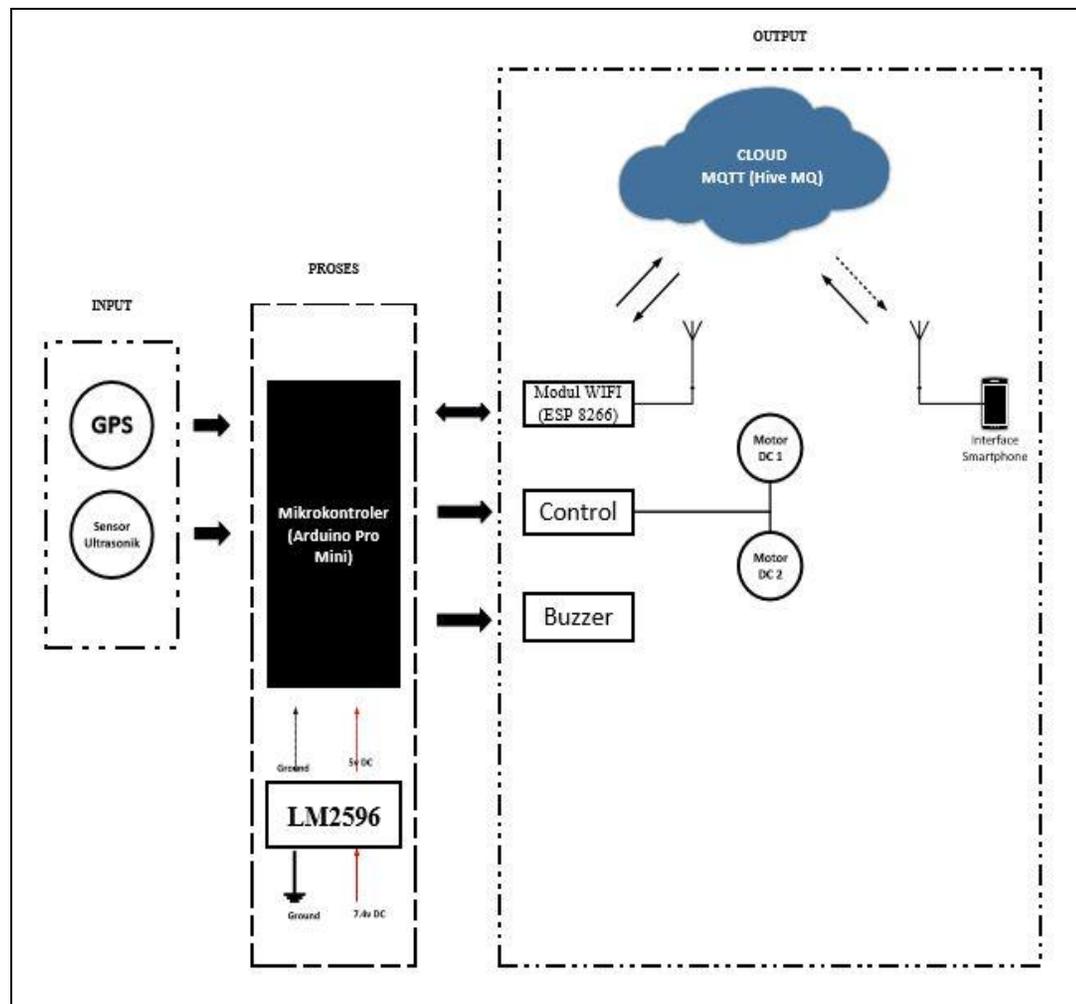
Berikut adalah tahapan yang dilakukan untuk mengoperasikan *prototype robot* pembawa barang yang dikontrol melalui *smartphone Android*



Gambar 3.2 *Flowchart software* Mikrokontroler Arduino Pro Mini
 Sumber :Penulis, 2019

3.3 Diagram Blok Sistem

Berikut ini adalah diagram blok sistem kontrol *robot* pembawa barang menggunakan android berbasis Arduino IOT (Internet Of Thing).



Gambar 3.3 Diagram Blok Sistem

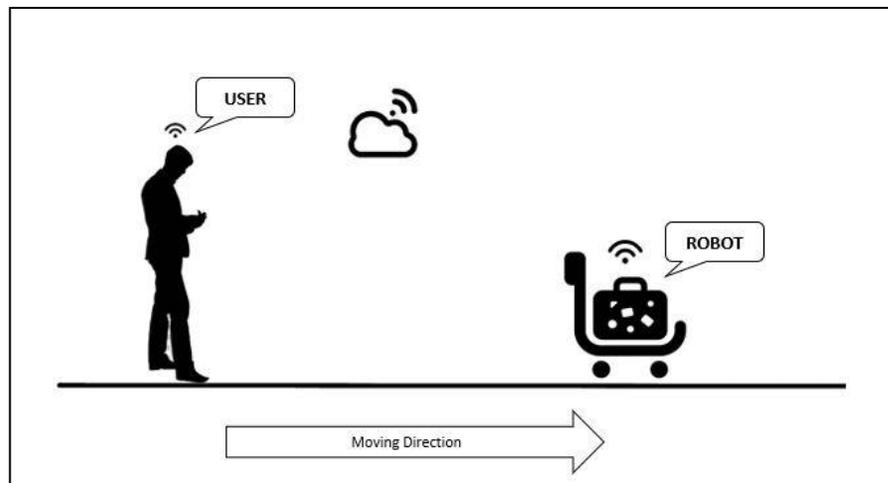
Sumber :Penulis, 2019

Dalam sistem ini, pemilik barang akan mengetahui dimana posisi *robot* pembawa barangnya melalui *smartphone* miliknya. Jika *smartphone* meminta koordinat posisi *robot*, maka mikrokontroler yang berada di *robot* akan membaca

GPS untuk mendapatkan informasi koordinat posisi dan selanjutnya mengirim informasi tersebut ke *smartphone* melalui *internet* dengan menggunakan modul WiFi (ESP 8266).

Selain itu, jika pemilik barang tidak mau mendorong *robot* pembawa barangnya, maka orang tersebut dapat menggunakan *smartphone* miliknya untuk menggerakkan *robot* tersebut. *Smartphone* dapat mengirim perintah maju, mundur, belok kiri, belok kanan, ataupun berhenti kepada *robot* yang sudah dilengkapi dengan mikrokontroler dan motor DC sebagai penggerak roda. Perintah tersebut dikirim melalui *internet* dan diterima oleh mikrokontroler melalui modul WiFi (ESP 8266).

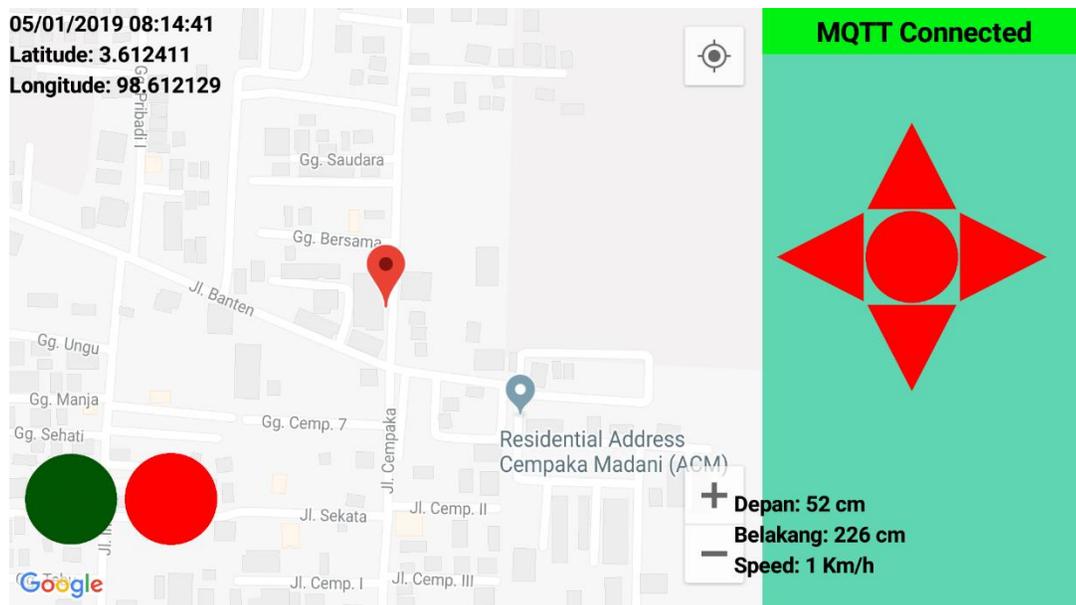
Sistem *robot* juga dilengkapi dengan *sensor* ultrasonik yang dipasang di bagian depan dan belakang *robot* sehingga mikrokontroler dapat mengetahui jarak *robot* terhadap halangan di depan maupun di belakangnya. Informasi tersebut juga dikirim ke *smartphone*.



Gambar 3.4 Penerapan sistem kontrol robot pembawa barang menggunakan android berbasis Arduino IOT (internet Of Thing)

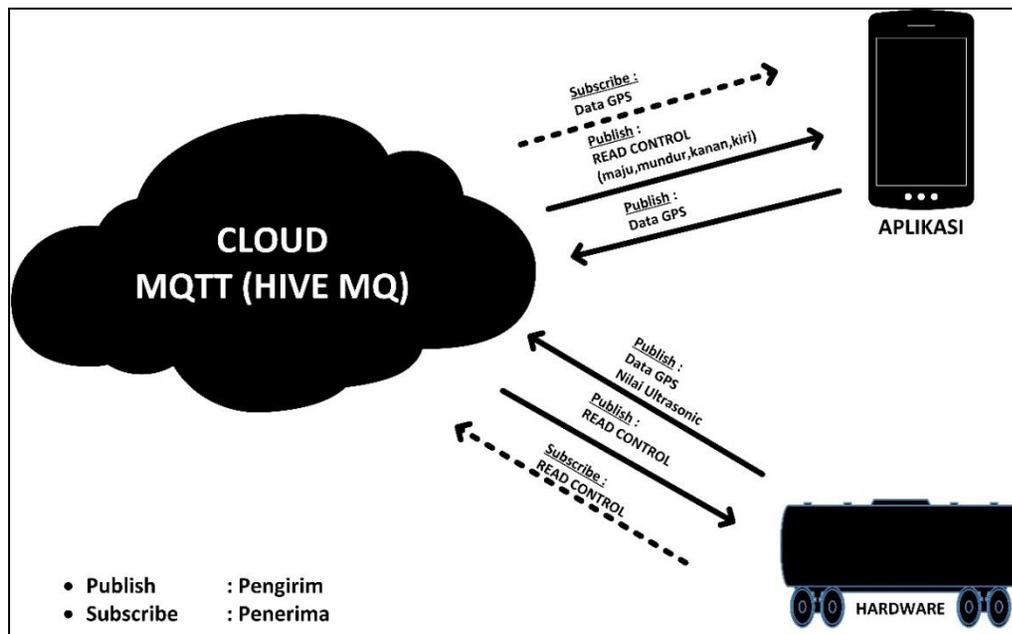
Sumber : Penulis, 2019

Pada tampilan di *smartphone*, terdapat 5 tombol yang fungsinya untuk menggerakkan roda (maju, mundur, kanan, kiri, dan stop). Alat yang telah berjalan akan menampilkan lokasi robot secara *real time* pada aplikasi Android serta menampilkan informasi tambahan berupa *latitude* dan *longitude*, jarak depan belakang, serta kecepatan robot yang sedang berjalan. Dua tombol (merah dan hijau) yang terdapat di aplikasi dapat digunakan untuk mengaktifkan dan mematikan alarm (buzzer on/off) yang menandakan troli tersebut hilang dari jangkauan pemantauan. Gambar di bawah ini merupakan tampilan dari aplikasi tersebut



Gambar 3.5 Tampilan Aplikasi Kendali
Sumber : Penulis, 2019

Komunikasi antara *smartphone* dan perangkat robot menggunakan konsep komunikasi *publish subscribe* MQTT, seperti gambar di bawah ini.



Gambar 3.6 Sistem komunikasi MQTT

Sumber :Penulis, 2019

Dalam protokol MQTT, terdapat *broker* sebagai pusat pertukaran informasi antara *publisher* dan *subscriber*. Istilah lain *broker* adalah *MQTT server*. Dalam sistem ini, perangkat di robot adalah sebagai *publisher*, sedangkan smartphone adalah sebagai *subscriber* dan informasi yang dihasilkan oleh perangkat robot sebagai *topic*. *Topic* tersebut dikirimkan oleh *publisher* kepada *broker*. Untuk mendapatkan informasi yang dikirimkan oleh *publisher*, maka *subscriber* harus melakukan *request* atau *subscribe* kepada *broker* dengan menyebutkan *topic* yang diinginkan. Informasi dikirim oleh *broker* sesuai dengan *topic* yang diminta oleh *subscriber*.

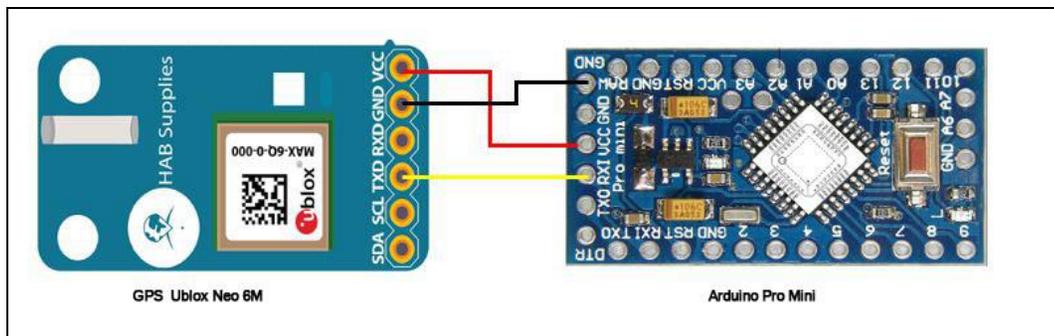
3.4 Perancangan *Hardware*

Perancangan *hardware* ini meliputi perancangan interkoneksi mikrokontroler Arduino Pro Mini terhadap modul GPS, modul sensor ultrasonik, modul WiFi (ESP 8266), dan motor DC melalui driver motor. Selain itu, dirancang juga interkoneksi

dari modul DC to DC Converter LM2956 terhadap Motor DC penggerak roda dan modul-modul yang memerlukan.

3.4.1 Interkoneksi Mikrokontroler terhadap GPS

Berikut ini adalah gambar interkoneksi antara mikrokontroler Arduino Pro Mini terhadap GPS Ublox Neo 6M. Tampak bahwa modul GPS mendapatkan tegangan Vcc dan GND dari mikrokontroler Arduino Pro Mini. Sedangkan data output GPS dikirimkan kepada Arduino Pro Mini melalui pin TXD (GPS) dan pin RX1 (Arduino Pro Mini).



Gambar 3.7 Interkoneksi Mikrokontroler Arduino Pro Mini terhadap GPS Ublox Neo 6M

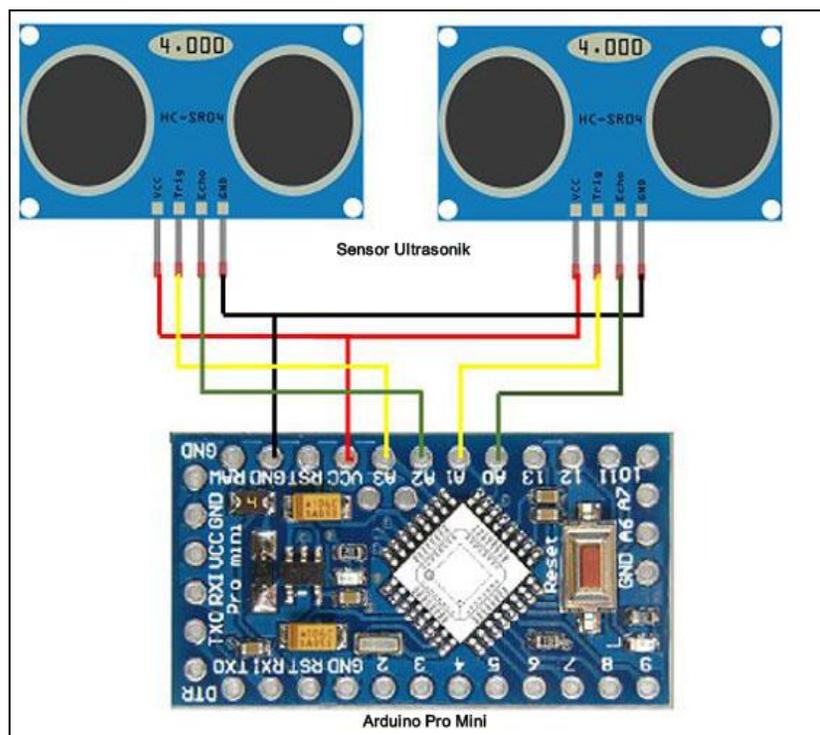
Penulis, 2019

3.4.2 Interkoneksi Mikrokontroler terhadap Sensor Ultrasonik

Berikut ini adalah gambar interkoneksi antara mikrokontroler Arduino Pro Mini terhadap modul sensor ultrasonik HC-SR04. Tampak bahwa 2 (dua) buah modul sensor ultrasonik tersebut mendapat tegangan Vcc dan GND dari modul Arduino Pro Mini.

Agar dapat bekerja, sinyal perintah kerja dari mikrokontroler kepada modul sensor ultrasonik-1 diberikan melalui pin A3 (Arduino Pro Mini) dan pin Triger

(Sensor Ultrasonik-1), sedangkan sinyal perintah kerja dari mikrokontroler kepada modul sensor ultrasonik-2 diberikan melalui pin A1 (Arduino Pro Mini) dan pin Triger (Sensor Ultrasonik-2). Selanjutnya *response* sensor ultrasonik-1 diberikan kepada mikrokontroler melalui pin Echo (Sensor Ultrasonik-1) dan pin A2 (Arduino Pro Mini), sedangkan *response* sensor ultrasonik-2 diberikan kepada mikrokontroler melalui pin Echo (Sensor Ultrasonik-2) dan pin A0 (Arduino Pro Mini).



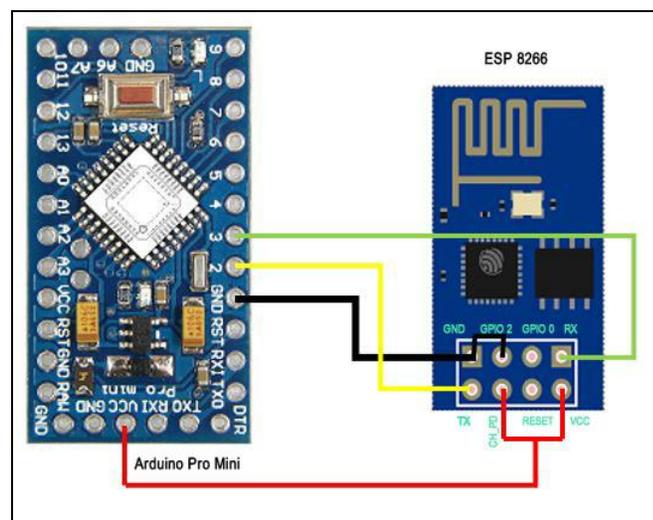
Gambar 3.8 Interkoneksi Mikrokontroler Arduino Pro Mini terhadap Dua Buah Modul Sensor Ultrasonik HC-SR04

Sumber : Penulis, 2019

3.4.3 Interkoneksi Mikrokontroler terhadap Modul WiFi (ESP 8266)

Berikut ini adalah gambar interkoneksi antara mikrokontroler Arduino Pro Mini terhadap Modul WiFi (ESP 8266). Tampak bahwa Wifi (ESP 8266) tersebut mendapat tegangan Vcc dan GND dari modul Arduino Pro Mini. Selain itu, pada modul WiFi ESP 8266, pin CH_PD (*chip enable*) diaktifkan dengan cara memberi logika *high* sehingga dihubungkan dengan tegangan Vcc (5 Volt). Demikian halnya dengan pin GPIO_2 diaktifkan dengan memberi logika *low* sehingga dihubungkan dengan GND.

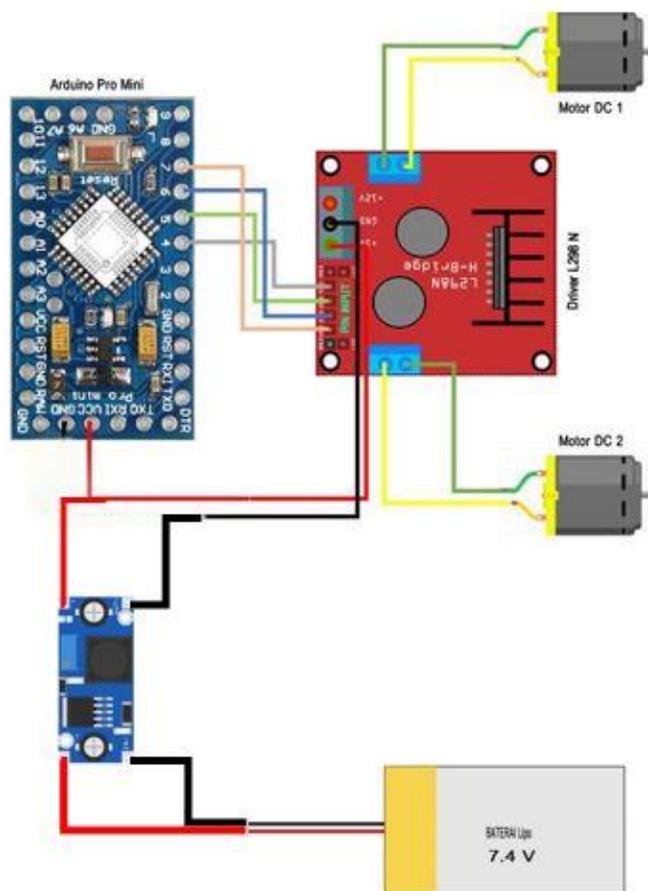
Untuk keperluan transfer data antara WiFi ESP8266 dan Arduino Pro Mini, dilakukan melalui pin 2 (Arduino Pro Mini) yang dihubungkan ke pin TX (ESP 8266), serta melalui pin 3 (Arduino Pro Mini) yang dihubungkan ke pin RX (ESP 8266).



Gambar 3.9 Interkoneksi Mikrokontroler Arduino Pro Mini terhadap Modul WiFi (ESP 8266)
Penulis, 2019

3.4.4 Interkoneksi Mikrokontroler terhadap Driver Motor L298N dan Baterai LiPo

Berikut ini adalah gambar interkoneksi antara mikrokontroler Arduino Pro Mini terhadap Motor DC melalui modul driver motor L298N. Tampak bahwa pin 4,5,6,7 (Arduino Pro Mini) terhubung dengan pin input driver motor (L298N). Pin 4,5,6,7 tersebut sebagai pengontrol atau pengendali arah putaran Motor DC.



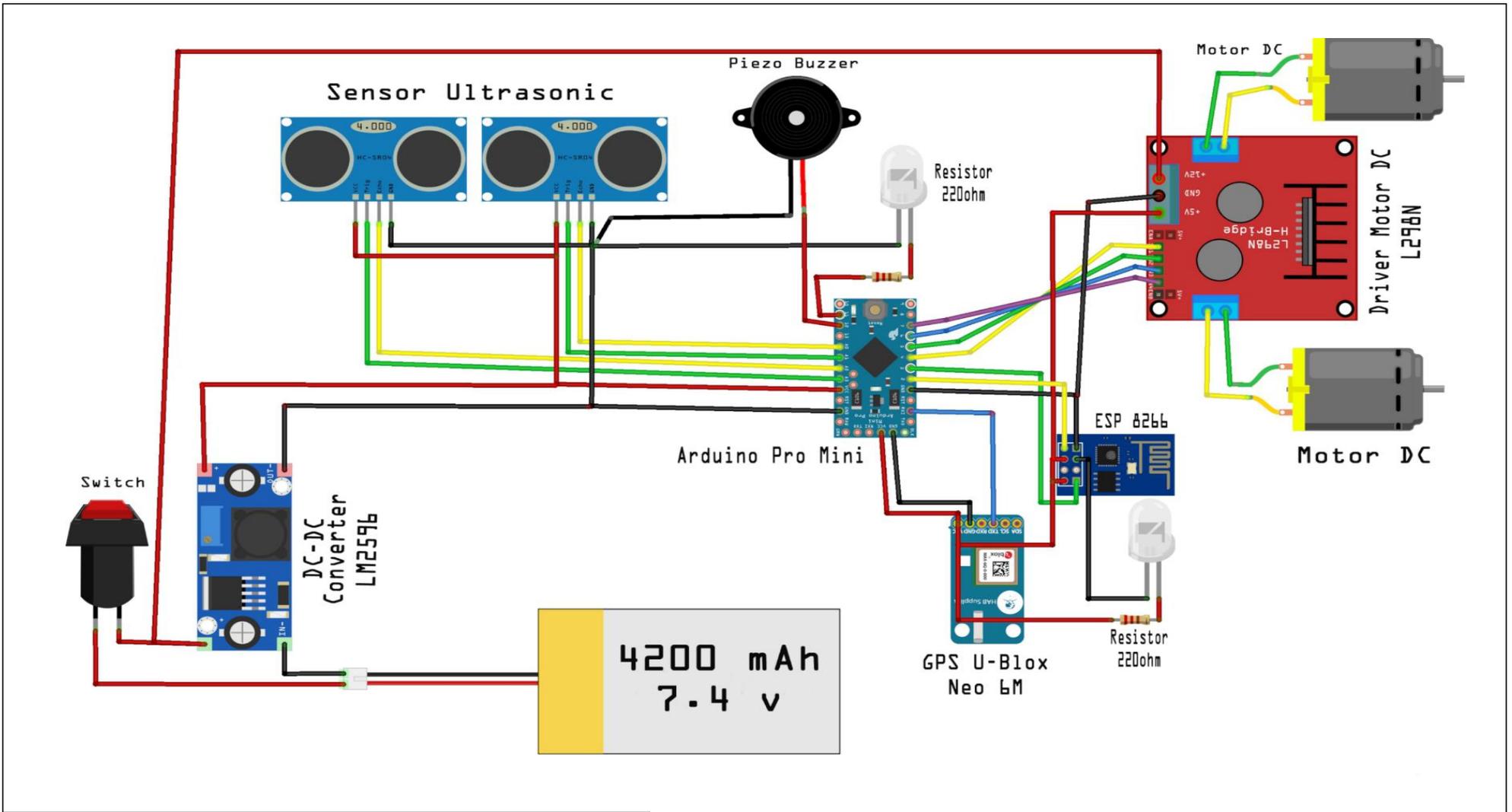
Gambar 3.10 Interkoneksi Mikrokontroler Arduino Pro Mini terhadap Driver Motor (L298N) dan Baterai LiPo
Penulis, 2019

Agar motor dapat bekerja, sumber dari baterai langsung menuju ke motor, karena motor memerlukan arus yang besar untuk dapat berputar sedangkan Arduino hanya mampu memasok arus kecil.

Dalam gambar tampak bahwa terminal positif baterai LiPo terhubung ke pin 12V (modul Driver Motor L298N) setelah melalui switch untuk mengatur on/off. Sedangkan terminal negatif baterai LiPo terhubung ke pin GND (modul Driver Motor L298N). Baterai LiPo tersebut mempunyai kapasitas supply arus 4200 mAH.

3.4.5 Rangkaian Keseluruhan

Berikut ini adalah gambar interkoneksi keseluruhan antara mikrokontroler terhadap seluruh modul, dan interkoneksi antara Baterai LiPo terhadap Modul Driver Motor L298N, serta modul Driver L298N terhadap Motor DC penggerak roda.



LAYOUT RANGKAIAN



Universitas Pembangunan Panca Budi

SANDI ANUGRAHA

SCALE 1 : 500

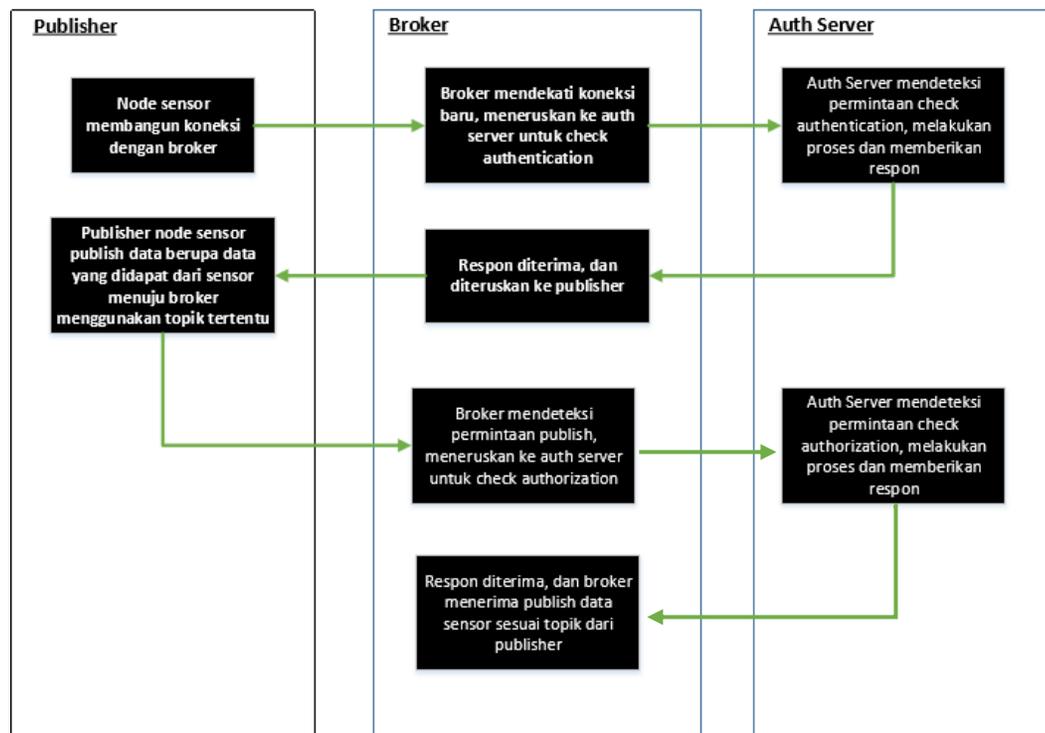


A4

Gambar 3.11 Rangkaian Keseluruhan
Penulis, 2019

3.5 Perancangan Software

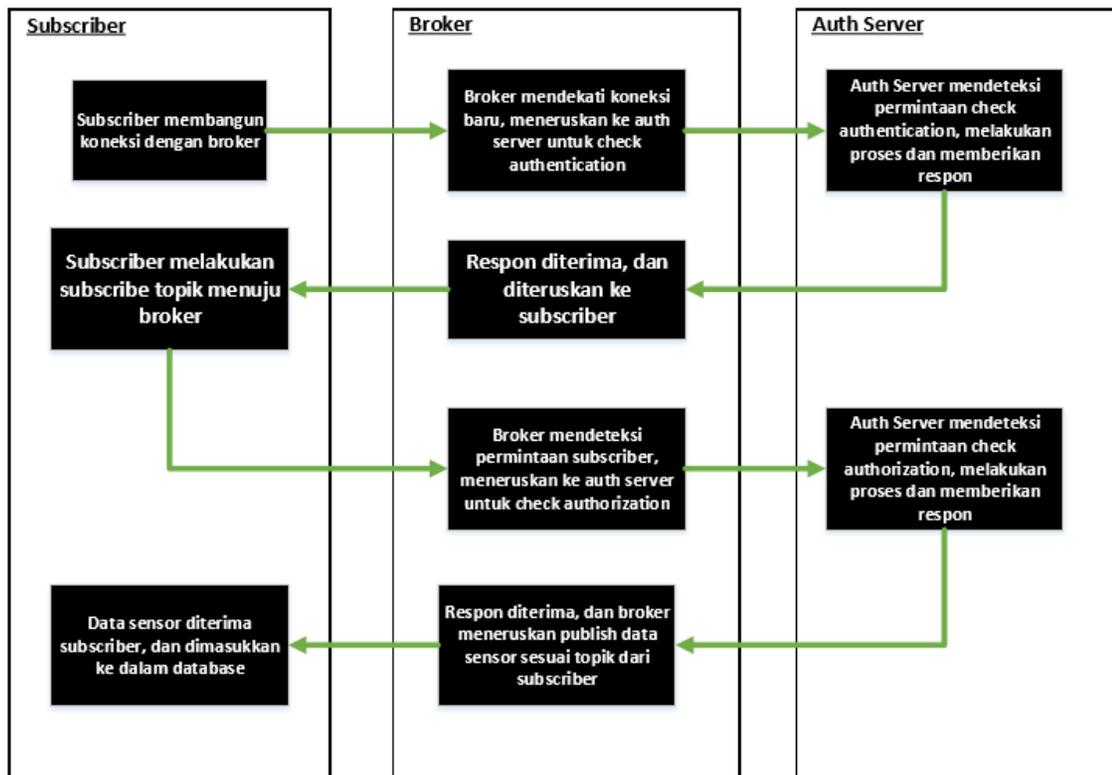
Setelah *power* dihidupkan dan aplikasi dinyalakan, mikrokontroler membaca sensor dan GPS secara *real time*, kemudian sensor dan data GPS yang terbaca dikirim ke *cloud* (MQTT). Disitulah proses *publisher* dan *subscriber* bekerja, dimana fungsi *publisher* sebagai pengirim informasi (data sensor) yang telah didapat dari *hardware* sedangkan *subscriber* adalah penerima pesan dari *publisher*. Berikut merupakan alur kerja penanganan *publisher* dan *subscriber*.



Gambar 3.12 Alur Kerja Penanganan Permintaan *Publisher*
 Sumber : Penulis, 2019

Pada Gambar 3.13 menunjukkan alur penanganan permintaan *publisher*. Dalam sistem ini, operasi dimulai dari mikrokontroler yang akan melakukan *publish* data. Oleh karena itu mikrokontroler yang bertindak sebagai *publisher*

diharuskan untuk membangun koneksi terlebih dahulu. *Broker* mendeteksi adanya permintaan koneksi dari *publisher* tersebut, dan kemudian data permintaan koneksi yang dikirimkan oleh *publisher* akan diambil dan dikirim menuju *auth server* untuk dilakukan *check authentication*. *Auth server* menerima permintaan tersebut, kemudian memproses apakah *publisher* dapat diijinkan untuk melakukan koneksi. Hasil dari *check authentication* tersebut dikirimkan kembali menuju *broker*, dan *broker* mengirim ke *publisher*. Jika *publisher* diijinkan untuk melakukan koneksi terhadap *broker*, maka tahap selanjutnya *publisher* akan melakukan *publish* data dengan topik yang telah ditentukan oleh *publisher*. Permintaan melakukan *publish* data akan dikirimkan terlebih dahulu oleh *publisher* menuju *broker*, dan *broker* akan mengirim data *publish* tersebut ke *auth server* untuk dilakukan *check authorization*. Pada *auth server* melakukan *check authorization* dan hasil dari pemeriksaan tersebut akan bertindak sebagai respon dari *auth server* kepada *broker*. *Broker* menerima respon dari *auth server* tentang *check authorization*, jika respon tersebut bernilai positif, maka data *publish* akan disimpan sementara oleh *broker* yang nantinya jika ada *subscriber* dengan topik yang sama melakukan *subscribe*, maka data tersebut akan diteruskan menuju *subscriber* tersebut. Jika respon tersebut bernilai negatif, maka *broker* tidak akan menyimpan data *publish* tersebut dan tidak akan meneruskannya menuju *subscriber* meskipun dengan topik yang sama. Selanjutnya untuk mengetahui alur kerja penanganan dari sisi *subscribe*, berikut merupakan alur kerjanya.



Gambar 3.13 Alur Kerja Penanganan Permintaan *Subscriber*
 Sumber : Penulis, 2019

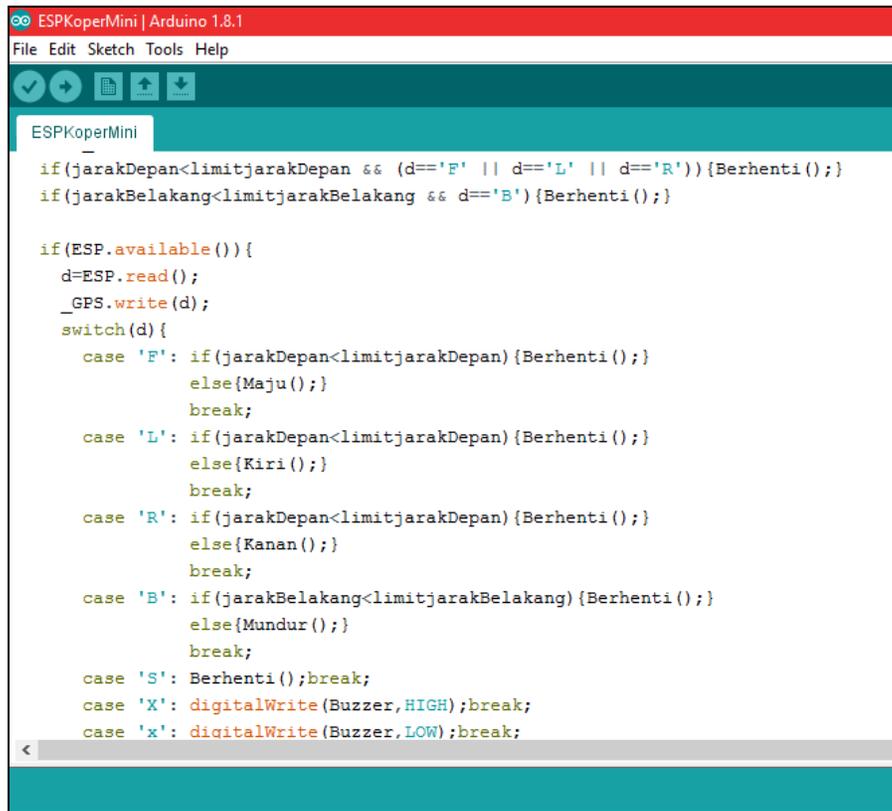
Pada Gambar 3.14 menunjukkan alur penanganan proses operasi yang dimulai dari *subscriber* yang akan melakukan *subscribe* pada topik tertentu. *Subscriber* diharuskan untuk membangun koneksi terlebih dahulu. *Broker* mendeteksi adanya permintaan koneksi dari *subscriber* tersebut, dan kemudian data permintaan koneksi yang dikirimkan oleh *subscriber* akan diambil dan dikirim menuju *auth server* untuk dilakukan *check authentication*. *Auth server* menerima permintaan tersebut, kemudian memproses menggunakan data yang dikirim oleh *broker*, apakah *subscriber* tersebut dapat melakukan koneksi. Hasil dari *check authentication* tersebut kemudian dikirimkan kembali menuju *broker*, dan *broker* mengirim ke

publisher. Jika *subscriber* diijinkan untuk melakukan koneksi terhadap *broker*, maka tahap selanjutnya *subscriber* akan melakukan *subscribe* pada topik yang telah ditentukan oleh *subscriber*. Permintaan melakukan *subscribe* pada topik tertentu akan dikirimkan terlebih dahulu oleh *subscriber* menuju *broker*.

Agar sistem ini bekerja, selain software yang terdapat di dalam smartphone, diperlukan juga software mikrokontroler Arduino Pro Mini yang akan membaca informasi yang diberikan oleh GPS dan Sensor Ultrasonik dan kemudian mengirimkan informasi tersebut ke *Smartphone* melalui internet dengan menggunakan modul WiFi (ESP 8266).

Selain itu, mikrokontroler selalu memeriksa apakah ada perintah dari Smartphone untuk menggerakkan roda robot ke arah maju, mundur, belok kiri, belok kanan, ataukah berhenti. Mikrokontroler juga selalu memeriksa informasi yang diberikan oleh modul ultrasonik dan memprosesnya sehingga dapat diketahui jarak robot terhadap halangan di depannya. Jika terdeteksi jarak terhadap halangan lebih kecil dari 50 cm, maka buzzer mengeluarkan suara sebagai notifikasi dan mikrokontroler akan menghentikan motor DC penggerak roda.

Berikut ini adalah bagian dari program untuk menggerakkan roda jika jarak prototype ke halangan sudah lebih kecil dari 50 cm.



```

ESPKoperMini | Arduino 1.8.1
File Edit Sketch Tools Help

ESPKoperMini
if(jarakDepan<limitjarakDepan && (d=='F' || d=='L' || d=='R')){Berhenti();}
if(jarakBelakang<limitjarakBelakang && d=='B'){Berhenti();}

if(ESP.available()){
  d=ESP.read();
  _GPS.write(d);
  switch(d){
    case 'F': if(jarakDepan<limitjarakDepan){Berhenti();}
              else{Maju();}
              break;
    case 'L': if(jarakDepan<limitjarakDepan){Berhenti();}
              else{Kiri();}
              break;
    case 'R': if(jarakDepan<limitjarakDepan){Berhenti();}
              else{Kanan();}
              break;
    case 'B': if(jarakBelakang<limitjarakBelakang){Berhenti();}
              else{Mundur();}
              break;
    case 'S': Berhenti();break;
    case 'X': digitalWrite(Buzzer,HIGH);break;
    case 'x': digitalWrite(Buzzer,LOW);break;
  }
}

```

Gambar 3.14 Tampilan *coding* penggerak motor dc
Sumber : Penulis, 2019

Berikut ini adalah bagian dari software di Smartphone berbasis Android untuk menampilkan *interface map*. Pada tampilan tersebut, *GPS receiver* menerima sinyal dari satelit GPS dan melempar balik sinyal *acknowledge* (ACK) ke satelit-satelit GPS. Berikut ini adalah bagian dari program di *smartphone* untuk menampilkan data *langitude* dan *longitude*-nya.

```

ESP8266Mini
String _Lat="";
String _NS=data.substring(batas3+1,batas4);/*mencari lintang*/
String _Lon="";
String _EW=data.substring(batas5+1,batas6);/*mencari bujur*/
String _Speed=data.substring(batas6+1,batas7);

if(_NS=="N") {_Lat=data.substring(batas2+1,batas3);}else{_Lat="-" + data.substring(batas2+1,batas3);}
if(_EW=="W") {_Lon="-" + data.substring(batas4+1,batas5);}else{_Lon=data.substring(batas4+1,batas5);}
Speed = _Speed.toFloat() * 1.852;/*nilai speed GPS satuannya not agar jadi km/jam dikali 1.852*/
Lat=GPSToDec(_Lat);/*merubah angka menjadi decimal untuk untuk latitude*/
Lon=GPSToDec(_Lon);
}
else{
  stateGPS=0;
  digitalWrite(LED_GPS,LOW); //Led GPS Mati
}
data="";
data="DataGPS;" + String(Lat,6) + "," + String(Lon,6) + "," + String(Speed) + "," + dd + "/" + MM + "/" + YY + " " +
String(H) + ":" + _Time.substring(2,4) + ":" + _Time.substring(4,6) + "," +
String(jarakDepan) + "," + String(jarakBelakang);
return data;
}
<

```

Gambar 3.15 Tampilan coding GPS

Sumber : Penulis, 2019

Keluaran GPS berupa kalimat laporan sesuai dengan format Standar NMEA (*National Marine Electronics Association*), yang berisi diantaranya adalah koordinat lintang (*latitude*), bujur (*longitude*), ketinggian (*altitude*), waktu sekarang standar UTC (*Universal Time*), dan kecepatan (*speed*). Berikut ini adalah format kalimat keluaran GPS.

\$GPRMC

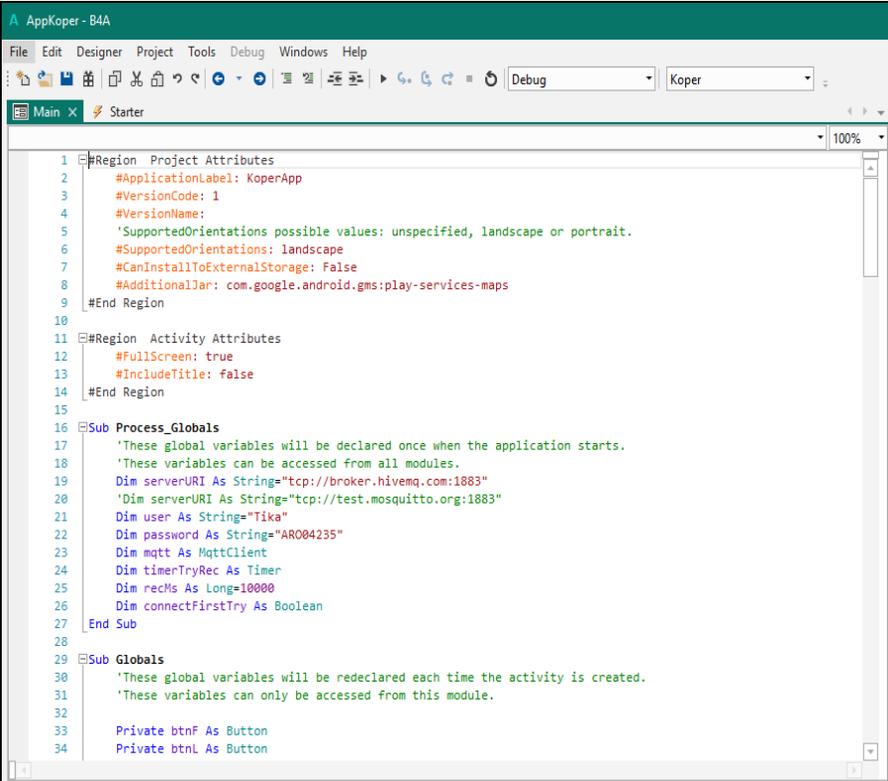
Contoh: \$GPRMC,092204.999,A,4250.5589,S,14718.5084,E,0.00,89.68,211200,,*25

Field	Contoh isi	Deskripsi
Sentence ID	\$GPRMC	
UTC Time	092204.999	hhmmss.sss
Status	A	A = Valid, V = Invalid
Latitude	4250.5589	ddmm.mmmm
N/S Indicator	S	N = North, S = South
Longitude	14718.5084	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Speed over ground	0.00	Knots
Course over ground	0.00	Degrees
UTC Date	211200	DDMMYY
Magnetic variation		Degrees
Checksum	*25	
Terminator	CR/LF	

Gambar 3.16 Standart NMEA – GPRMC [21]

Penulis, 2019

Satuan nilai yang dihasilkan dari *speed* adalah nots, oleh sebab itu agar menjadi km/jam di kalikan 1.852. Hal terpenting lainnya dalam pembuatan *software* adalah pembuatan aplikasi di Android, dimana *interface*-nya dibuat dengan menggunakan *software* B4A. Tampilan yang dibuat meliputi tombol (maju, mundur, kanan, kiri dan stop), tampilan Google *map*, tampilan waktu, *longitude* dan *longitude* serta tombol buzzer on/off. Berikut merupakan gambaran dari B4A.



```

1 #Region Project Attributes
2 #ApplicationLabel: KoperApp
3 #VersionCode: 1
4 #VersionName:
5 'SupportedOrientations possible values: unspecified, landscape or portrait.
6 #SupportedOrientations: landscape
7 #CanInstallToExternalStorage: False
8 #AdditionalJar: com.google.android.gms:play-services-maps
9 #End Region
10
11 #Region Activity Attributes
12 #FullScreen: true
13 #IncludeTitle: false
14 #End Region
15
16 #Sub Process_globals
17 'These global variables will be declared once when the application starts.
18 'These variables can be accessed from all modules.
19 Dim serverURI As String="tcp://broker.hivemq.com:1883"
20 'Dim serverURI As String="tcp://test.mosquitto.org:1883"
21 Dim user As String="Tika"
22 Dim password As String="AR004235"
23 Dim mqtt As MqttClient
24 Dim timerTryRec As Timer
25 Dim recMs As Long=10000
26 Dim connectFirstTry As Boolean
27 End Sub
28
29 #Sub Globals
30 'These global variables will be redeclared each time the activity is created.
31 'These variables can only be accessed from this module.
32
33 Private btnF As Button
34 Private btnL As Button

```

Gambar 3.17 Tampilan B4A
Penulis, 2019

BAB 4

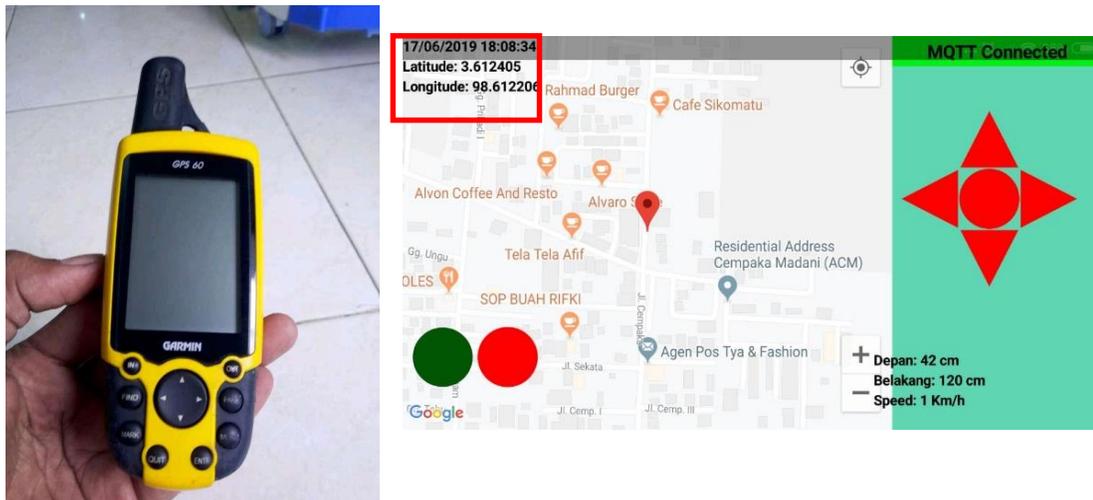
PENGUJIAN DAN PEMBAHASAN

4.1 Pengujian

Setelah berhasil melakukan implementasi perlu dilakukan pengujian untuk melihat kehandalan dan kecepatan sistem yang telah dibuat, berikut ini merupakan pengujian dari pembacaan GPS, sensor ultrasonic dan delay terhadap kendali sistem kontrol troli pembawa barang.

4.1.1 Pengujian Akurasi Pembacaan GPS

Berikut ini adalah hasil pengujian pembacaan GPS yang dilakukan dengan 3 kali *sample* pada lokasi dan medan landasan yang berbeda. Proses pengujian dilakukan untuk mengetahui keakuratan data GPS dengan membandingkan hasil yang ada di aplikasi *smartphone* dengan pembacaan pada modul GPS Garmin. Adapun data hasil pengujian keakurasian GPS tampak pada tabel berikut ini.



Gambar 4.1 Membandingkan data GPS pada prototype alat dengan GPS Garmin
(a) GPS Garmin (b) GPS pada aplikasi android

Sumber :Penulis, 2019

Percobaan dilakukan di 3 lokasi yang berbeda dengan *robot* yang berjalan dan di catat sebanyak 6 kali titik koordinat GPS. Koordinat GPS yang diambil ialah membandingkan antara GPS (Garmin) dengan GPS yang ada pada aplikasi *Android* tugas akhir (GPS App *Robot*).

1. Percobaan Pertama (Halaman Depan Kantor)

Tabel 4.1 Hasil Pengujian Akurasi Pembacaan GPS

No	Lokasi	Latitude		Penyimpangan	No	Lokasi	Longitude		Penyimpangan
		GPS Aplikasi	GPS Garmin				GPS Aplikasi	GPS Garmin	
1	Halaman Depan Kantor	3.579920	3.579857	0,18%	1	Halaman Depan Kantor	98.671988	98.671008	0,10%
2		3.579918	3.579858	0,17%	2		98.672899	98.671002	0,19%
3		3.579902	3.579865	0,10%	3		98.671501	98.670012	0,15%
4		3.579920	3.579886	0,09%	4		98.673003	98.671065	0,20%
5		3.579848	3.579820	0,08%	5		98.671854	98.671008	0,09%
6		3.579916	3.579890	0,07%	6		98.671987	98.671020	0,10%
Average Error				0,12%	Average Error				0,14%

Sumber :Penulis, 2019

Percobaan **Pertama** dilakukan di Halaman Depan Kantor, dengan perbedaan posisi *robot* dan dicatat titik koordinat sebanyak 6 kali dengan hasil rata – rata penyimpangan (*error*) antar kedua GPS yaitu sebesar 0.12% (*latitude*) 0.14% (*longitude*).

2. Percobaan Kedua (Parkiran Mobil Kantor)

Tabel 4.2 Hasil Pengujian Akurasi Pembacaan GPS

No	Lokasi	Latitude		Penyimpangan	No	Lokasi	Longitude		Penyimpangan
		GPS Aplikasi	GPS Garmin				GPS Aplikasi	GPS Garmin	
1	Parkiran Mobil Kantor	3.579907	3.579854	0,15%	1	Parkiran Mobil Kantor	98.671988	98.671004	0,10%
2		3.579905	3.579848	0,16%	2		98.672599	98.671002	0,16%
3		3.579912	3.579855	0,16%	3		98.671544	98.670012	0,16%
4		3.579921	3.579866	0,15%	4		98.672003	98.671066	0,09%
5		3.579845	3.579825	0,06%	5		98.673454	98.672008	0,15%
6		3.579905	3.579860	0,13%	6		98.672553	98.671010	0,16%
Average Error				0,13%	Average Error				0,14%

Penulis, 2019

Percobaan Kedua dilakukan di Parkiran Mobil Kantor, dengan perbedaan posisi robot dan dicatat titik koordinat sebanyak 6 kali dengan hasil rata – rata penyimpangan (*error*) antar kedua GPS yaitu sebesar 0.13% (*latitude*) 0.14% (*longitude*).

3. Ketiga (Lobby Kantor)

Tabel 4.3 Hasil Pengujian Akurasi Pembacaan GPS

No	Lokasi	Latitude		Penyimpangan
		GPS Aplikasi	GPS Garmin	
1	Lobby Kantor	3.579864	3.579807	0,16%
2		3.579856	3.579808	0,13%
3		3.579844	3.579815	0,08%
4		3.579859	3.579817	0,12%
5		3.579863	3.579820	0,12%
6		3.579875	3.579811	0,18%
Average Error				0,13%

No	Lokasi	Longitude		Penyimpangan
		GPS Aplikasi	GPS Garmin	
1	Lobby Kantor	98.671988	98.671204	0,08%
2		98.672589	98.671102	0,15%
3		98.672544	98.671112	0,15%
4		98.672503	98.671066	0,15%
5		98.673354	98.672008	0,14%
6		98.673213	98.672010	0,12%
Average Error				0,13%

Sumber :Penulis, 2019

Percobaan Ketiga dilakukan di dalam Ruangan Lobby Kantor, dengan perbedaan posisi *robot* dan dicatat titik koordinat sebanyak 6 kali dengan hasil rata – rata penyimpangan (*error*) antar kedua GPS yaitu sebesar 0.13% (*latitude*) 0.13% (*longitude*).

Hasil perhitungan penyimpangan (*error*) di peroleh dari rumus berikut :

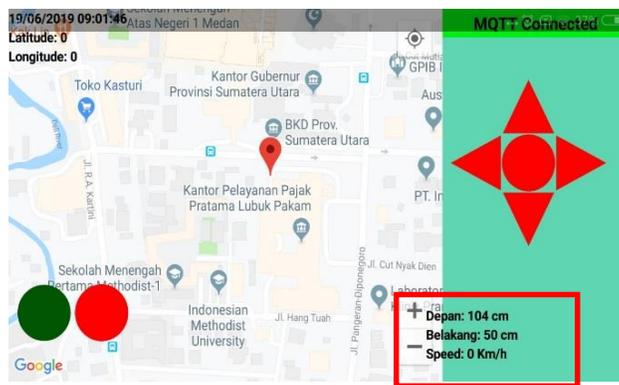
$$Error \text{ Koordinat GPS} = \frac{[(X_{asli} - X_{GPS}) * 100]}{X_{asli}} = \dots \%$$

dimana, X_{asli} = GPS Garmin, dan X_{GPS} = GPS App *Robot*

Berdasarkan ketiga hasil percobaan di atas, terlihat penyimpangan terbesar yaitu 0.20 % dan terkecil yaitu 0.06 %, dengan total rata-rata penyimpangan (*error*) adalah 0.13 – 0.14 %. Hal tersebut membuktikan bahwa koordinat GPS pada aplikasi yang menggunakan GPS Ublox Neo 6M menghasilkan akurasi penyimpangan (*error*) yang sama dengan *datasheet* GPS Garmin sebesar 0.138%.

4.1.2 Pengujian Akurasi Pembacaan Jarak Dengan Sensor Ultrasonik

Berikut ini adalah tabel hasil pengujian fungsi sensor ultrasonik terhadap halangan. Dalam pengujian ini, apabila *smartphone* memberikan *trigger* (maju, mundur, kanan, kiri) dan selanjutnya terdeteksi adanya halangan, maka *Buzzer* akan berbunyi dan roda akan berhenti. Pada saat itulah dilakukan pengukuran jarak posisi roda terakhir terhadap posisi awal. Selanjutnya penyimpangan jarak riil terhadap nilai jarak yang terdapat di Aplikasi Smartphone dihitung nilai rata-ratanya.



(a)



(b)

Gambar 4.2 Membandingkan Respond Buzzer pada aplikasi dan
Sumber : Penulis, 2019

1. Percobaan Pertama (Conblox Besar)

Tabel 4.4 Hasil Pengujian Akurasi Pembacaan Jarak dengan Sensor Ultrasonic

No.	Medan Landasan	Trigger	Respond Buzzer saat ada halangan (ON/OFF)	Jarak Aplikasi (cm)	Jarak Real (cm)	Penyimpangan
1	Conblox Besar	Maju	ON	15	15	0%
2		Mundur	ON	29	25	14%
3		Kanan	ON	27	24	11%
4		Kiri	ON	16	15	6%
<i>Average Error</i>						8%

Sumber : Penulis, 2019

Pada Percobaan pertama dilakukan di area conblok besar. Pada saat *Smartphone* memberi *trigger* maju kepada *robot* dan pada saat itu juga mendeteksi adanya halangan, tampak bahwa pada tampilan aplikasi *smartphone* mendeteksi jarak 15 cm, sedangkan pada pengukuran jarak *real* juga menghasilkan 15 cm. Ini membuktikan bahwa pembacaan jarak oleh Aplikasi tidak adanya perbedaan dengan jarak *real*-nya.

Pada saat *Smartphone* memberi *trigger* mundur kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 29 cm, sedangkan *real* nya 25 cm, sehingga terdapat perbedaan 4 cm atau perbedaan pembacaan sebesar 14%.

Pada saat *Smartphone* memberi *trigger* kanan kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 27 cm, sedangkan *real* nya 24 cm, sehingga terdapat perbedaan 3 cm atau perbedaan pembacaan sebesar 11%.

Pada saat *Smartphone* memberi *trigger* Kiri kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 16 cm, sedangkan *real* nya 15 cm, sehingga terdapat perbedaan 1 cm atau perbedaan pembacaan sebesar 6%.

Dari keempat percobaan tersebut di atas ambil rata-rata sebesar 8 % untuk pembacaan *error* nya pada kondisi jalan conblok besar.

2. Percobaan Kedua (Aspal)

Tabel 4.5 Hasil Pengujian Akurasi Pembacaan Jarak dengan Sensor Ultrasonic

No.	Medan Landasan	Trigger	Respond Buzzer saat ada halangan (ON/OFF)	Jarak Aplikasi (cm)	Jarak Real (cm)	Penyimpangan
1	Aspal	Maju	ON	21	18	14%
2		Mundur	ON	7	7	0%
3		Kanan	ON	12	11	8%
4		Kiri	ON	49	46	6%
<i>Average Error</i>						7%

Sumber : Penulis, 2019

Dilakukan lagi percobaan Kedua pada kondisi jalan Aspal ketika *robot* diberikan *trigger* maju dari *smartphone*, pembacaan pada aplikasi mendeteksi jarak 21 cm, sedangkan jarak *real*-nya 18 cm, sehingga terdapat perbedaan 3 cm atau perbedaan pembacaan sebesar 14%.

Pada saat *Smartphone* memberi *trigger* mundur kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 7 cm, sedangkan *real* nya 7 cm, sehingga terdapat perbedaan 0 cm atau perbedaan pembacaan sebesar 0%.

Pada saat *Smartphone* memberi *trigger* kanan kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 12 cm, sedangkan *real* nya 11 cm, sehingga terdapat perbedaan 1 cm atau perbedaan pembacaan sebesar 8%.

Pada saat *Smartphone* memberi *trigger* Kiri kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 49 cm, sedangkan *real* nya 46 cm, sehingga terdapat perbedaan 3 cm atau perbedaan pembacaan sebesar 6%.

Dari keempat percobaan tersebut di atas ambil rata-rata sebesar 7 % untuk pembacaan *error* nya pada kondisi jalan Aspal.

3. Percobaan Pertama (Lantai)

Tabel 4.6 Hasil Pengujian Akurasi Pembacaan Jarak dengan Sensor Ultrasonic

No.	Medan Landasan	Trigger	Respond Buzzer saat ada halangan (ON/OFF)	Jarak Aplikasi (cm)	Jarak Real (cm)	Penyimpangan
1	Lantai	Maju	ON	32	31	3%
2		Mundur	ON	29	27	7%
3		Kanan	ON	50	50	0%
4		Kiri	ON	27	25	7%
<i>Average Error</i>						4.25%

Sumber : Penulis, 2019

Percobaan terakhir untuk medan landasan lantai ketika robot diberikan *trigger* maju dari *smartphone*, tampak bahwa pembacaan pada Aplikasi mendeteksi jarak 32 cm, sedangkan jarak *real*-nya 31 cm, sehingga terdapat perbedaan 1cm atau perbedaan pembacaan sebesar 3%.

Pada saat *Smartphone* memberi *trigger* mundur kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 29 cm, sedangkan *real* nya 27 cm, sehingga terdapat perbedaan 2 cm atau perbedaan pembacaan sebesar 7%.

Pada saat *Smartphone* memberi *trigger* kanan kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 50 cm, sedangkan *real* nya 50 cm, sehingga terdapat perbedaan 0 cm atau perbedaan pembacaan sebesar 0%.

Pada saat *Smartphone* memberi *trigger* Kiri kepada *robot*, pembacaan pada aplikasi mendeteksi jarak 27 cm, sedangkan *real* nya 25 cm, sehingga terdapat perbedaan 2 cm atau perbedaan pembacaan sebesar 7%.

Dari hasil percobaan yang dilakukan di kondisi jalan lantai, rata-rata penyimpangan antara pembacaan jarak halangan yang dilakukan oleh aplikasi *sistem* terhadap jarak *real* yaitu sebesar 4.25%.

Dari ketiga percobaan di atas, dapat diambil kesimpulan bahwa medan landasan (lantai) berpengaruh terhadap *prototype* ini. Rata-rata penyimpangan (*error*) pembacaan jarak oleh Aplikasi Sensor Ultrasonik terhadap jarak *real* terkecil adalah 4.25%. Dari nilai rata-rata penyimpangan (*error*) yang didapat, menunjukkan adanya faktor eksternal yaitu faktor cahaya, banyak sedikitnya cahaya yang diterima oleh sensor Ultrasonik berpengaruh terhadap pembacaan jarak. Namun dari sekian banyak percobaan, juga tampak bahwa Buzzer sudah bekerja sesuai dengan rencana.

4.1.3 Pengujian Delay Sistem

Berikut ini adalah tabel hasil pengujian delay. Pengukuran dilakukan dengan mengukur selisih waktu antara datangnya perintah maju, mundur, kanan, kiri dan berhenti dari *Smartphone* terhadap waktu dari *robot* merespon perintah tersebut.

1. Percobaan Pertama (Ruang Belakang)

Tabel 4.7 Hasil Pengujian Delay

No	Medan Landasan	Lokasi	Trigger					Delay (s)
			Maju (F)	Mundur (B)	Kanan (R)	Kiri (L)	Berhenti (S)	
1	Conbox Besar	Halaman Belakang Rumah	√					1.28
				√				2.04
					√			1.5
						√		1.51
							√	1.17
<i>Average Error</i>							1.60	

Sumber : Penulis, 2019

Pada percobaan pertama yang dilakukan pada medan landasan conblox dilakukan percobaan dengan memberikan *trigger* (maju,mundur,kanan,kiri,berhenti) dari *smartphone*, *delay* yang terjadi saat menekan tombol di aplikasi *Android* hingga *robot* merespond yaitu pada rata-rata dibawah 1.6 *second*. hal ini masih dalam tahap wajar dan tentunya hal ini juga dipengaruhi kondisi sinyal dan koneksi dari Internet yang digunakan sehingga di dapatkan hasil tersebut.

2. Percobaan Kedua (Dalam Ruangan)

Tabel 4.8 Hasil Pengujian Delay

No	Medan Landasan	Lokasi	Trigger					Delay (s)
			Maju (F)	Mundur (B)	Kanan (R)	Kiri (L)	Berhenti (S)	
2	Lantai	Ruangan Tamu	√					1.43
				√				2.1
					√			1.5
						√		0.98
							√	1
<i>Average Error</i>							1.40	

Sumber : Penulis, 2019

Percobaan kedua yang dilakukan di ruang tamu dengan medan landasan keramik, percobaan dengan memberikan *trigger* (maju,mundur,kanan,kiri,berhenti) dari *smartphone*, *delay* yang terjadi saat menekan tombol di aplikasi *Android* hingga *robot* merespond yaitu pada rata-rata 1.4 *second*. Percobaan kedua ini lebih baik dibandingkan dengan percobaan pertama dengan melihat pada kondisi sinyal dan koneksi Internet yang digunakan oleh robot yaitu sinyal dan koneksi lebih baik pada saat berada di ruang tamu sehingga menghasilkan *delay* yang lebih kecil.

3. Percobaan Ketiga (Luar Ruangan)

Tabel 4.9 Hasil Pengujian Delay

No	Medan Landasan	Lokasi	Trigger					Delay (s)
			Maju (F)	Mundur (B)	Kanan (R)	Kiri (L)	Berhenti (S)	
3	Aspal	Depan Rumah	√					1.12
				√				1.04
					√			0.87
						√		1.12
							√	1.65
Average Error							1.16	

Sumber : Penulis, 2019

Percobaan ketiga dilakukan di luar rumah dengan medan landasan aspal. percobaan dengan memberikan *trigger* (maju,mundur,kanan,kiri,berhenti) dari *smartphone*, *delay* yang terjadi saat menekan tombol di aplikasi Android hingga *robot* merespond yaitu pada rata-rata 1.16 *second*. Percobaan ketiga ini lebih baik dibandingkan dengan percobaan kedua dengan melihat pada kondisi sinyal dan koneksi Internet yang digunakan oleh robot yaitu sinyal dan koneksi lebih baik pada saat berada di luar ruangan sehingga menghasilkan *delay* yang lebih kecil.

Dari keseluruhan percobaan, dapat disimpulkan bahwa rata-rata delay yang paling kecil di dapat pada saat di halaman depan rumah, hal ini dapat terjadi karena melihat faktor sinyal, koneksi sinyal dan space ruang yang tidak ada halangan sehingga lebih cepat merespond perintah dari *smartphone* Android ke Hardware (*robot*).

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil pengujian terhadap sistem *robot* yang dibuat, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem dapat bekerja sesuai dengan yang direncanakan, yaitu:
 - a. Koordinat posisi *robot* dapat diketahui melalui Smartphone.
 - b. *Robot* dapat digerakkan melalui aplikasi di *Smartphone Android*.
2. Penyimpangan rata-rata pembacaan koordinat posisi adalah 0.13%
3. Penyimpangan rata-rata pembacaan jarak terhadap halangan adalah 6%
4. Besarnya delay rata-rata antara pelaksanaan oleh troli terhadap perintah Smartphone adalah 1.16 detik.

5.2 Saran

Prototype Robot Pembawa Barang ini dapat dikembangkan dengan melakukan antara lain hal-hal berikut ini:

1. Dikarena menggunakan *server free*, terkadang dalam pengetesan *server* terjadi *down* bahkan di *block* oleh MQTT Hive MQ, oleh karena itu alangkah baiknya menggunakan *server* yang berbayar untuk penerapannya.
2. Untuk pengembangan yang lebih lanjut, sebaiknya *prototype* diberikan tambahan kamera agar dapat melihat area sekitar yang nantinya akan ditampilkan di aplikasi *Android*.

DAFTAR PUSTAKA

- Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.
- Aryza, S. (2018). Arduino Uno based water turbidity meter using LDR. *International Journal of Engineering & Technology*, 2114-2117.
- Bishop, O. (2014). Dasar - dasar Elektronika. Jakarta: PT. Gelora Aksara Pratama.
- Conferences on Internet of Things, and Cyber, Physical and Social Computing*, 620-3.
Depok: Universitas Indonesia.
- Ecadio. (2017, Januari 23). *Berkenalan dengan Arduino Pro Mini*. Retrieved from <http://ecadio.com/belajar-dan-mengajar-arduino-pro-mini/>
- F Abedan Kondori., F. Y. (2015). Smart Baggage in Aviation. *IEEE International*
- Hariyanto, E., & Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR)*, 5(10), 1363-1365.
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
- Hivemq. (2016, Januari 25). *HiveMQ*. Retrieved from <http://www.hivemq.com>
- Indah, N. (2017, Januari 31). *curhat porter bandara cengkareng yang kini gratis semangatnya beda*. Retrieved from <http://news.detik.com>
- Jatmiko, W. (2015). Robot Lego Mindstorm. In *Teori dan Praktek* (pp. 1-134).
- K Kovalchuck, O. S. (2016). IoT Device for Object's Power Remote Control. *IEEE*, 3-112.

- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), 1-6.
- Kurnia, D. (2017). Analisis QoS Pada Pembagian Bandwidth Dengan Metode Layer 7 Protocol, PCQ, HTB Dan Hotspot Di SMK Swasta Al-Washliyah Pasar Senen. *CESS (Journal of Computer Engineering, System and Science)*, 2(2), 102-111.
- Kurnia, D., Dafitri, H., & Siahaan, A. P. U. (2017). RSA 32-bit Implementation Technique. *Int. J. Recent Trends Eng. Res*, 3(7), 279-284.
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.
- Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. *Int. J. Recent Trends Eng. Res*, 3(8), 210-217.
- NMEA. (2018, Februari 1). *GPS Information - NMEA*. Retrieved from <http://www.gpsinformation.org/dale/nmea.htm>
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, 3(8), 196-201.
- Rahim, R., Aryza, S., Wibowo, P., Harahap, A. K. Z., Suleman, A. R., Sihombing, E. E., ... & Agustina, I. (2018). Prototype file transfer protocol application for LAN and Wi-Fi communication. *Int. J. Eng. Technol.*, 7(2.13), 345-347.
- Ramadhani, S., Suherman, S., Melvasari, M., & Herdianto, H. (2018). Perancangan Teks Berjalan Online Sebagai Media Informasi Nelayan. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
- Ruwaida, D., & Kurnia, D. (2018). Rancang Bangun File Transfer Protocol (FTP) dengan Pengamanan Open SSL pada Jaringan VPN Mikrotik di SMK Dwiwarna. *CESS (Journal of Computer Engineering, System and Science)*, 3(1), 45-49.
- Singh Ashwini, S. M. (2016). Baggage Tracking and handling System using RFID and IoT for Airport. 0-466.
- W Shehieb, S. H. (2016). A smart System to minimize Mishandled Luggage at Airport. *IEEE*, 154-8.

