

BAB 3

PERANCANGAN SISTEM

3.1 ALAT DAN BAHAN

Dalam pembuatan alat ini dibutuhkan beberapa komponen yang dibutuhkan dalam perancangan sistem dengan kecepatan ping *internet* yang digunakan sebesar 30ms-70ms. Alat dan bahan yang digunakan dapat dilihat pada tabel 3.1 dibawah ini :

Tabel 3.1 Daftar Alat dan Bahan

No	Alat dan Bahan	Jumlah
1	Laptop	1
2	<i>Smartphone</i>	1
3	NodeMCU ESP8266	1
4	Sensor <i>Rain-FC37</i>	2
5	Sensor Ultrasonik-HC SR04	3
6	Motor Servo	1
6	LED	2
7	<i>Buzzer</i>	2
8	<i>Power Supply</i>	1
9	<i>Software Arduino</i>	1
10	<i>Google Firebase</i>	1
11	<i>Software MIT App Inventor</i>	1
12	<i>Software Wireshark</i>	1

3.1.1 Laptop

Pada perancangan tugas akhir ini laptop yang digunakan sebagai alat dalam mengolah seluruh bahan data yang ada, selain itu laptop digunakan untuk memberikan pengkodean pada sistem serta sebagai media pengambilan hasil data. Spesifikasi laptop yang digunakan pada

tugas akhir ini yaitu prosessor Intel(R) Core(TM) i3-6006U, kecepatan *clock* sebesar 2.00 GHz, dan RAM *memory* sebesar 4GB.

3.1.2 Smartphone

Pada perancangan alat tugas akhir ini *smartphone* yang digunakan sebagai alat yang mempunyai aplikasi android dalam monitoring sistem dari perancangan alat. Spesifikasi pada *smartphone* yang digunakan yaitu prosessor *Quad-core*, *speed core* sebesar 1,8 GHz, penggunaan OS android v5.0 lollipop, RAM sebesar 2GB.

3.1.3 NodeMCU ESP8266

Pada perancangan alat tugas akhir ini NodeMCU ESP8266 sebagai mikropengendali, NodeMCU ini digunakan sebagai pengendali utama pada sistem yang akan dibuat. Pada perangkat selain dalam pengendali utama juga digunakan sebagai media dalam pengiriman hasil data ke *google firebase* yang telah dibuat. Pengiriman data tersebut menggunakan media modul *wifi* ESP8266 yang telah terpasang pada NodeMCU dengan koneksi *internet*. NodeMCU yang akan digunakan yaitu NodeMCU yang dilengkapi GPIO, PWM, IIC, *1-wire*, dan ADC.

3.1.4 Sensor Rain FC-37

Pada perancangan alat tugas akhir ini sensor hujan atau sensor *rain-fc37* digunakan dalam sistem untuk membedakan jenis sampah yang masuk berdasarkan nilai ADC sampah tersebut. Pada saat sampah masuk dan ditampung sementara pada tempat pemilah sampah, sensor hujan akan mendeteksi nilai ADC pada sampah tersebut yang nantinya akan menggerakkan motor servo ke arah tempat sampah basah atau kering sesuai dengan nilai ADC yang dikeluarkan.

3.1.5 Sensor Ultrasonik HC SR04

Pada perancangan alat tugas akhir ini sensor ultrasonik digunakan pada sistem sebagai pendeteksi ketinggian sampah pada setiap tempat sampah dan mendeteksi adanya sampah yang masuk. Ketinggian sampah tersebut akan dimonitor menggunakan aplikasi

android sesuai dengan jenis tempat sampah yang digunakan baik tempat sampah basah ataupun tempat sampah kering.

3.1.6 Motor Servo

Pada perancangan alat tugas akhir ini motor servo digunakan dalam sistem untuk menggerakkan tempat pemilah jenis sampah secara otomatis dengan melihat nilai ADC yang dihasilkan oleh sensor hujan.

3.1.7 LED (*Light Emitting Diode*)

Pada perancangan alat tugas akhir ini komponen LED digunakan sebagai indikator pada setiap tempat sampah apabila tempat sampah penuh maka LED akan menyala.

3.1.8 Buzzer

Pada perancangan alat tugas akhir ini *buzzer* digunakan sebagai alarm pada setiap tempat sampah apabila tempat sampah penuh bersamaan dengan menyalanya indikator lampu LED.

3.1.9 Software Arduino IDE

Pada perancangan alat tugas akhir ini *software* arduino IDE digunakan untuk memprogram sistem kepada masing-masing perangkat yang digunakan. Pada *software* arduino IDE bahasa yang digunakan yaitu java, bahasa C dan bahasa C++.

3.1.10 Google Firebase

Pada perancangan alat tugas akhir ini *google firebase* digunakan sebagai *database* untuk menyimpan hasil data dari perancangan secara sistematis, yang nantinya data akan diteruskan ke aplikasi android.

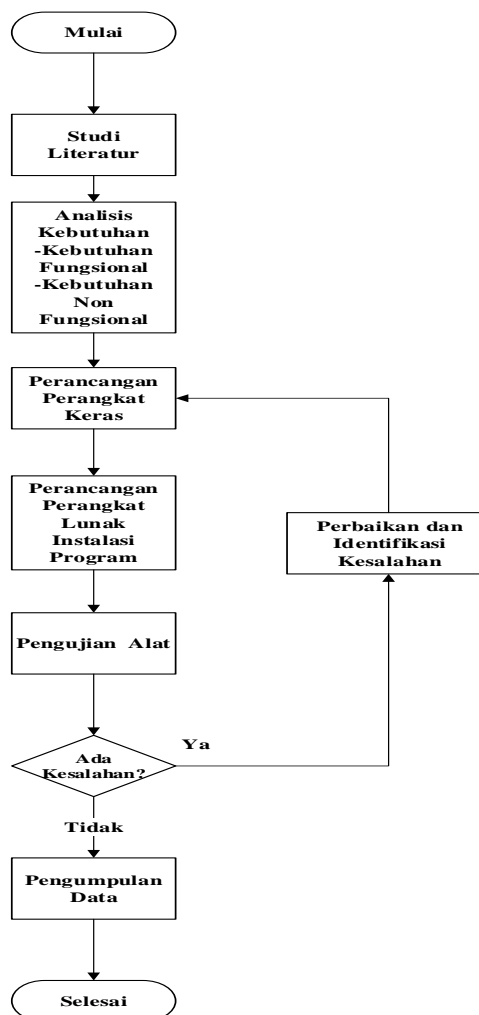
3.1.11 Software MIT App Inventor

Pada perancangan alat tugas akhir ini *software* MIT App *Inventor* digunakan untuk membuat sistem monitoring ketinggian tempat sampah basah dan tempat sampah kering, dimana data monitoring tersebut diambil dari *database* pada *software thingspeak*. *Design view* yang tersedia terdiri dari lima komponen dasar yaitu *palette*, *viewer*, *component*, *media and properties*. *Code block* pada app inventor digunakan untuk mengatur dan melakukan jalannya program yang dibuat.

3.1.12 Software Wireshark

Pada perancangan alat tugas akhir ini *software wireshark* yang digunakan adalah versi 2.6.6. *Software* ini digunakan untuk mendapatkan hasil dari kualitas QoS saat menjalankan pengiriman data menggunakan *internet* kepada sebuah *website* ataupun aplikasi di android. Pada saat melakukan penginstalan juga dapat menginstal aplikasi WinCap yang merupakan *driver-driver* khusus yang dipakai oleh *software wireshark*.

3.2 ALUR PENELITIAN

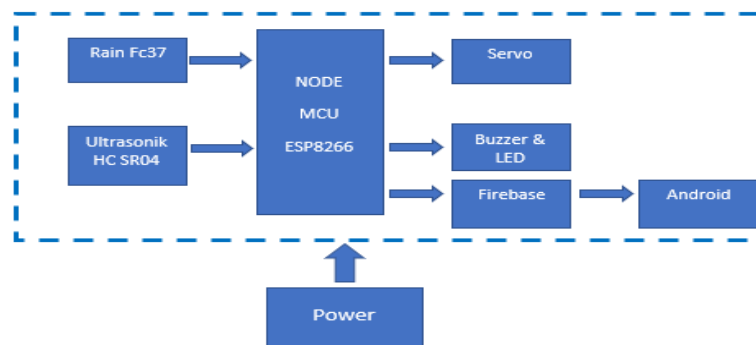


Gambar 3.1 Flowchart Alur Penelitian

Alur penelitian yang dilakukan dalam tugas akhir ini yaitu menggunakan studi literatur, dimana pada alur untuk melakukan pengumpulan data refrensi dari buku artikel, jurnal, dan situs yang berkaitan

dengan materi tugas akhir. Analisis kebutuhan yang digunakan yaitu kebutuhan fungsional dan non fungsional. Maksud dari kebutuhan fungsional dan non fungsional adalah kebutuhan yang diperlukan pada proses perancangan alat. Perancangan perangkat keras ini dilakukan agar mempermudah proses yang nantinya akan dilakukan secara bertahap. Perancangan perangkat lunak dan instalasi program, dimana pada alur ini dilakukan dengan cara membuat *flowchart*. Dalam *flowchart* terdapat *step by step* proses jalannya perancangan alat. Dalam perancangan perangkat lunak ini digunakan bahasa pemrograman arduino. Kemudian pada pengujian alat, dimana proses ini bertujuan menguji perangkat atau alat yang telah dibuat sudah sesuai dengan rencana dan dapat berfungsi dengan baik. Pengujian yang akan dilakukan adalah menguji setiap bagian blok sistem, jika ada kesalahan akan dilanjutkan dengan perbaikan dan identifikasi kesalahan pada sistem. Terakhir Pengumpulan data, setelah alat diuji dan berhasil maka dilanjutkan dengan mengambil dan mengumpulkan data yang diperlukan dari hasil pengujian pada perancangan alat yang dibuat.

3.2.1 Blok Diagram Perancangan Sistem

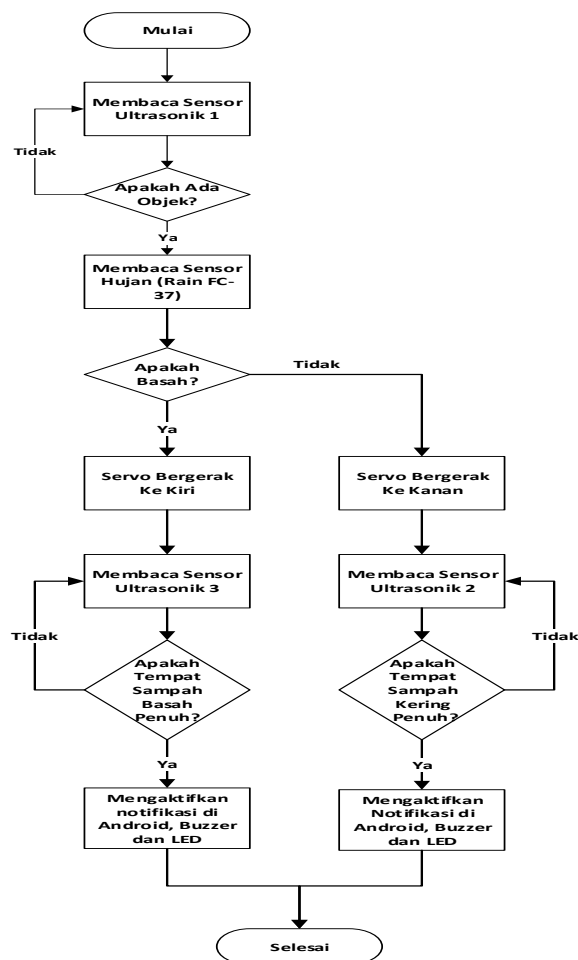


Gambar 3.2 Blok Diagram Perancangan Sistem

Pada alur tahapan ini akan dijelaskan perancangan sistem yang akan dibuat. Pada gambar 3.2 merupakan blok diagram perancangan sistem yang dibuat. NodeMCU esp8266 berfungsi sebagai otak sistem atau mikrokontroler untuk menggerakkan sistem yang menerima daya dari *power supply* kemudian, sensor *rain fc 37* dan sensor ultrasonik berfungsi sebagai *input* yang mengirim data ke NodeMCU. Setelah menerima data, data tersebut akan di proses dan diteruskan menuju

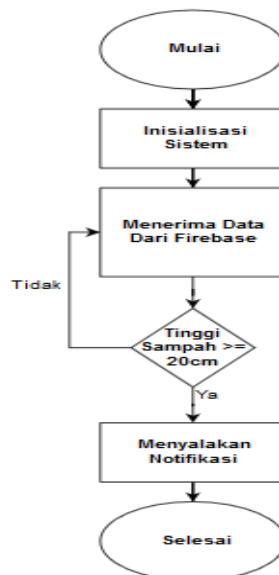
output. Output dari sistem yaitu motor servo, buzzer, LED, firebase, dan Aplikasi pada android. Untuk motor servo akan bergerak sesuai nilai ADC dari sensor *rain fc-37*, sedangkan sensor ultrasonik berfungsi untuk memantau ketinggian tempat sampah serta mendeteksi objek sampah yang masuk. Kemudian ketinggian sampah akan dimonitoring lewat aplikasi pada android dan jika tempat sampah penuh maka buzzer dan LED menyala. NodeMCU akan mengirim data ketinggian tempat sampah ke *google firebase*, dimana pada *web* ini data akan disimpan secara *real time*. Kemudian data yang tersimpan secara *real time* di *firebase* dikirim ke aplikasi android yang dibuat menggunakan MIT APP inventor. Aplikasi pada android berfungsi untuk memonitor ketinggian sampah dan memberikan notifikasi apabila tempat sampah sudah penuh.

3.2.2 Flowchart Alur Sistem



Gambar 3.3 Flowchart Alur Mikrokontroler

Diagram alur pada gambar 3.3 menunjukkan alur kerja mikrokontroler sesuai dengan perangkat lunak yang akan dirancang. Mikrokontroler akan melakukan tindakan sesuai program yang dimasukkan, dimana semua perangkat yang terhubung akan berjalan dan dikendalikan oleh mikrokontroler. Untuk proses pertama yaitu membaca sensor ultrasonik 1, jika ada objek yang mendekat maka diteruskan dengan membaca sensor *rain fc-37* atau sensor hujan. Jika objek tersebut terdeteksi basah maka akan menggerakkan motor servo kekiri dan jika tidak terdeteksi maka motor servo akan bergerak kekanan. Selanjutnya pembacaan sensor ultrasonik 3 untuk tempat sampah basah, dan sensor ultrasonik 2 untuk tempat sampah kering. Untuk pembacaan sensor ultrasonik 3, jika tempat sampah basah terdeteksi penuh maka akan mengaktifkan notifikasi di android dan menyalakan *buzzer* serta LED pada tempat sampah basah. Sedangkan untuk pembacaan sensor ultrasonik 2 apabila tempat sampah kering penuh akan mengaktifkan notifikasi di android dan menyalakan *buzzer* serta LED pada tempat sampah kering.



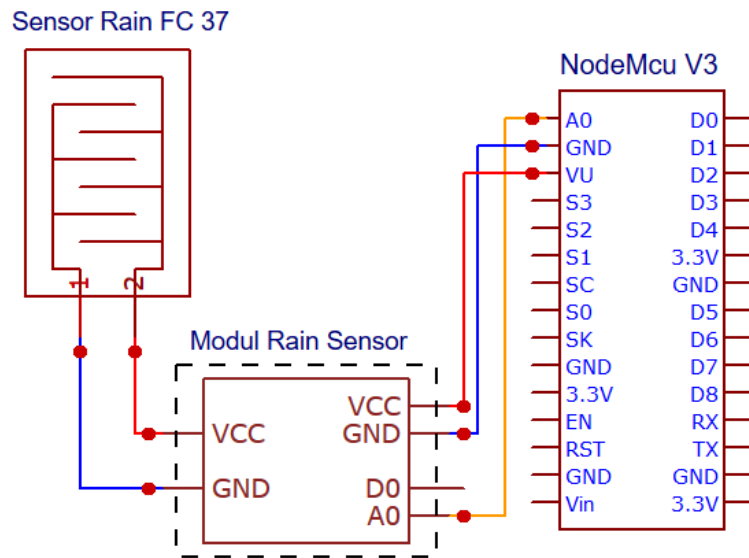
Gambar 3.4 Flowchart Alur Aplikasi Android

Pada gambar 3.4 proses pertama yang dilakukan adalah inisialisasi sistem untuk pemberian nilai awal yang dilakukan saat

deklarasi variabel. Kemudian proses selanjutnya yaitu menerima data dari *firebase* yang dibuat, dimana data dari *firebase* berasal dari data yang diperoleh pada mikrokontroler yang kemudian akan diteruskan ke aplikasi android. Setelah data diperoleh dari *firebase* Jika tinggi sampah pada tempat sampah ≥ 20 cm maka akan menyalakan notifikasi pada aplikasi android.

3.2.3 Perancangan Perangkat Keras

a. Antarmuka NodeMCU dengan Sensor *Rain FC-37*



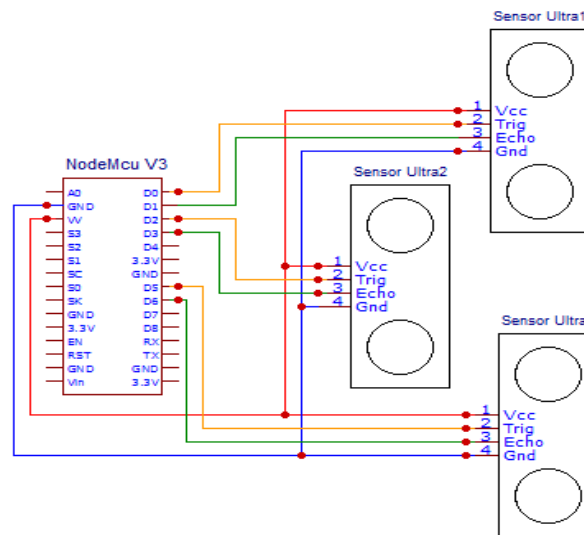
Gambar 3.5 Koneksi Pin NodeMCU dengan Sensor *Rain FC-37*

Sensor *rain fc-37* adalah komponen yang dapat berfungsi sebagai pengukur atau mendeteksi kering basah sampah yang masuk. Prinsip kerja pada sensor ini, yaitu ketika air menyentuh kedua elektroda (tembaga) maka tegangan 5V akan terhubung dengan *output* dan sebagian tegangan akan berkurang karena air berfungsi sebagai penghantar. Pada gambar 3.5 menunjukkan pembagian pin yang digunakan sensor hujan untuk mendeteksi jenis sampah, apakah sampah basah atau kering. Koneksi pin antara NodeMCU dengan sensor *rain fc-37* dapat dilihat penjelasannya pada tabel 3.2 berikut :

Tabel 3.2 Koneksi antara Node MCU dengan Sensor *Rain FC-37*

No.	PIN Sensor <i>Rain FC-37</i>	Fungsi
1	A0	Pembacaan Sensor di <i>Port</i> A0 NodeMCU
2	VCC	Catu Daya Sensor 5V dari NodeMCU
3	GND	Grounding

b. Antarmuka NodeMCU dengan Sensor Ultrasonik



Gambar 3.6 Koneksi Pin NodeMCU dengan Sensor Ultrasonik

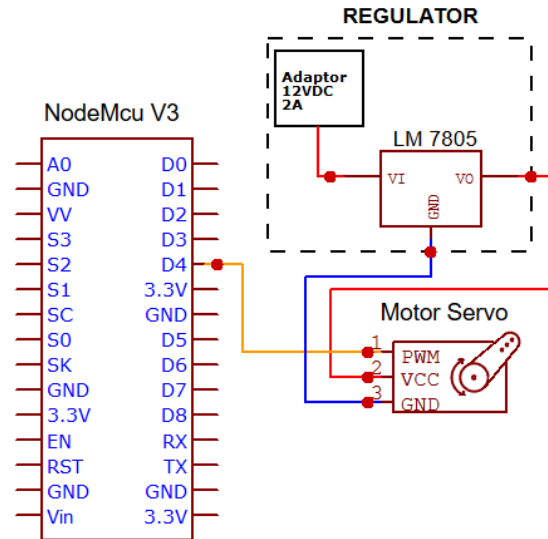
Sensor ultrasonik HC-SR04 adalah komponen yang dapat berfungsi untuk pengukur atau mendeteksi jarak dari suatu objek, sensor ini memiliki dua elemen yaitu elemen pendeteksi gelombang ultrasonik dan juga sekaligus elemen pembangkit gelombang ultrasonik. Sensor ultrasonik pada sistem ini digunakan untuk mendeteksi adanya sampah yang masuk dan mengukur ketinggian sampah pada tempat sampah. Metode pengukuran yang dilakukan untuk menguji sensor ultrasonik menggunakan penggaris sebagai perbandingan dalam menentukan baik atau tidaknya hasil dari nilai pengukuran pada sensor ultrasonik. Pada gambar 3.6 menunjukkan

pembagian pin yang digunakan sensor ultrasonik untuk mendeteksi objek dan mengukur ketinggian sampah. Koneksi pin antara NodeMCU dengan sensor ultrasonik dapat dilihat penjelasannya pada tabel 3.3 berikut :

Tabel 3.3 Koneksi antara NodeMCU dengan Sensor Ultrasonik

No.	PIN Sensor Ultrasonik	Fungsi
1	D0	Pembacaan Trigger Sensor Ultrasonik 1 di <i>Port</i> D0 NodeMCU untuk membangkitkan sinyal
2	D1	Pembacaan Echo Sensor Ultrasonik di <i>Port</i> D1 NodeMCU untuk menerima sinyal
3	D2	Pembacaan Trigger Sensor Ultrasonik 2 di <i>Port</i> D2 NodeMCU untuk membangkitkan sinyal
4	D3	Pembacaan Echo Sensor Ultrasonik 2 di <i>Port</i> D3 NodeMCU untuk menerima sinyal
5	D5	Pembacaan Trigger Sensor Ultrasonik 3 di <i>Port</i> D5 NodeMCU NodeMCU untuk membangkitkan sinyal
6	D6	Pembacaan Echo Sensor Ultrasonik 3 di <i>Port</i> D6 NodeMCU untuk menerima sinyal
7	VCC	Catu Daya 5V dari NodeMCU
8	GND	Grounding

c. Antarmuka NodeMCU dengan Servo



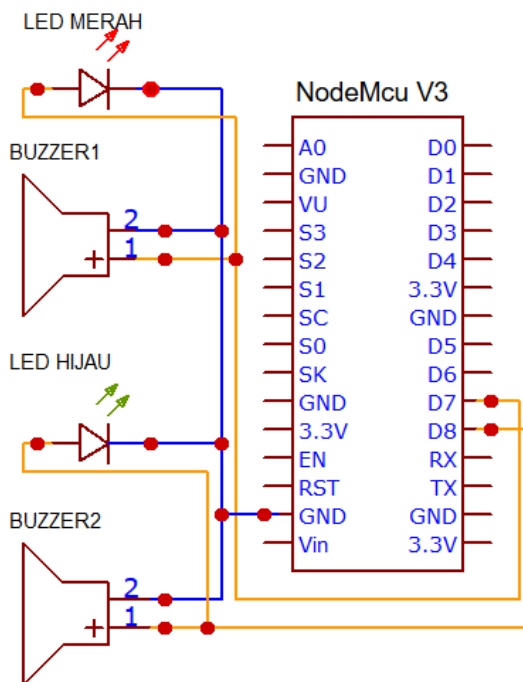
Gambar 3.7 Koneksi PIN NodeMCU dengan Motor Servo

Pada sistem motor servo digunakan sebagai penggerak untuk menggerakkan pemilah sampah antara sampah basah dan sampah kering. Pada gambar 3.7 menunjukkan pembagian pin yang digunakan motor servo untuk menggerakkan pemilah sampah. Motor servo yang digunakan menggunakan catu daya 5V. Koneksi pin antara NodeMCU dengan motor servo mg995 dapat dilihat penjelasannya pada tabel 3.4 berikut :

Tabel 3.4 Koneksi antara Node MCU dengan Motor Servo MG995

No.	PIN Motor Servo	Fungsi
1	D4	Pin Data Servo di Port D4 NodeMCU
2	VCC	Catu Daya Sensor 5V dari Regulator
3	GND	Grounding dari Regulator

d. Antarmuka NodeMCU dengan *Buzzer* dan LED



Gambar 3.8 Koneksi PIN NodeMCU dengan *Buzzer* dan LED

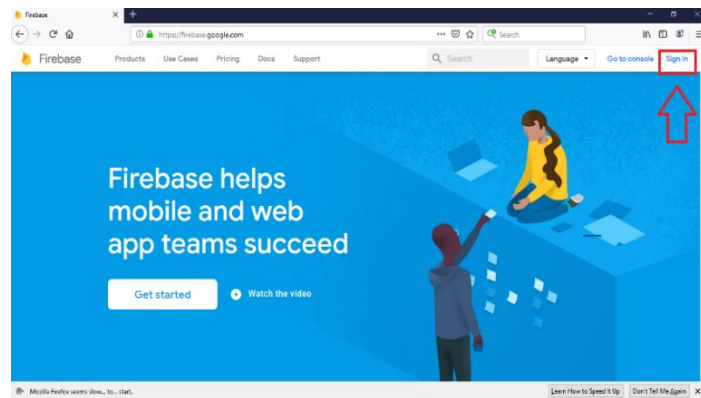
Pada sistem *buzzer* dan LED berfungsi sebagai alarm dan indikator saat ketinggian sampah pada tempat sampah sudah penuh. Pada gambar 3.8 menunjukkan pembagian pin yang digunakan *buzzer* dan LED untuk menyalakan alarm dan indikator lampu saat ketinggian sampah sudah penuh. Koneksi pin antara NodeMCU dengan *buzzer* dan LED dapat dilihat penjelasannya pada tabel 3.5 berikut :

Tabel 3.5 Koneksi antara Node MCU dengan *Buzzer* dan LED

No.	PIN LED dan Buzzer	Fungsi
1	D7	Kutub Positif LED merah dan <i>buzzer</i> 1 di Port D7 NodeMcu
2	D8	Kutub Positif LED hijau dan <i>buzzer</i> 2 di Port D8 NodeMcu
3	GND	Grounding

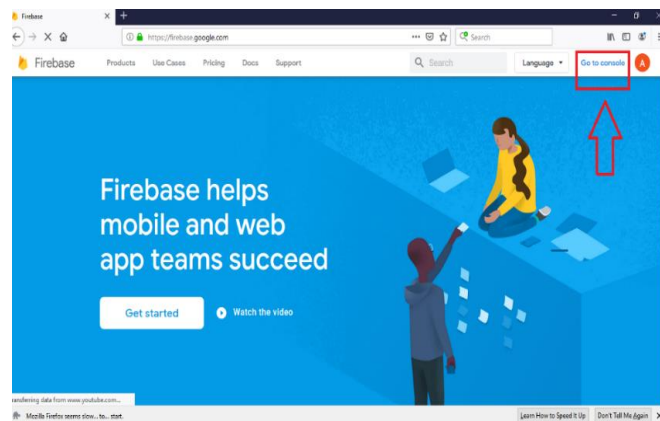
3.2.4 Perancangan Perangkat Lunak

a. *Google Firebase*



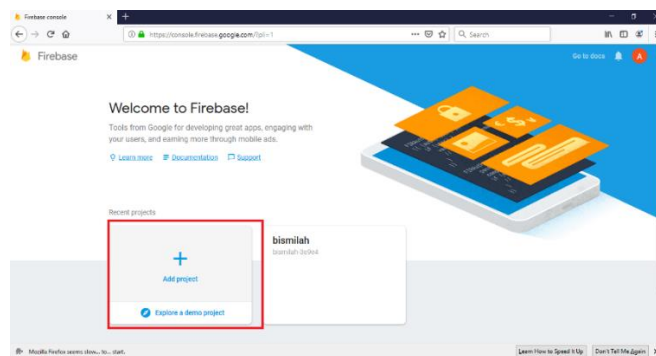
Gambar 3.9 Tampilan Awal *Firebase*

Pada tugas akhir ini *google firebase* digunakan sebagai *database*. Pada gambar 3.9 sebelum membuat *database* pada *google firebase*, terlebih dahulu melakukan *sign in* menggunakan akun *google*.



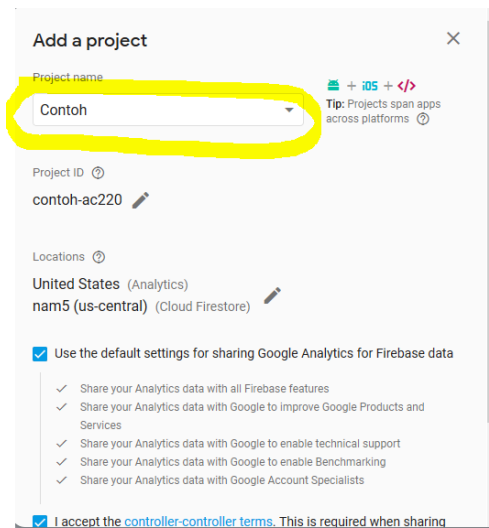
Gambar 3.10 Menuju *Console*

Pada gambar 3.10 setelah melakukan *sign in* lalu klik *go to console* untuk menuju pembuatan proyek.



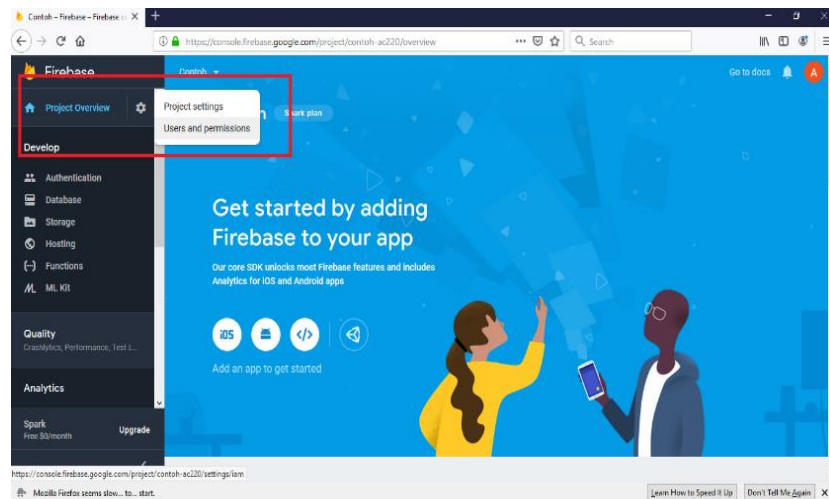
Gambar 3.11 Membuat *Proyek*

Kemudian setelah menuju *console* pilih *add project* untuk membuat proyek pada *firebase*.



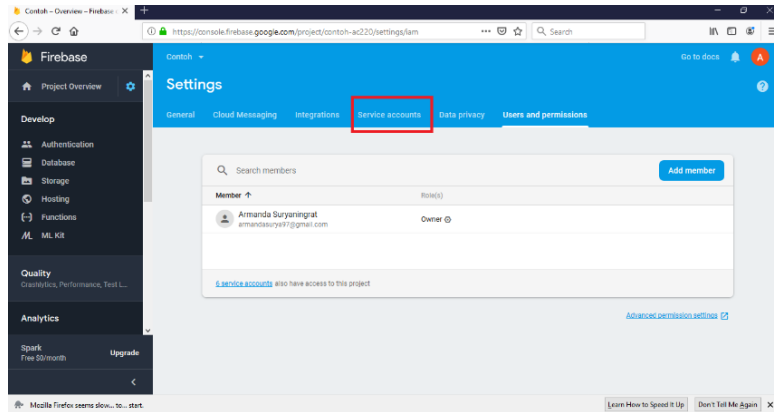
Gambar 3.12 Mengisi Nama *Project*

Ketikan nama proyek yang akan dibuat lalu isi semua persetujuan yang ada kemudian klik *create project*.



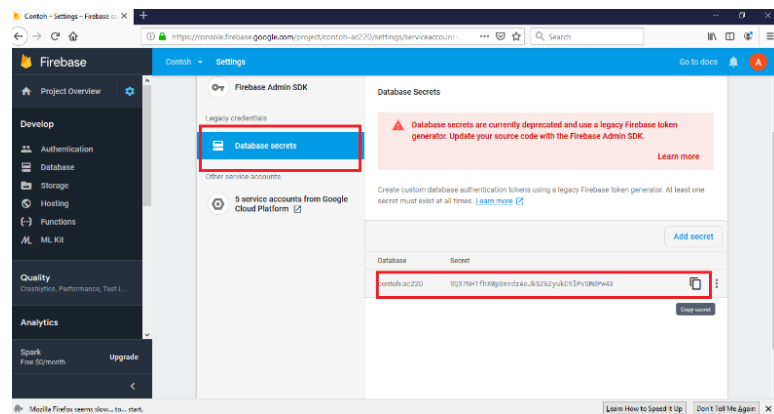
Gambar 3.13 Tampilan Awal *Project*

Setelah masuk pada *project firebase* pilih menu *project overview*, pilih *user and permissions* maka akan muncul tampilan seperti gambar 3.14



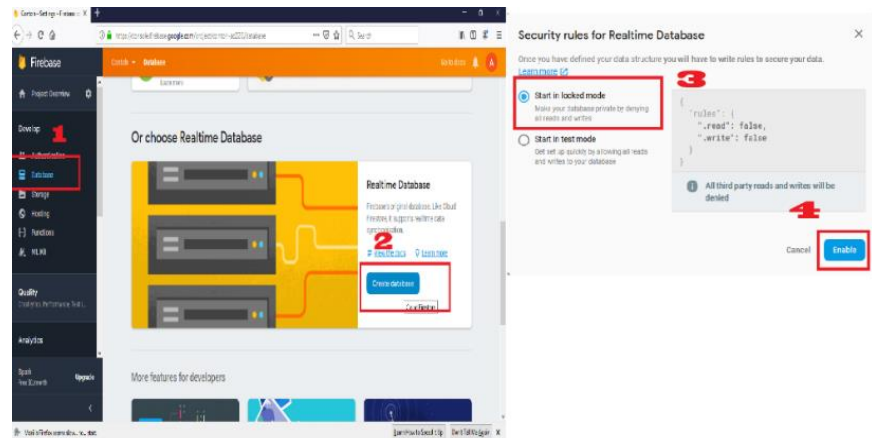
Gambar 3.14 Tampilan Settings

Pada gambar 3.14 merupakan tampilan pengaturan, untuk melihat kode atau token yang dimasukan pada *coding* arduino IDE pilih *service accounts* maka akan muncul tampilan seperti gambar 3.15.



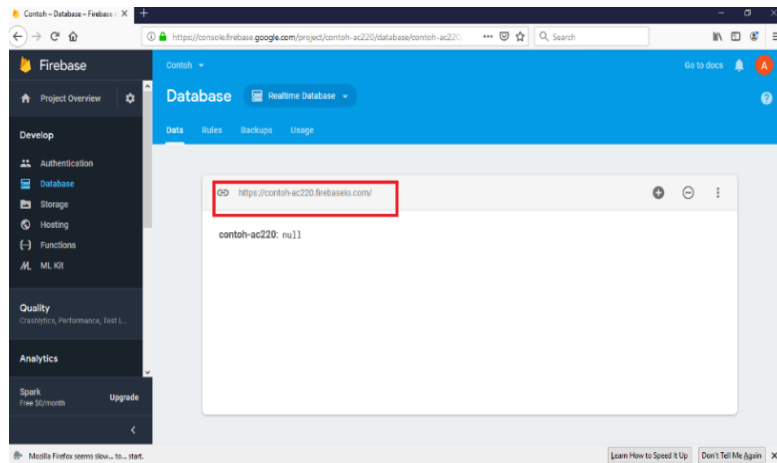
Gambar 3.15 Tampilan Service Account

Setelah menuju *service account*, pilih *database secret* lalu klik *show* pada bagian bawah maka akan muncul kode yang digunakan untuk mengisi *firebase authentication* pada arduino IDE.



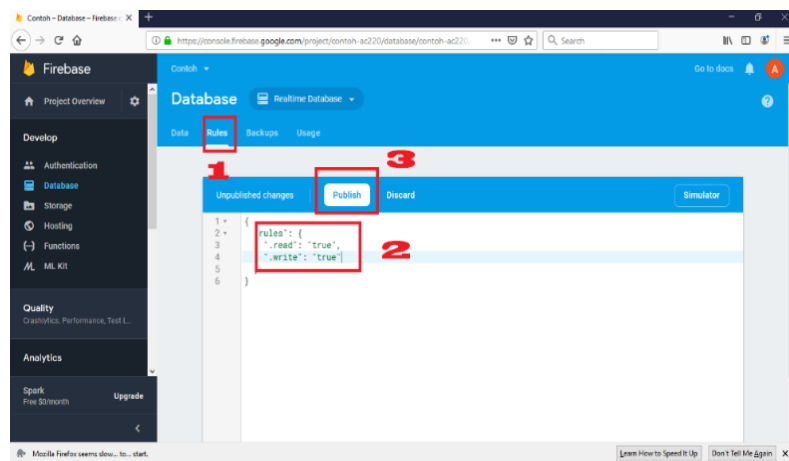
Gambar 3.16 Membuat Realtime Database

Selanjutnya klik *database* pada menu pilih *real time database*, setelah terbuka jendela baru pilih *start in locked mode* lalu klik *enable*.



Gambar 3.17 Tampilan Realtime Database

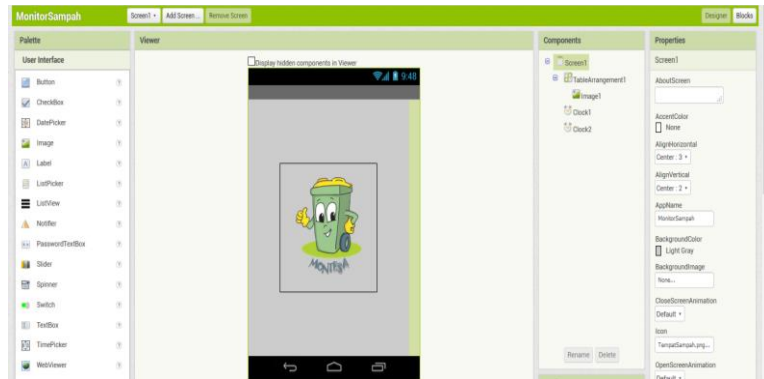
Pada gambar 3.17 merupakan tampilan awal *real time database*, salin link yang ada pada gambar untuk mengisi *firebase host* pada arduino IDE dan *firebase token* pada MIT App Inventor.



Gambar 3.18 Tampilan Rules Pada Database

Pada gambar 3.18 merupakan aturan pembacaan nilai yang di kirim ke *database*. Pada MIT App Inventor untuk membaca nilai yang ada pada *firebase* maka pilih *rules*, lalu ubah perintah yang ada dari "*false*" menjadi "*true*" seperti gambar 3.18 lalu klik *publish*.

b. Desain Aplikasi MIT App Inventor



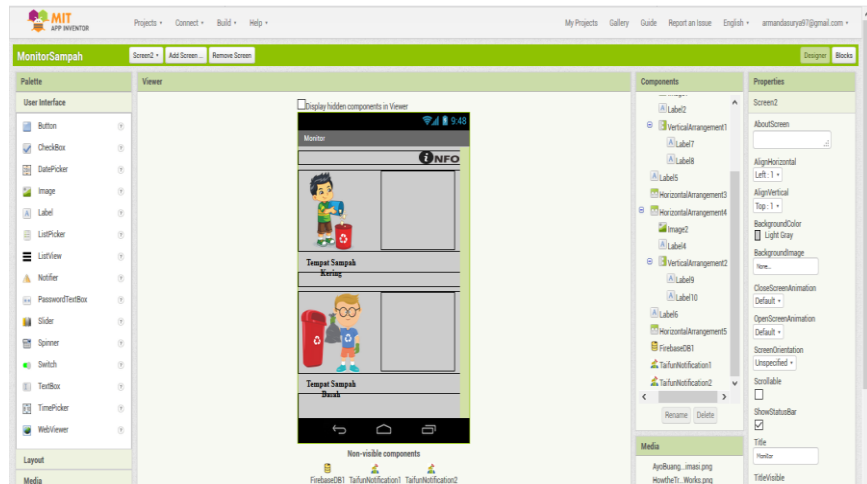
Gambar 3.19 Tampilan *Flash Screen* Aplikasi

Desain aplikasi yang digunakan pada tugas akhir ini menggunakan MIT app inventor. App inventor digunakan pengguna untuk memprogram komputer dalam menciptakan aplikasi tanpa melakukan koding, hanya cukup dengan *drag and drop* objek. Gambar 3.19 diatas merupakan tampilan awal yang digunakan sebagai *flash screen* yang berdurasi sekitar 3detik sebelum menuju *screen* berikutnya. Pada gambar 3.20 akan menampilkan *block* untuk membuat *flash screen*.



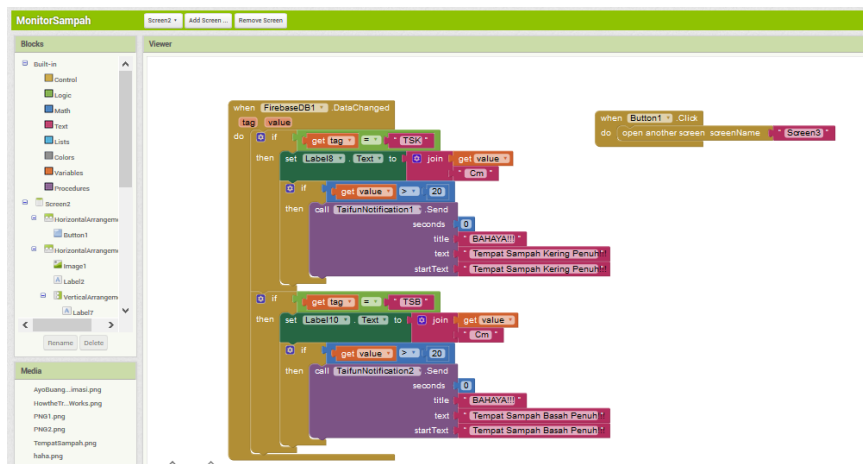
Gambar 3.20 *Block Screen 1* Aplikasi

Pada gambar 3.20 *block* yang digunakan berfungsi untuk menampilkan *flash screen* dengan durasi 3detik dan kemudian akan menuju ke *screen 2*.



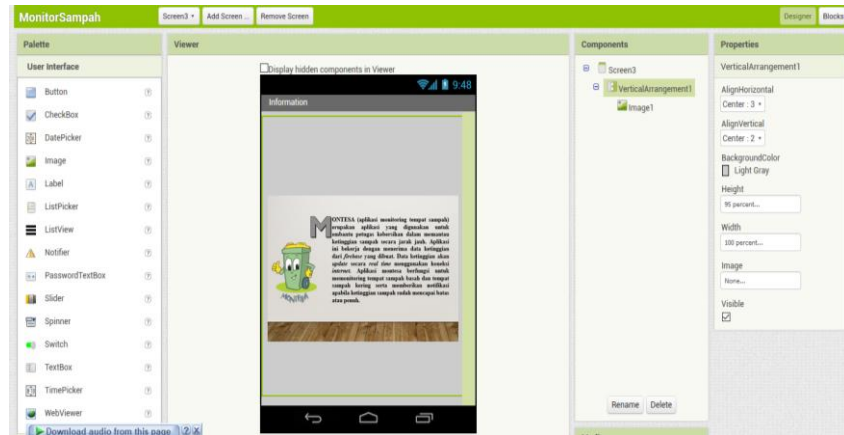
Gambar 3.21 Tampilan Screen 2 Monitoring

Pada gambar 3.21 diatas merupakan tampilan untuk monitoring tempat sampah pada alat yang akan dibuat. Dalam *screen* ini terdiri dari monitoring tempat sampah basah dan tempat sampah kering, yang nantinya nilai ketinggian di tempat sampah akan muncul dan kemudian akan memunculkan notifikasi apabila tinggi pada tempat sampah sudah penuh.



Gambar 3.22 Block Screen 2 Aplikasi

Pada gambar 3.22 digunakan untuk membuat sistem pada *screen 2* untuk monitoring, dimana nilai yang di dapat di ambil dari *database* melalui *firebase*. Kemudian notifikasi yang digunakan akan muncul ketika nilai yang tertampil pada aplikasi lebih dari 20 Cm. Tombol informasi digunakan untuk memunculkan cara kerja sistem pada *screen 3* seperti pada gambar 3.23 di bawah ini.

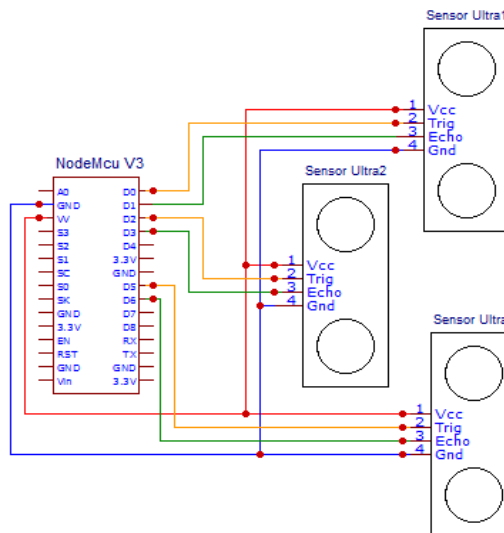


Gambar 3.23 Tampilan Informasi pada Screen 3

3.2.5 Pengujian *Hardware*

Pada tahap ini adalah proses pengambilan hasil data bagian *hardware* dilakukan, untuk mendapatkan hasil data pada tugas akhir menggunakan beberapa alat dan bahan seperti penggaris, meteran dan osiloskop.

a. Pengujian Sensor Ultrasonik HC SR04 Untuk Mengukur Ketinggian Sampah



Gambar 3.24 Koneksi Pin NodeMCU dengan Sensor Ultrasonik

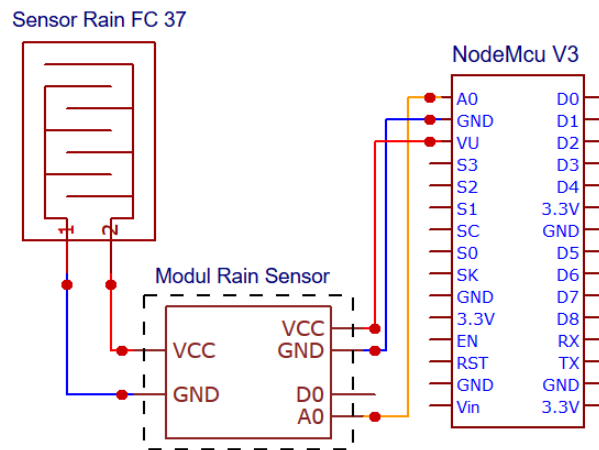
Pada pengujian ini bertujuan untuk mengetahui ketinggian sampah menggunakan sensor ultrasonik. Untuk kalibrasi pada

sensor ultrasonik, ini dapat digunakan metode persamaan untuk mencari tingkat akurasi dengan rumus

$$Error (\%) = \frac{a-b}{a} \times 100\% \quad (3.1)$$

Dimana b adalah Tinggi pembacaan sensor, dan a adalah tinggi sampah sebenarnya. Untuk tinggi sebenarnya dilakukan melalui pengukuran secara manual menggunakan penggaris atau meteran. Pengambilan data pengukuran dilakukan menggunakan 3 kali percobaan dengan rentang jarak 10 cm, 15 cm, dan 20 cm. Untuk perbandingan nilai pengukuran maka digunakan meteran atau penggaris sebagai perbandingan hasilnya.

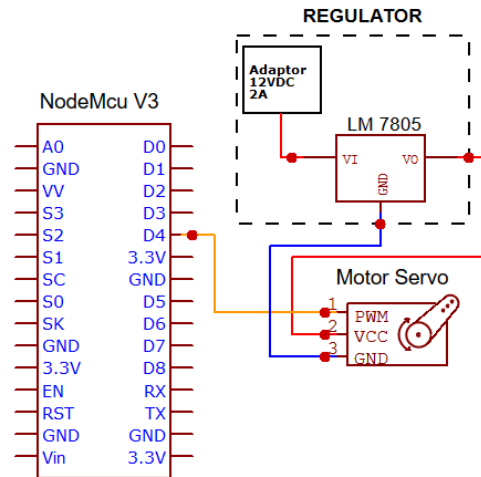
b. Pengujian Perangkat Sensor *Rain FC 37* Untuk Mengukur Nilai Sampah Basah dan Sampah Kering



Gambar 3.25 Pembagian PIN NodeMCU dengan Sensor *Rain FC 37*

Pada pengujian ini bertujuan untuk mengetahui nilai sampah basah dan sampah kering menggunakan sensor hujan. Dimana sampah basah adalah sampah yang mengandung air dan susah untuk terbakar, sedangkan sampah kering tidak mengandung air dan mudah terbakar. Pada pengujian sensor ini dengan cara mengambil nilai dari beberapa contoh sampah seperti kayu, kertas, sisa buah, tisu, botol plastik, botol kaleng, dan plastik pembungkus makanan.

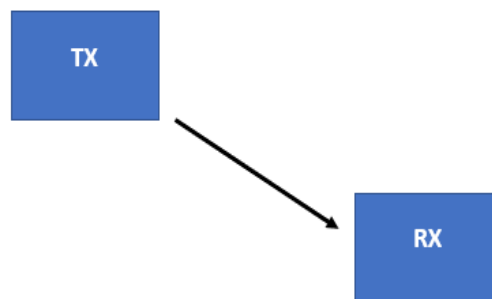
c. Pengujian Perangkat Motor Servo



Gambar 3.26 Koneksi PIN NodeMCU dengan Motor Servo

Pada pengujian ini bertujuan untuk mengetahui kinerja dari motor servo apakah berfungsi dengan baik atau tidak. Pada proses pengambilan data menggunakan osiloskop . Terdapat dua pengukuran yaitu saat motor servo aktif dan saat motor servo mati. Pada proses pengukuran hasil data menggunakan osiloskop digunakan untuk menemukan *duty cycle* dan menemukan *interval pulse* dalam 1 gelombang.

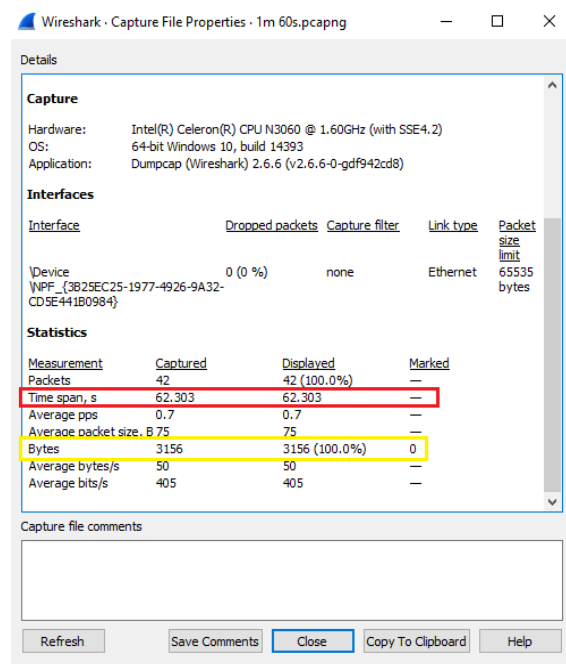
3.2.6 Pengujian *Quality Of Service* (QoS)



Gambar 3.27 Skema Pengujian *Quality Of Service*

Pada gambar 3.27 merupakan model pengukuran yang digunakan untuk mendapatkan parameter *quality of service* (QoS). Pada Tx terdapat NodeMCU yang digunakan sebagai sisi *transmitter* dan *access point*. Sedangkan pada Rx atau *receiver* terdapat laptop yang terinstall

aplikasi *wireshark*. Pengujian QoS yang akan dilakukan meliputi pengujian *delay*, *throughput*, dan *packet loss* dengan menggunakan *software wireshark* yang terinstall pada laptop.



Gambar 3.28 Hasil Pengujian Menggunakan Wireshark

Pada gambar 3.28 merupakan contoh hasil pengujian koneksi jaringan menggunakan aplikasi *wireshark*. Pada gambar 3.28 kotak berwarna merah adalah total *delay* yang dihasilkan dari seluruh paket, sedangkan kotak kuning adalah jumlah data yang dilakukan selama pengiriman paket.

a. Pengujian Delay

Pengujian *delay* dilakukan untuk mengetahui total waktu tunda suatu *paket*. Pada pengujian *delay* menguji total selisih waktu data yang diterima di sisi Rx ke data terima selanjutnya. Proses pengujian *delay* dilakukan dengan pengujian pengiriman data dari sisi Tx ke sisi Rx dengan jumlah data dan jarak yang berbeda. Nilai *delay* dapat dinyatakan persamaan berikut :

$$\text{Total delay} = \text{Jumlah delay seluruh paket} \quad (3.2)$$

b. Pengujian Packet Loss

Pengujian *packet loss* dilakukan untuk mengetahui jumlah total paket pengiriman data yang hilang dari sisi Tx ke sisi Rx.

Proses pengujian *packetloss* dilakukan dengan pengujian pengiriman data dari sisi Tx ke sisi Rx dengan jumlah data dan jarak yang berbeda. Dimana pada pengujian *packet loss* untuk melihat *loss* yang terjadi pada *wireshark*.

c. Pengujian *Throughput*

Pengujian *throughput* dilakukan untuk mengetahui laju data pada pengiriman data. Proses pengujian *throughput* pada penelitian ini dilakukan dengan pengujian pengiriman data dari Tx ke Rx dengan waktu dan jarak yang berbeda. Nilai *throughput* diperoleh menggunakan persamaan berikut :

$$Throughput \text{ (bit/s)} = \frac{Ukuran \ data}{total \ delay} \times 8 \quad (3.3)$$