



9th Applied Business and Engineering Conference

Pengisian Kwh Meter Prabayar Menggunakan Plotter 2D

Muhammmad Irham¹, Noptin Harpawi²

¹Teknik Elektronika Telekomunikasi, Politeknik Caltex Riau, Rumbai, Pekanbaru, 27265

²Teknik Elektronika Telekomunikasi, Politeknik Caltex Riau, Rumbai, Pekanbaru, 27265

E-mail: irham17tet@mahasiswa.pcr.ac.id

Abstract

The rapid development of technology in the IoT (Internet of Things) field gave rise to the idea to create a final project entitled Charging Prepaid KWH Meters Using a 2D Plotter. The problem to be solved in the final project is to overcome the problem in terms of filling the Prepaid KWH Meter which is still done manually by pressing the keypad directly on the Prepaid KWH Meter, this will be a problem if the position of the Prepaid KWH Meter is difficult to reach or out of reach of hands, the second is to avoid direct contact with the Prepaid KWH Meter keypad so as to prevent the spread of the Covid 19 virus that can lodge in inanimate objects such as the Prepaid KWH Meter keypad. This final project uses the ThingSpeak IoT platform as a place to store and retrieve data via an internet connection and use the Android application as a place to remotely and monitor the status of Prepaid KWH Meters. Based on the tests that have been carried out, the number of steps needed by the stepper motor in theory and practice at a distance of 14 mm on the X axis and 9 mm on the Y axis, the error percentage is 1.785%. When ThingSpeak sends token data to the NodeMCU ESP8266 which can be seen on the Arduino IDE serial monitor, the average delay is 23.19 seconds. And in application testing there are 3 status conditions, namely the first "WAITING" status occurs when the token data transmission process is in progress or the 2D Plotter is working, the second "ADA LISTRIK" status occurs when the sensor does not detect the buzzer sound on the Prepaid KWH Meter, which The third status "TOKEN MAU HABIS" occurs when the sensor detects a buzzer sound on the Prepaid KWH Meter.

Keywords : *IoT (Internet Of Things), Prepaid KWH Meter, 2D Plotter, Android Aplication, NodeMCU ESP8266, Stepper Motor, Arduino IDE, MIT App Inventor*

Abstrak

Perkembangan teknologi pada bidang IoT (Internet of Things) yang cukup pesat menimbulkan ide untuk membuat suatu Proyek Akhir yang berjudul Pengisian KWH Meter Prabayar Menggunakan *Plotter 2D*. Masalah yang hendak diselesaikan dalam Proyek Akhir adalah mengatasi masalah dalam hal pengisian KWH Meter Prabayar yang masih dilakukan secara manual yaitu dengan cara menekan langsung *keypad* yang ada pada KWH Meter Prabayar, hal ini akan menjadi masalah jika posisi dari KWH Meter Prabayar tersebut susah untuk dijangkau atau jauh dari jangkauan tangan, yang kedua adalah untuk menghindari bersentuhan langsung dengan *keypad* KWH Meter Prabayar sehingga dapat mencegah

1099

ISSN: 2339 – 2053

Pekanbaru, 25 Agustus 2021



9th Applied Business and Engineering Conference

penyebaran virus Covid 19 yang dapat bersarang di benda-benda mati seperti halnya *keypad* KWH Meter Prabayar. Proyek Akhir ini menggunakan platform IoT *ThingSpeak* sebagai tempat untuk menyimpan dan mengambil data melalui koneksi internet dan menggunakan aplikasi *Android* sebagai tempat untuk meremote dan memonitor status KWH Meter Prabayar. Berdasarkan pengujian yang telah dilakukan adalah jumlah *step* yang dibutuhkan motor stepper secara teori dan praktek pada jarak 14 mm pada sumbu X dan 9 mm pada sumbu Y didapatkan persentase error sebesar 1,785 %. Pada saat *ThingSpeak* mengirimkan data token menuju *NodeMCU ESP8266* yang dapat dilihat pada serial monitor *Arduino IDE* maka didapatkan rata – rata *delay* 23,19 detik. Dan pada pengujian aplikasi terdapat 3 kondisi status, yaitu yang pertama status “*WAITING*” terjadi ketika proses pengiriman data token sedang berlangsung atau *Plotter 2D* sedang bekerja, yang kedua status “*ADA LISTRIK*” terjadi ketika sensor tidak mendeteksi bunyi *buzzer* pada KWH Meter Prabayar, yang ketiga status “*TOKEN MAU HABIS*” terjadi ketika sensor mendeteksi bunyi *buzzer* pada KWH Meter Prabayar.

Kata Kunci : *IoT (Internet Of Things)*, KWH Meter Prabayar, *CNC 2D* atau *Plotter 2D*, Aplikasi *Android*, *NodeMCU ESP8266*, Motor Stepper, *Arduino IDE*, *MIT App Inventor*

PENDAHULUAN

Pengendalian listrik merupakan suatu kegiatan sistem PLN yang dari dulu hingga sekarang yang terus dikembangkan. Mulai dari KWH Analog, hingga saat ini yang paling akurat yaitu KWH Meter yang berbasis digital atau prabayar (Token). Salah satu permasalahan pada sistem KWH Meter yang terbaru ini adalah cara pengontrolan yang dilakukan masih secara manual yaitu dengan cara menekan langsung keypad yang ada pada KWH Meter Prabayar tersebut. Dimana cara ini akan sedikit berbahaya jika tempat untuk pengisian ulang token susah untuk dijangkau misalnya : harus menggunakan tangga, kursi dan sebagainya dapat dilakukan dengan mudah jika menggunakan sebuah alat *Plotter 2D* yang terpasang di KWH Meter Prabayar.

(tseindonesia.id 2019) Dalam penelitian ini difokuskan pada cara pengisian pulsa KWH Meter Prabayar menggunakan prinsip *Plotter 2D*. KWH Meter prabayar merupakan KWH Meter yang digunakan untuk mengatasi kelemahan dari KWH Meter Analog. Adapun kelebihan dari KWH Meter Digital antara lain sebagai berikut : Sistem pembayarannya dengan sistem prabayar, dengan sistem prabayar lebih efisien dibandingkan dengan metode pembayaran umumnya, Akurasi perhitungan KWH Meter, tidak adanya tunggakan pembayaran tagihan listrik, kemudahan memutus sambungan



9th Applied Business and Engineering Conference

listrik pelanggan yang melakukan tunggakan tagihan dengan menggunakan alat yang bisa di set up dari jarak maksimal 200 meter.

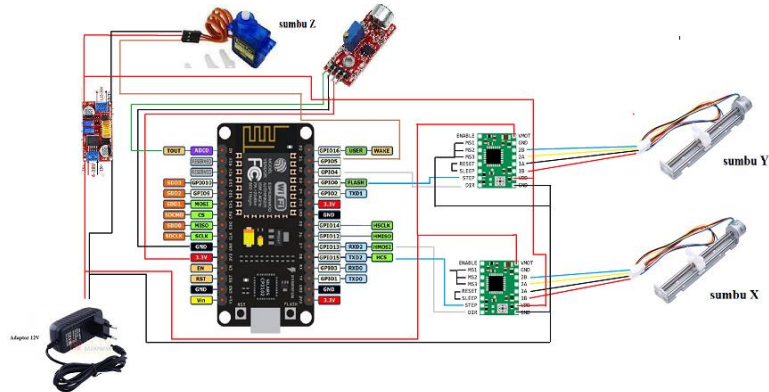
Oleh karena muncullah suatu gagasan untuk merancang sebuah alat pengisian pulsa pada KWH Meter Prabayar dengan menggunakan prinsip plotter 2D, dimana mikrokontroler *NodeMCU ESP8266* terhubung ke *smartphone* via *Wifi* menggunakan jaringan internet untuk mengendalikan *Plotter 2D*. Pengertian *Plotter 2D* adalah sebuah mesin yang secara otomatis akan menggambar grafik berdasarkan data yang dimasukkan. Sehingga konsumen dapat mengisi pulsa meteran prabayar menggunakan *smartphone* tanpa harus menekan langsung *keypad* yang ada pada KWH Meter Prabayar.

METODE PENELITIAN

Pengujian alat Pengisian Kwh Meter Prabayar Menggunakan Plotter 2D dilakukan dengan cara observasi atau pengambilan data langsung di lapangan. Adapun beberapa parameter pengujian yang diamati yaitu : jumlah step yang dibutuhkan untuk mencapai titik koordinat di sumbu x dan sumbu y, *delay* yang terjadi selama pengiriman data token baik dari *smartphone* (tempat menginputkan nomor token) menuju *thingspeak* dan dari *thingspeak* menuju *NodeMCU ESP8266* dan *delay* yang terjadi ketika sensor suara KY-037 mengirim data pembacaan sensor dari *NodeMCU* menuju *thingspeak* dan *thingspeak* menuju aplikasi *smartphone* (menampilkan hasil pembacaan sensor).

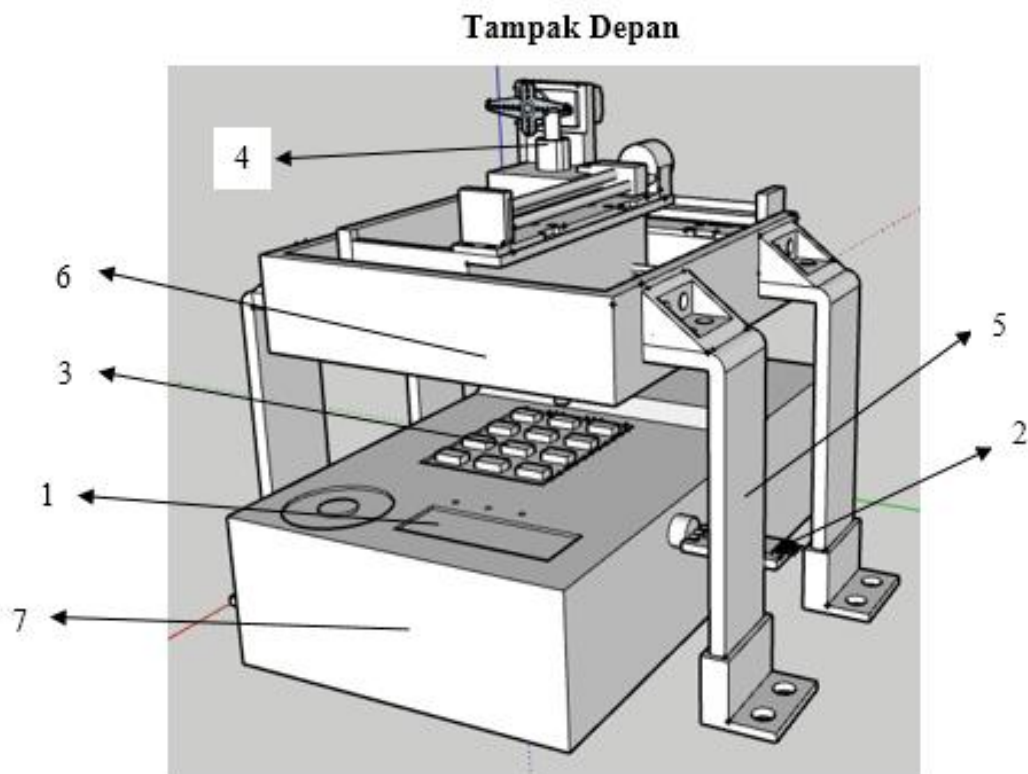
PERANCANGAN

1. Perancangan Hardware



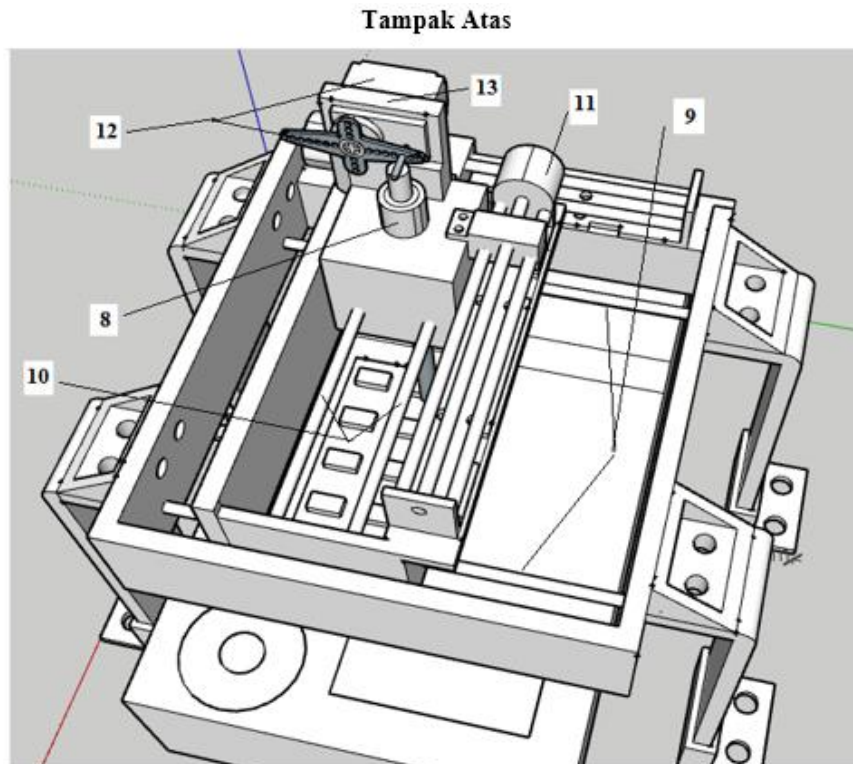
Gambar 63. Rangkaian Elektronika Plotter 2D

Gambar diatas merupakan rancangan perangkat elektronika. Dimana motor stepper pada sumbu X dan sumbu Y digerakkan oleh *driver* A4988 yang berfungsi untuk menguatkan sinyal dari NodeMCU ESP8266 sehingga dapat menggerakkan motor stepper. Dan NodeMCU ESP8266 berfungsi sebagai mikrokontroler yang berfungsi untuk memerintah motor stepper X dan Y serta servo pada sumbu Z (Solenoid) memproses sesuai program yang telah ditanam pada NodeMCU ESP8266 dan berfungsi sebagai pengkoneksian sinyal internet dari ISP (*Internet Service Provider*) kepada client atau pengguna (*Smartphone*) sehingga pengontrolan dapat dilakukan menggunakan *smartphone* via jaringan internet. Sensor suara KY-037 pada rangkaian elektronika disini berfungsi sebagai sensor suara yang membaca suara *buzzer* pada KWH meter Prabayar. Dimana jika *buzzer* pada KWH Meter Prabayar berbunyi maka mengindikasikan bahwa token listrik mau habis dan jika *buzzer* pada KWH Meter Prabayar tidak berbunyi maka mengindikasikan bahwa token listrik masih aman.



Gambar 2. Tampak Depan Desain Mekanik

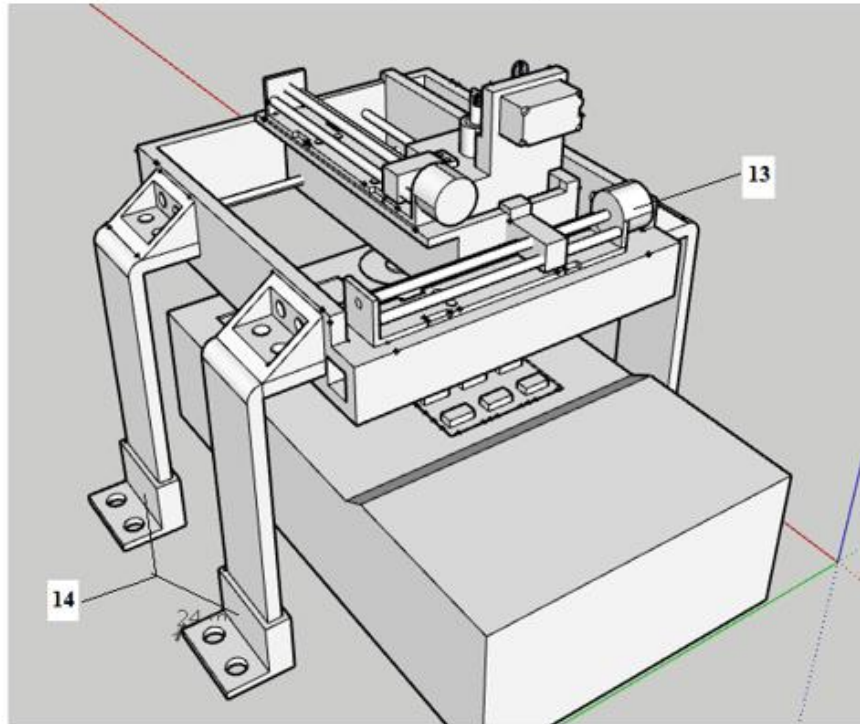
1. LCD
2. sensor Suara KY-037
3. Keypad
4. Sumbu Z (Solenoid)
5. Penyangga
6. Cover Plotter 2D
7. Cover Kwh Meter Prabayar



Gambar 3. Tampak Atas Desain Mekanik

8. Cover atau kedudukan sumbu Z (Selenoid)
9. Besi As Behel diameter 3 mm panjang 110 mm pada sumbu X
10. Besi As Behel diameter 3 mm panjang 103 mm pada sumbu Y
11. Motor Stepper Slider 12V pada sumbu Y
12. Motor Servo pada sumbu Z (Selenoid)

Tampak Belakang



Gambar 4. Tampak Belakang Desain Mekanik

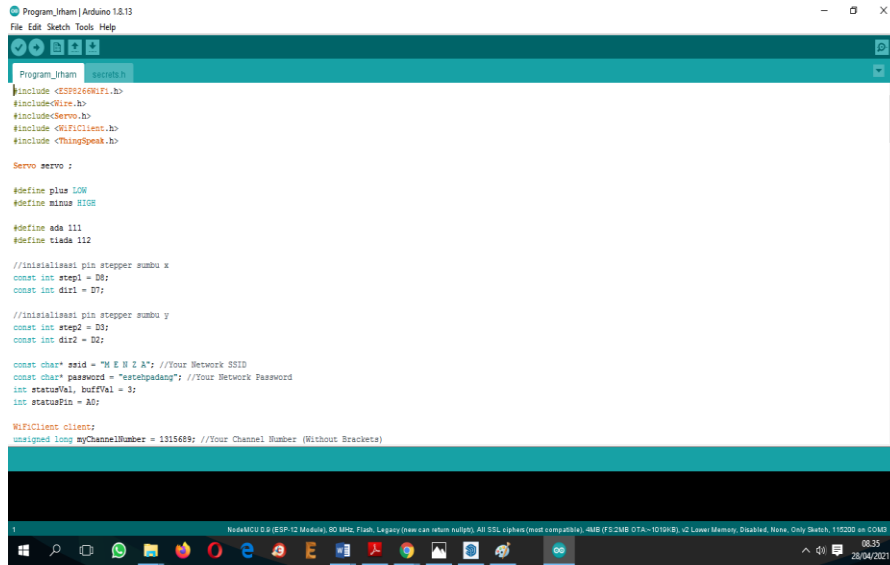
- 13. Motor Stepper Slider 12V pada sumbu X
- 14. Penyangga Plotter pada dinding atau sejenisnya

2. Perancangan Software

Dalam perancangan software menggunakan software Arduino IDE (Integrated Software Development). Dengan menggunakan software Arduino IDE, maka dapat melakukan penulisan program, compile serta upload program ke board NodeMCU ESP8266. Software Arduino IDE (Integrated Software Development) menggunakan bahasa pemrograman C / C++ atau yang dikenal dengan bahasa pemrograman Arduino.

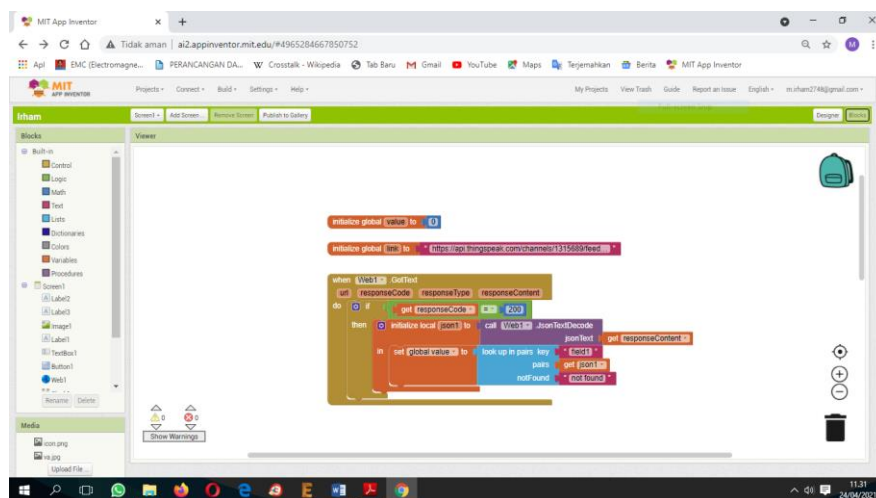


9th Applied Business and Engineering Conference



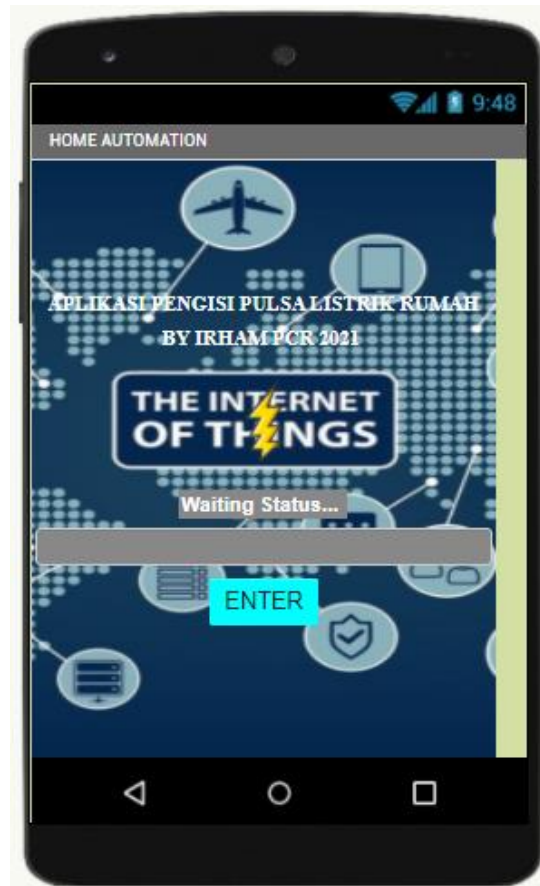
Gambar 5. Software Arduino IDE

Selain itu pada Proyek Akhir ini pada sisi user menggunakan aplikasi *Android* yang dibuat menggunakan App Inventor 2. Dimana App Inventor 2 merupakan IDE generasi kedua dari App Inventor yang dikelola oleh Massachusetts Institute of Technology (MIT). App Inventor 2 berbasis cloud yang diakses menggunakan internet browser. Masuk kategori dalam visual programming, App Inventor 2 menggunakan block puzzle yang disusun untuk menjadi rangkaian kode seperti gambar di bawah ini.



Gambar 6. Tampilan MIT App Inventor

Sebelum melakukan block puzzle tersebut, maka hal yang dilakukan terlebih dahulu adalah mendesain tampilan dari aplikasi yang ingin kita buat. Berikut tampilan aplikasi yang didesain menggunakan App Inventor 2, seperti gambar di bawah ini :



Gambar 7. Tampilan Desain Aplikasi

HASIL DAN PEMBAHASAN

Pengujian ini bertujuan untuk mengetahui alat yang telah dirancang dapat bekerja sesuai dengan apa yang direncanakan. Pengujian dilakukan dengan menguji fungsional alat dan unjuk kerja alat.



9th Applied Business and Engineering Conference

Alat yang dirancang menggunakan mikrokontroler NodeMCU ESP8266 yang sudah dilengkapi modul wifi terintegrasi didalamnya. Sumber daya alat ini menggunakan listrik 12VDC.

Cara kerja alat ini adalah dengan menginputkan nomor token di aplikasi *Android* yang dibuat menggunakan MIT App Inventor yang kemudian data tersebut akan dikirim atau data tersebut terkirim dengan perintah *write* pada MIT App Inventor sehingga data dapat terkirim menuju *thingspeak*, kemudian data yang telah dikirim oleh aplikasi *Android* di *thingspeak* akan disimpan dan divisualisasikan dalam bentuk grafik dan terakhir data tersebut akan di baca atau NodeMCU ESP8266 akan men-*read* data dari *thingspeak* sehingga data akan diproses di NodeMCU ESP8266 yang terhubung dengan internet yang kemudian akan dieksekusi dalam bentuk gerakan sumbu x, sumbu y yang diterima oleh driver A4988 dalam bentuk sinyal- sinyal dari NodeMCU ESP8266 dan sumbu z yang diterima oleh servo.

Berikut adalah hasil pengujian pengiriman data mulai dari *smartphone*, *thingspeak* dan *NodeMCU ESP8266*.

1. Pengujian Aplikasi



Gambar 864. Tampilan Aplikasi Android Sebagai Tempat Penginputan Nomor Token

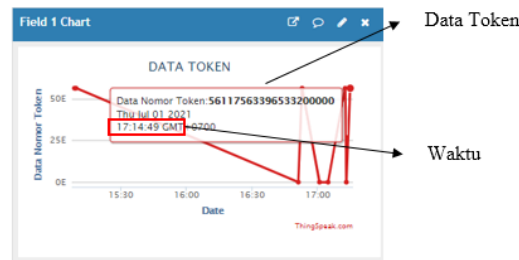
Berdasarkan pengujian aplikasi *Android* pada *smartphone* maka dapat dijelaskan bahwa status “INPUT BERHASIL” (Pada label 1 gambar 4.4) akan muncul setelah pengguna menekan tombol “ENTER” (Pada *button 1* gambar 4.4), tombol *button* tersebut berfungsi untuk merubah status pada label 1 menjadi “INPUT BERHASIL”, status tersebut menandakan proses pengiriman data token dari aplikasi *Android* menuju *Thingspeak* telah berhasil dikirim. Dan dari penjelasan diatas dapat terjadi karena blocks program gambar 9 di bawah ini :

```

when Button1 - .Click
do
  set Web2 - .Uri - to join | " https://api.thingspeak.com/update?api_key=7C3U8D..."
  call Web2 - .Get
  set Label1 - .Text - to " INPUT BERHASIL "
  
```

Gambar 9. Blocks Program

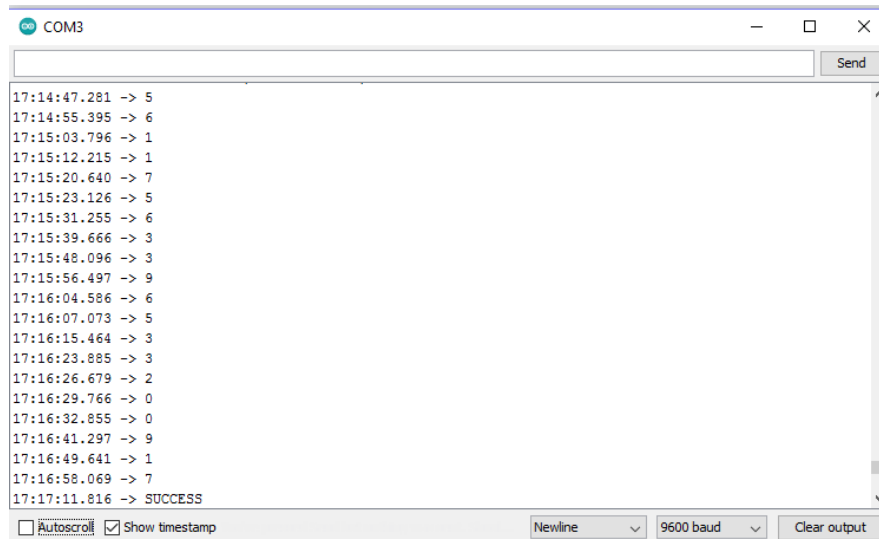
2. Pengujian Thingspeak



Gambar 10. Wujud alat keseluruhan

Berdasarkan pembacaan data token di *Thingspeak* pada gambar 10 diatas maka dapat dianalisa bahwa *Thingspeak* dapat menerima data dengan baik dari aplikasi *Android* jika memiliki jaringan internet yang kuat. *Thingspeak* dapat berkomunikasi dengan *Mit App Inventor* karena *Thingspeak* memiliki fitur API (*Application Programming Interface*) yang berfungsi untuk menyimpan dan mengambil data dari *Mit App Inventor* menggunakan protokol *Hyper Text Transfer Protocol* (HTTP) dan fungsi *write API Keys* pada *blocks* kode *Mit App Inventor 2* adalah mengirimkan data dari publisher (*Mit App Inventor*) ke broker (*Thingspeak*).

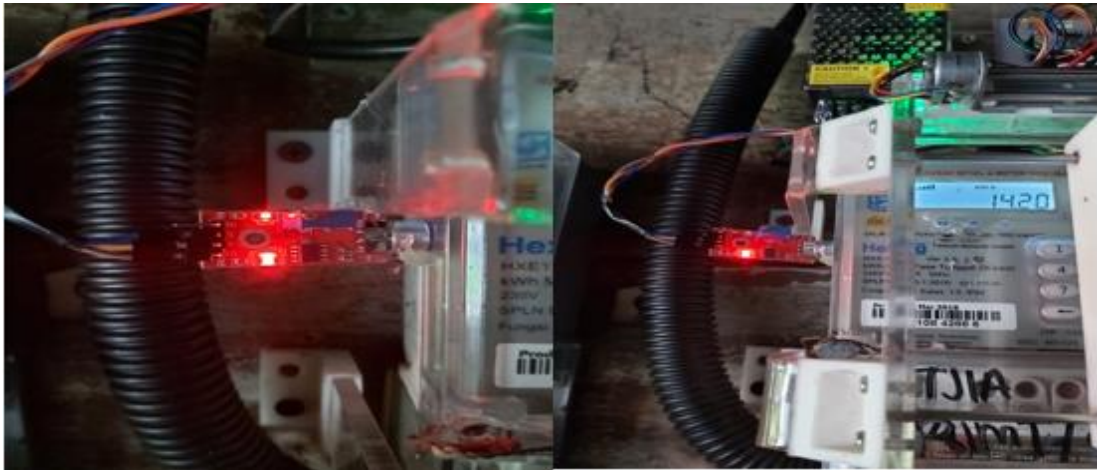
3. Pengujian Serial Monitor NodeMCU ESP8266



Gambar 11. Tampilan Serial Monitor

Dari data diatas maka dapat dianalisa bahwa data dapat terkirim dengan rata-rata *delay* ~7 detik. Hal tersebut dapat dipengaruhi oleh kecepatan internet yang terhubung dengan *Thingspeak* dan selain itu yang membuat data memiliki *delay* ~7 detik adalah karena data yang dikirimkan dari aplikasi *Android* harus melewati *server Thingspeak* terlebih dahulu dan setelah itu data akan di kirim menuju *NodeMCU ESP8266* yang dapat dilihat pada serial monitor pada gambar 4.7. Karena hal tersebut akan membuat pembacaan pada serial monitor pada *Arduino IDE* akan terlambat sekian detik.

Dan kemudian *Sensor Suara K-037* berfungsi untuk membaca suara atau bunyi pada *KWH Meter Prabayar*. Data yang terbaca oleh sensor suara akan dikirim oleh *NodeMCU ESP8266* menuju *thingspeak* dengan menggunakan perintah *write* di *Arduino IDE* kemudian data yang telah berhasil dikirim menuju *thingspeak* akan disimpan dan divisualisasikan dalam bentuk grafik dan data tersebut akan di *read* atau dibaca oleh *MIT App Inventor* menggunakan perintah *read* di *blocks puzzle* di *MIT App Inventor* sehingga hasil pembacaan sensor dapat ditampilkan di aplikasi *Android*.



Gambar 12. Tampilan Sensor Suara Ketika Mendeteksi Suara dan Tidak Mendeteksi Suara

Berikut adalah hasil pengujian pengiriman data sensor suara mulai dari *NodeMCU ESP8266*, *thingspeak* dan *smartphone*.

1. Pengujian di Serial Monitor

```

07:46:09.336 -> Nilai Sensor, Status Value, Status KWH = 910 => 0 => TOKEN MAU HABIS
07:46:27.835 -> Nilai Sensor, Status Value, Status KWH = 890 => 1 => ADA LISTRIK
07:46:46.265 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:46:47.127 -> Nilai Sensor, Status Value, Status KWH = 1017 => 0 => TOKEN MAU HABIS
07:47:05.567 -> Nilai Sensor, Status Value, Status KWH = 877 => 1 => ADA LISTRIK
07:47:23.830 -> Nilai Sensor, Status Value, Status KWH = 928 => 0 => TOKEN MAU HABIS
07:47:42.575 -> Nilai Sensor, Status Value, Status KWH = 845 => 1 => ADA LISTRIK
07:48:01.098 -> Nilai Sensor, Status Value, Status KWH = 822 => 1 => ADA LISTRIK
07:48:01.999 -> Nilai Sensor, Status Value, Status KWH = 866 => 1 => ADA LISTRIK
07:48:02.840 -> Nilai Sensor, Status Value, Status KWH = 874 => 1 => ADA LISTRIK
07:48:03.724 -> Nilai Sensor, Status Value, Status KWH = 860 => 1 => ADA LISTRIK
07:48:04.587 -> Nilai Sensor, Status Value, Status KWH = 907 => 0 => TOKEN MAU HABIS
07:48:23.068 -> Nilai Sensor, Status Value, Status KWH = 1024 => 0 => TOKEN MAU HABIS
07:48:23.919 -> Nilai Sensor, Status Value, Status KWH = 777 => 1 => ADA LISTRIK
07:48:42.530 -> Nilai Sensor, Status Value, Status KWH = 928 => 0 => TOKEN MAU HABIS
07:49:00.958 -> Nilai Sensor, Status Value, Status KWH = 834 => 1 => ADA LISTRIK

07:24:42.939 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:25:01.830 -> Nilai Sensor, Status Value, Status KWH = 897 => 1 => ADA LISTRIK
07:25:02.687 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:25:03.568 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:25:04.543 -> Nilai Sensor, Status Value, Status KWH = 895 => 1 => ADA LISTRIK
07:25:05.470 -> Nilai Sensor, Status Value, Status KWH = 897 => 1 => ADA LISTRIK
07:25:06.347 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:25:07.265 -> Nilai Sensor, Status Value, Status KWH = 896 => 1 => ADA LISTRIK
07:25:08.099 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:25:08.982 -> Nilai Sensor, Status Value, Status KWH = 898 => 1 => ADA LISTRIK
07:25:09.818 -> Nilai Sensor, Status Value, Status KWH = 896 => 1 => ADA LISTRIK
07:25:10.637 -> Nilai Sensor, Status Value, Status KWH = 896 => 1 => ADA LISTRIK
07:25:11.504 -> Nilai Sensor, Status Value, Status KWH = 897 => 1 => ADA LISTRIK
07:25:12.328 -> Nilai Sensor, Status Value, Status KWH = 895 => 1 => ADA LISTRIK
07:25:13.198 -> Nilai Sensor, Status Value, Status KWH = 897 => 1 => ADA LISTRIK
07:25:14.217 -> Nilai Sensor, Status Value, Status KWH = 896 => 1 => ADA LISTRIK
07:25:15.060 -> Nilai Sensor, Status Value, Status KWH = 896 => 1 => ADA LISTRIK
07:25:15.910 -> Nilai Sensor, Status Value, Status KWH = 896 => 1 => ADA LISTRIK
07:25:16.947 -> Nilai Sensor, Status Value, Status KWH = 897 => 1 => ADA LISTRIK
07:25:17.812 -> Nilai Sensor, Status Value, Status KWH = 895 => 1 => ADA LISTRIK
07:25:18.877 -> Nilai Sensor, Status Value, Status KWH = 897 => 1 => ADA LISTRIK

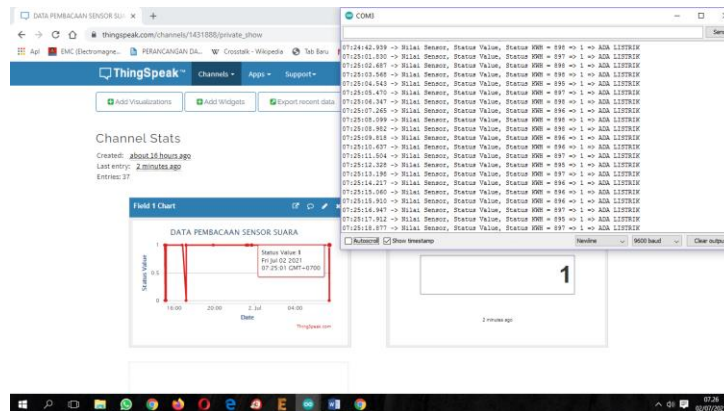
```

Gambar 13. Tampilan Serial Monitor Ketika Mendeteksi Suara dan Tidak Mendeteksi Suara

Berdasarkan gambar 13.1 dan 13.2 diatas, maka dapat dianalisa bahwa status “ADA LISTRIK” jika sensor Suara KY-037 membaca sinyal analog < 900. Dan oleh karena itu dari gambar 13.2 diatas.

Dan status “TOKEN MAU HABIS” jika sensor Suara KY-037 membaca sinyal analog > 900 . Dan oleh karena itu dari gambar 13.1 diatas.

2. Pengujian Thingspeak



Gambar 14. Tampilan Pembacaan Sensor Suara di Thingspeak

Berdasarkan gambar di atas dapat dianalisa bahwa terjadi keterlambatan pengiriman data sensor Suara KY-037 dari serial monitor menuju thingspeak, hal ini dikarenakan keterbatasan thingspeak yang hanya dapat melakukan pembaharuan data atau *message update interval limit* setiap 15 detik sekali pada performa terbaiknya, walaupun pada serial monitor mampu melakukan update setiap 1 detik. Selain itu, Thingspeak sangat membutuhkan internet yang sangat cepat agar dapat terus menerus memperbaharui pembacaan sensor, jika internet tidak pada kecepatan tinggi, *Thingspeak* tidak akan memperbaharui pembacaan sensor.

3. Pengujian Aplikasi Android



Gambar 15. Tampilan Aplikasi Android Ketika Mendeteksi Suara dan Tidak Mendeteksi Suara

Berdasarkan gambar 15.1 dan 15.2 diatas, maka dapat dianalisa bahwa aplikasi pada smartphone akan menerima data dari Sensor Suara KY-037 dan pembacaan sensor dapat dilihat pada serial monitor Arduino IDE. Kemudian data pembacaan sensor akan diteruskan menuju thingspeak menggunakan perintah *write API Keys*, dimana *write API Keys* yang berisi kode unik yang berfungsi agar dapat mengirimkan data pembacaan sensor menuju *Thingspeak* dan data yang ada pada *Thingspeak* akan divisualisasikan dalam bentuk grafik dan selanjutnya akan diteruskan menuju aplikasi smartphone. Smartphone dapat membaca data pembacaan sensor Suara KY-037 menggunakan perintah *read API Keys* yang berisi kode unik yang berfungsi membaca atau mengambil data pembacaan sensor pada *Thingspeak* dan menampilkannya pada aplikasi *Android*. Aplikasi *Android* membaca intruksi “TOKEN MAU HABIS” pada aplikasi (gambar 4.17) jika pada *thingspeak* berlogika “0” dan aplikasi akan membaca intruksi “ADA LISTRIK” pada aplikasi (gambar 4.18) jika pada *thingspeak* berlogika “1”. Aplikasi *Android* pada *smartphone* dapat meng-*read* logika “1” atau “0” pada *Thingspeak* karena pada *Mit App Inventor blocks* telah diprogram program untuk membaca atau men-*read* data logika “1” atau “0” yang ada *Thingspeak*.

SIMPULAN



9th Applied Business and Engineering Conference

Dari penelitian ini yang berjudul Pengisian KWH Meter Prabayar Menggunakan Plotter 2D dapat disimpulkan bahwa data token yang diinputkan pada aplikasi Android dapat diterima di *serial monitor* Arduino IDE NodeMCU ESP8266 terjadi *delay* ~7 detik, terdapat keterlambatan update data di *thingspeak* sekitar 15 detik, aplikasi Android membaca intruksi “TOKEN MAU HABIS” di aplikasi jika sensor Suara KY-037 membaca nilai sebesar > 900 dan aplikasi akan membaca intruksi “ADA LISTRIK” di aplikasi jika sensor Suara KY-07 membaca nilai sebesar < 900 .

Prototipe Pengisian KWH Meter Prabayar Menggunakan *Plotter 2D* telah dirancang dengan baik. Terdapat dua perancangan pada Proyek Akhir yaitu perancangan mekanik yang dirancang menggunakan akrilik dan hasil printer 3D dan perancangan *software* yaitu sebuah aplikasi Android yang dirancang menggunakan *MIT App Inventor* yang mana setiap *interface* pada aplikasi Android dapat digunakan untuk menampilkan status KWH Meter Prabayar, *textbox* yang digunakan sebagai tempat untuk penginputan nomor token dan *button* untuk mengirimkan instruksi data angka nomor token yang diketikkan di *textbox*.

DAFTAR PUSTAKA

- Mulyana Asep, Tengku Ahmad Riza, Dadan Nur Ramadan, Muhammad Dzakwan Falih, Universitas Telkom, and Terusan Buahbatu. 2017. “*Sistem Pengisian Pulsa Pada Kwh Meter Prabayar Menggunakan Ponsel Pulse Filling System on Prepaid Electrical Energi.*” *Jurnal Elektro Telekomunikasi Terapan*, 560–69.
- Nugroho, Ernes Cahyo, Agung Nugroho, and Isti Hendriyanto. 2019. “*Prototipe Mesin CNC 2D Berbasis Arduino Uno.*” *Go Infotech: Jurnal Ilmiah STMIK AUB* 25 (1): 43. <https://doi.org/10.36309/goi.v25i1.103>.
- scheiner. n.d. “*Apa Itu Servo Motor Dan Kegunaannya?*” <https://www.se.com/>. <https://www.se.com/id/id/faqs/FA374507/>.
- shalehmms. 2018. “*Serial Monitor Pada IDE Arduino.*” 2018. <https://shalehmms.wordpress.com/2018/04/02/serial-monitor-pada-ide->



9th Applied Business and Engineering Conference

arduino/#::~text=Serial Monitor pada IDE Arduino adalah sebuah fasilitas yang dapat,sensor suhu di serial monitor.