# Electricity Token Charging Based on Internet of Things

**Hendro Darmono[1], Waluyo[2], Nugroho Suharto[3], Koesmarijanto[4]**

[1,2,3] Digital Telecommunication Network Study Program,
Department of Electrical Engineering, State Polytechnic of Malang, 65141 INDONESIA
[4] Telecommunication Engineering Study Program,
Department of Electrical Engineering, State Polytechnic of Malang, 65141 INDONESIA

[1]hendrodarmono@polinema.ac.id, [2]waluyo@polinema.ac.id , [3]nugroho.suharto@polinema.ac.id, [4]koesmarijanto@polinema.ac.id

***Abstract*—Many prepaid electricity meters have been implemented for new and existing customers, and these meters offer advantages over conventional meters. Purchasing token pulses on prepaid electricity meters can be done anytime and anywhere, making it more practical and easier for customers to recharge electricity vouchers. Customers can purchase electricity tokens at available counters or through an application, and they can add KWh inventory as needed. The use of electricity tokens depends on the power consumption in each household, and the remaining KWh can be viewed on the screen of the prepaid meter. However, customers must enter a 20-digit code every time they recharge an electricity voucher. This paper developed the Internet of Things-based electricity token charging system is designed to address the limitations of manual token charging, which requires users to be at home. This system enables remote token charging via a web application, equipped with a sound sensor that sends notifications to a smartphone when the electricity meter alarm sounds, indicating that the KWh balance is nearly depleted. By using an ESP32 microcontroller and relay module, the system controls solenoids to press the numbers on the prepaid meter keypad. This system can operate anytime as long as it is connected to the internet, achieving high accuracy in the token charging process.***

***Keywords*— ESP32, Token charging, Sound sensor, IoT**

## I. INTRODUCTION

Electric tokens are also called smart electricity meters. This smart electricity is called prepaid electricity, which means we have to pay before using electricity. After paying we will get a code that is entered into the prepaid electricity meter so that it can distribute a certain amount of electricity (kWh) to the consumers. The token consists of 20 unique numbers, only for certain prepaid meter numbers. Charging prepaid electricity pulses that exist today has the disadvantage that if the electricity pulses run out, it can only be known if the owner is at home by hearing the sound of an alarm on the prepaid meter. In addition, filling in the electricity token code must be done by manually by pressing the keypad numbers on the prepaid meter, so that if the house owner is left for a long time, the electricity will go out until the homeowner comes and refills the electricity token.

In this study, an IoT-based electricity token charging system is proposed to make it easier for users to top up electricity token pulses. In this system, users no longer need to press the prepaid meter keypad when filling up electricity token pulses. Users can fill in electricity tokens by buying token pulses at counters or e-commerce applications, and then the number voucher code obtained can be entered through the token.datakrip.com website by logging in to the user and typing the voucher code in the column provided. With this, the solenoid will work to press the numbers on the prepaid meter keypad.

Previous research [1] conducted by Muhammad Syafar with the title "Long Distance Electricity Voucher Filling System Via Microcontroller-Based SMS" has the same goal, namely to make it easier to fill in electricity vouchers wherever and whenever the user is. This system was built using SIM 800 to receive SMS in the form of an electricity voucher code. This research uses a servo motor controlled by a microcontroller to press the buttons on the keypad of an electric meter. This mechanism requires a relatively long time and a larger space for placing the device on the electric meter. The success of the servo motor in pressing the button is not known by the device user.

Whereas research [2] conducted by Sudimanto with the title "Token Electricity Using SMS (Short Messages Services)" has a way of working that starts when checking the remaining credit quota sent by the system via Wavecom, the owner will receive an SMS warning that the remaining quota is approaching the minimum limit. In modeling the automatic top-up of data quota, communication between the Wavecom modem and the microcontroller is required to support the process of sending and receiving SMS messages. The communication used between the Wavecom and the microcontroller is serial communication. The success of the device in completing the token top-up has not yet been clearly or effectively communicated.

Then research [3] conducted by Samuel Manullang with the title "Robot Filling in Electricity Token Numbers" uses a microcontroller as a control center receiving tokens and controlling solenoids. This robot is also designed to replace the human position in pressing electricity token numbers. Research uses Blynk as an application that is used as an intermediary between the microcontroller and the Internet of Things.

Ridho, Devi A, Sri Lestari Department of Airport Electrical Engineering, Surabaya Aviation Polytechnic [4] also conducted research using current sensors that work detect the use of the kWh meter then sent via Arduino to displayed on android. From android send commands via the bluetooth module to arduino that will be sent to gsm modem in the form of an AT command. Command AT Command on this GSM modem will be communicated through a serial intermediary RS232 communication. Next arduino will send commands to other components in order to work according to their respective functions.

Hasna naufal Giarniasih Faculty of Engineering/Electrical Engineering Mercu Buana University Jakarta, [5] also researched Smart Monitoring Systems and charging prepaid electricity pulses via smartphone namely a monitoring and recharging system electricity via smartphone without having to do it on kwh meter again and did not let the house go out to linger. Besides that, you can also make a notification if the electricity pulse (token) will soon run out or the remaining quota is close to the minimum limit through current readings on the ACS712 sensor which works with a voltage of 5 watts processed with NodeMCU microcontroller and sends notifications to the mobile phone in the MIT App Inventor application form.

Furthermore, research was also carried out by Dedi Triyanto, Gajah Mada University [6], namely making a prototype monitoring system and charging prepaid electricity tokens using Arduino Uno web-based to make it easier for users to do remote automatic charging of tokens and find out token status information remaining on the kWh meter.

In terms of security to reduce human injury when charging electricity tokens. This system was created primarily to make it easier for users to fill in electricity tokens efficiently and practically. The charging of this electricity token can also be monitored via a live streaming camera located in front of the prepaid meter screen via the website.

## II. RESEARCH METHODS

The type of research that is being carried out belongs to the type of design, this type of research has stages such as literature studies, making tools, analyzing, and at the same time making conclusions. The block diagram that will be used for electricity token changing is shown in Figure 1. The resulting work can be summarized as follows:

**Step 1: Block Diagram System**

The first step involves describing the block diagram of the electricity token changing and outlining the parameters that need to undergo testing.

**Step 2: Hardware Design**

It involves designing the keypad controller circuit using the ESP32 microcontroller.

**Step 3: Software Design**

Involves designing the coding for the microcontroller's functions and creating the web design.

**Step 4: Fabrication Process**

This step Implements the device according to the plan, including both hardware and software components.

**Step 5: System Testing**

Involves testing the functionality of the system, including both hardware and software functions.

**Step 6: Performance Measurement and Analysis**

Performance measurement is necessary to determine the level of accuracy, while analysis is required to conclude.

**Step 7 : QoS Test**

QoS (Quality of Service) parameters are used to generate information in the form of:

- The time taken by packet data is calculated from the time of delivery by the transmitter until it is received by receivers (throughput).
- Differences between arrival times packet at the destination terminal (delay/latency).
- The number of packets lost during the process of transmission to the destination (packet loss).

The number of bits received successfully per second through a system or medium communication (the actual ability of a network to send data) (jitter)

### *A. System Block Diagram*

The system works as follows: When the kWh meter emits a buzzer sound, indicating that the electricity token needs to be recharged because the token balance is at its lowest limit, the sound sensor detects the alarm sound from the prepaid meter. The analog data from the sound sensor is then sent to the ESP32 board. The data is processed by the ESP32 board and sent to the database. The website queries the data in the notification table with ID 1; if it indicates a value of 1 or ON, a notification will appear on the user's phone. However, if the database table shows a value of 0, the notification will not be forwarded to the user. The user then logs in to the website token.dataskripsi.com by entering the username "admin" and password "admin." The user types in the 20-digit code from the purchased electricity token voucher, which is then sent to the token table in the database. Next, the relay module connects to the solenoid to press the keypad on the prepaid meter, and the user can monitor the meter display through a camera.

The following Figure 1 is an image of a block diagram of charging Internet of Things (IoT) based electricity tokens consisting of a website, sound sensor, lm2596 step-down, step-up, selenoid, relay and esp 32 cams that are interconnected to function to make it easier for users to top up electricity token pulses.

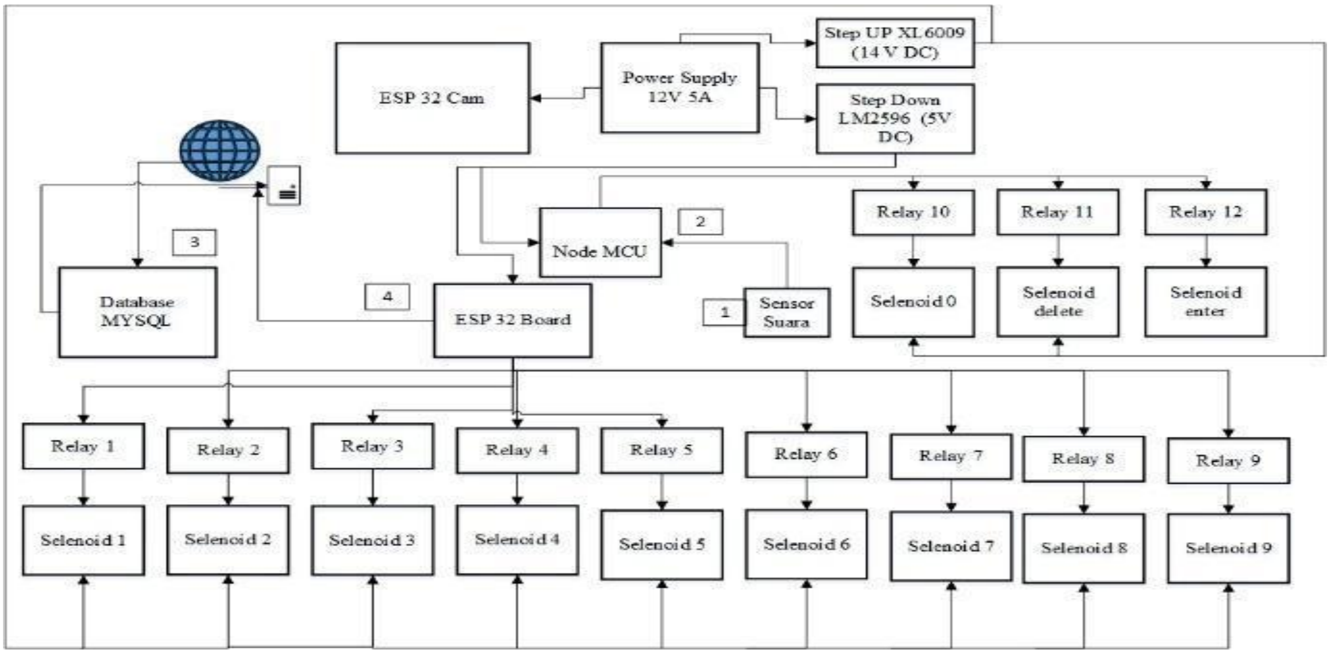



Figure 1. Block diagram system electricity token charging

Figure 1 presents an overview of the system, where the microcontroller is a chip capable of receiving input signals, processing them, and producing output signals according to the program loaded into it. The microcontroller's input signals come from sensors that gather information from the environment, while the output signals are directed to actuators that can impact the environment. Simply put, the microcontroller functions as the brain of a device, enabling interaction with its surroundings [7].

For real-time monitoring, an ESP32-Cam microcontroller is used. The ESP32-Cam is a platform that facilitates real-time monitoring by incorporating a camera and Wi-Fi modules. To configure the ESP32-Cam, an FTDI USB to TTL adapter is required, which connects the camera module to a computer or laptop. To upload the program to the ESP32-Cam, open-source software like Arduino IDE is utilized. Arduino IDE is software designed for developing microcontroller applications, including programming, compiling, and uploading. It also features a serial terminal, which simplifies communication between the user and the computer via USART/RS232. Through this setup, a security and real-time monitoring system is established [8].

The relay consists of three pins, a normally open pin, a normally closed pin, a common pin and a coil. When the coil is supported on the resulting magnetic field the contacts are connected to each other. Regardless of the relay channel number, the pin configuration is the same for each channel except for the power (VCC and GND) pins for the board itself. Input signal pin (IN) for each relay[9].

Figure 2 is a series of tools there are several parts. The parts of this tool kit consist of iron planks (to install esp 32 cam), electric token machine [10][11][13] (electric token filling system media), sound sensor (detect sound meter alarm), esp 32 cam (charging monitoring), solenoid [14] (to perform emphasis on prepaid meter keypad numbers), and PVC housing (as a solenoid housing).

### *B. Software Design*

The kWh meter emits a buzzer sound, signaling that the electricity token needs to be recharged as the token balance has reached a low limit [15]. Then, the analog data from the sound sensor is sent to the ESP32 board, where it is processed and forwarded to the database. The website queries the notification data table at ID 1; if it indicates a digit 1 or ON, a notification will appear on the user's mobile phone. However, if the database table value is 0, the notification will not be sent to the user

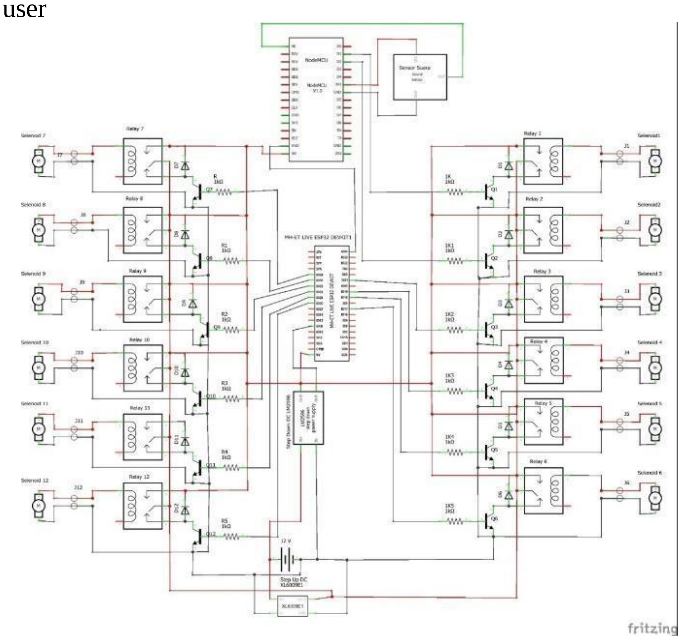



Figure 2. Hardware Design

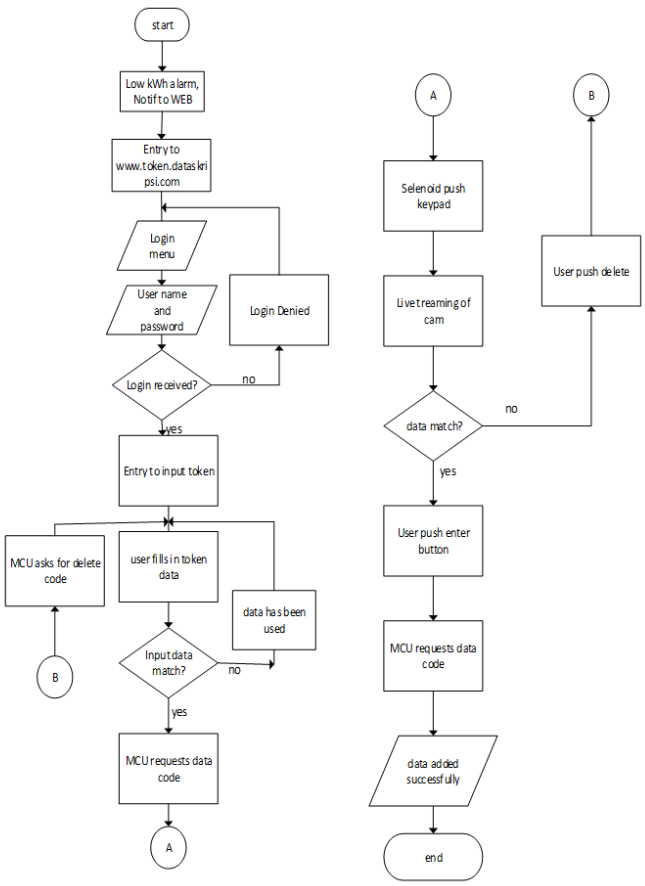



Figure 3. Software Flow Diagram

Next, the user logs in to the website token.dataskripsi.com by entering the admin username and password. If the login is successful, the user proceeds to the token input menu. The user enters the 20-digit voucher code. The database will check this 20-digit code. If the code has already been used, a notification will appear stating that the electricity token has already been entered and cannot be reused.

If the token is accepted, the ESP32 requests data from the token table and downloads it via an HTTP request. This data, which is in char or ASCII code (a code only understood in machine language), is then converted to an integer data type, making it understandable to humans. The ESP32 sends a signal to the corresponding pin based on the relay number, determining whether the relay should be activated or deactivated according to the previously converted data type, with a 2-second delay between each relay activation.

When the relay switches on or off, the solenoid will press the keypad according to the relay activation. The ESP32 Cam provides live streaming on the homepage/dashboard to ensure whether the code has been entered correctly or not. If it is incorrect, there is an option to press delete to retry entering the wrong code. If it is correct, press enter, and the token will be successfully loaded.

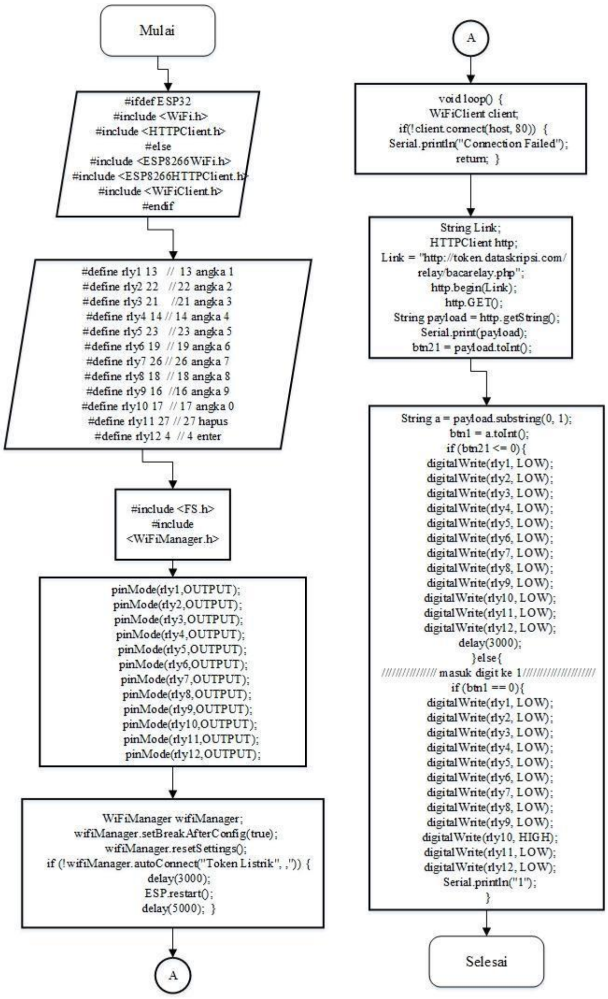
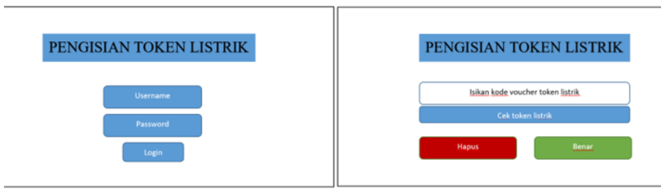



Figure 4. Short Coding for the System

The display design of this application is a display that will be made on an Internet of Things (IOT) based electricity token charging system. Starting from the login page, the electricity token filling page, live streaming electricity token charging to find out pressure errors when charging, the electricity token charging history data page, and the edit profile page as shown in Figure 5.



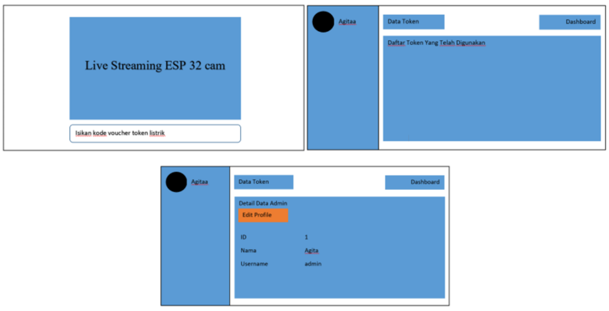



Figure 5.Website Design

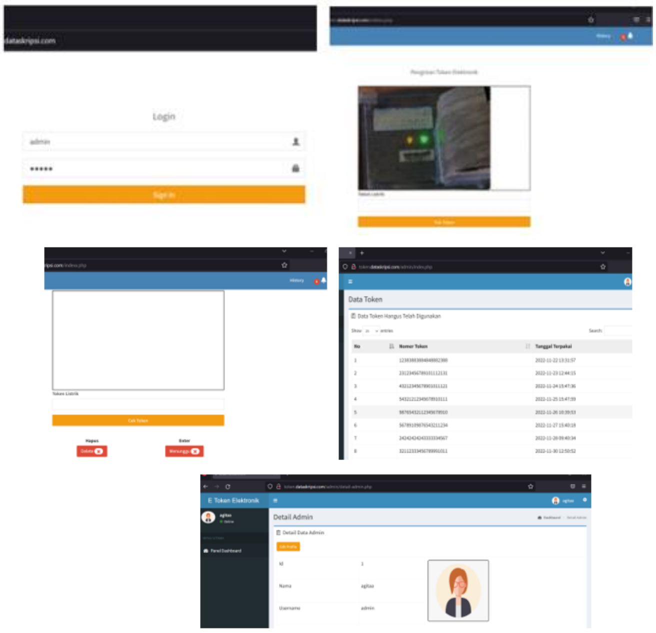



Figure 6. Website Design Result

### C. *Fabrication Process*

Figure 7 is a series of tools there are several parts. The parts of this tool kit consist of iron planks (to install esp 32 cam), electric token machine (electric token filling system media), sound sensor (detect sound meter alarm), esp 32 cam (charging monitoring), solenoid (to perform emphasis on prepaid meter keypad numbers), and PVC housing (as a solenoid housing).

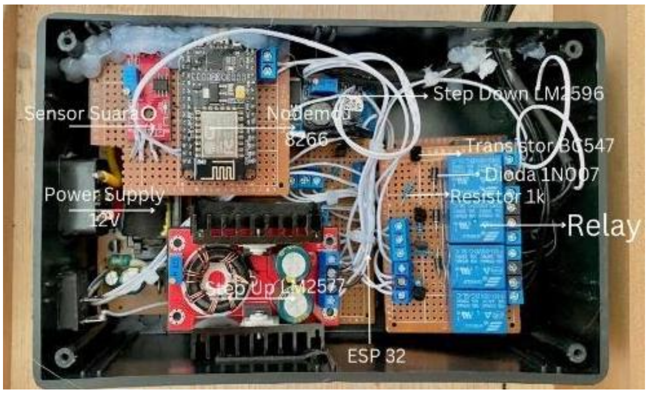



Figure 7. Whole Suites


Figure 8. Prototype Hardware

The result of this hardware design is the result/display of the hardware from filling in Internet Of Things (IOT) based electricity tokens. Starting from the overall circuit, displaying the results of the system design on the prepaid meter machine, such as in Figure 8 and 9

## III. RESULTS AND DISCUSSION

The design results were divided into 2 parts, namely the software design in the form of a finished application, and the hardware/hardware design that has been made.

### A. *The Results of this Sofware (Application)*

The results of this software (application) design are the results of an application from an Internet of Things (IOT) based car-sharing system. Starting from the login page, the electricity token filling page, live streaming of electricity token charging to find out pressure errors when charging, the electricity token charging history data page, and the edit profile page as shown in Figure 7.

### B. *Token Data Testing to Database*

In this test the data taken is success in filling in the electricity token number. The electricity token number will be entered via the token.dataskripsi.com website. The testing was conducted for each digit 20 times, from 0 to 9. This testing was intended to determine the success rate of sending digits via the mobile phone and their response by the solenoid module, as well as to assess the accuracy of the data displayed on the meter screen, as shown in Figure 10

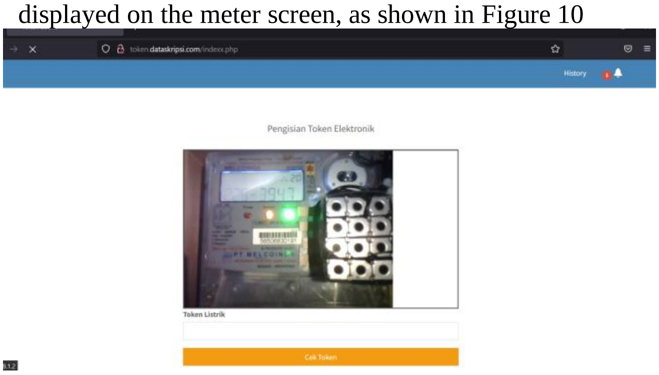



Figure 9. Solenoid Test Display

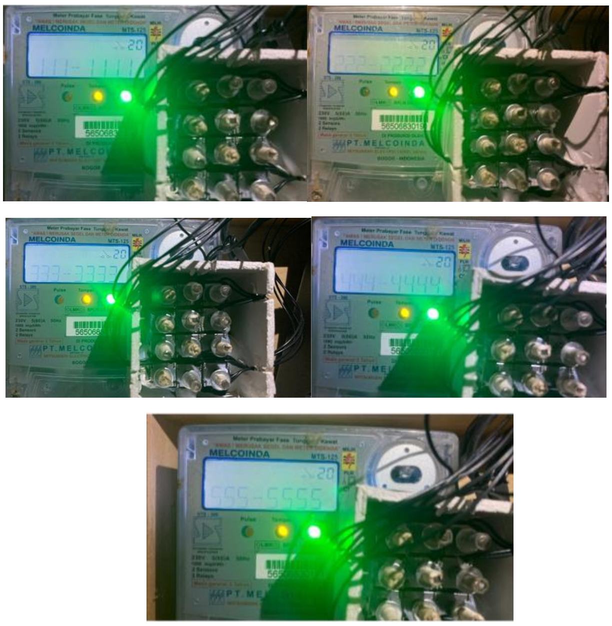
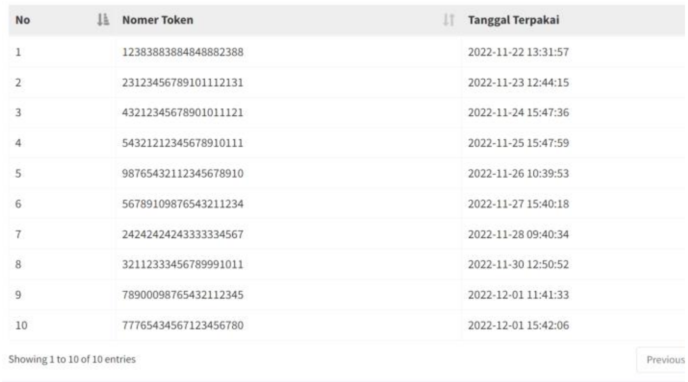


Figure 10. Solenoid Test



| No | Nomer Token | Tanggal Terpakai |
|---|---|---|
| 1 | 12383883884848882388 | 2022-11-22 13:31:57 |
| 2 | 23123456789101112131 | 2022-11-23 12:44:15 |
| 3 | 43212345678901011121 | 2022-11-24 15:47:36 |
| 4 | 54321212345678910111 | 2022-11-25 15:47:59 |
| 5 | 98765432112345678910 | 2022-11-26 10:39:53 |
| 6 | 56789109876543211234 | 2022-11-27 15:40:18 |
| 7 | 24242424243333334567 | 2022-11-28 09:40:34 |
| 8 | 32112333456789991011 | 2022-11-30 12:50:52 |
| 9 | 78900098765432112345 | 2022-12-01 11:41:33 |
| 10 | 77765434567123456780 | 2022-12-01 15:42:06 |

Showing 1 to 10 of 10 entries

Previous

Figure 11. Token Data Testing To Database

## C. *Selenoid Testing Results*

The function of the solenoid button was tested to determine the success rate of mechanical pressing on the meter keypad. Two hundred numbers were tested randomly, and one number did not appear on the meter as shown in Table 1 and Figure 11. There are several possible causes for this, one of which is that the solenoid mechanism did not function properly due to insufficient electric current to activate the solenoid to press the keypad. Another possible cause is that the microcontroller could not adequately respond to the speed of digit changes when numbers were pressed on the mobile phone screen. A third possibility that may cause data transfer failure is an unstable network connection, a viable solution involves shortening the cable length for each solenoid by replacing it with circuit pathways on a PCB. Additionally, enhancing the battery's power capacity can further improve performance.

Although this is considered unlikely. The calculated success rate of the solenoid pressing the keypad reached 99%, but when evaluating the success rate of the 20-digit token entry, a success rate of 90% was achieved. Compared to previous research, the performance of this device is significantly better, although for commercial purposes, improvements are needed, including redesigning the system plan.


Figure 12. Live Streaming of Solenoid Process to the Meter Keypad

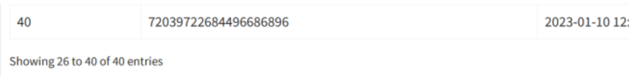

Figure 13. Solenoid Suppresses the Numbers Sent to the Database

TABLE I
CONVERSION POTENTIAL OF SOME RADIONUCLIDES

| No | Number pressed on handphone | The number displayed on the meter | Result |
|---|---|---|---|
| 1. | 12383883884848882388 | 12383883884848882388 | Match |
| 2. | 23123883884848882388 | 23123883884848882388 | Match |
| 3. | 43212345678901011121 | 43212345678901011121 | Match |
| 4. | 54321212345678910111 | 54321212345678910111 | Match |
| 5. | 98765432112345678910 | 98765432112345678910 | Match |
| 6. | 56789109876543211234 | 56789109876543211234 | Match |
| 7. | 24242424243333334567 | 24242424243333334567 | Match |
| 8. | 32112333456789991011 | 32112333456789991011 | Match |
| 9. | 78900098765432112345 | 78900098765432112345 | Match |
| 10. | 77765434567123456780 | 7765434567123456780 | No Match |

### *D. System Test Results*

Electricity Token Charging is designed to be able to suppress electricity token numbers on prepaid meters without using human power. In this case the tests carried out are the connection of the electricity token with the database, the accuracy of the solenoid in pressing numbers on the prepaid meter keypad entered by the user on the website, and the connection of the camera to the website when the solenoid presses.



| | | |
|---|---|---|
| 40 | 72039722684496686896 | 2023-01-10 12: |

Showing 26 to 40 of 40 entries

Figure 15. The numbers typed on the website are sent to the database

The picture above shows that the numbers typed on the website when filling in electricity tokens were successfully sent to the history database and pressed as many as 20 digit numbers by the solenoid to the prepaid meter keypad and live streaming can run which indicates that the esp 32 cam can be connected properly.

### *E. Quality of Service (QoS) Results*

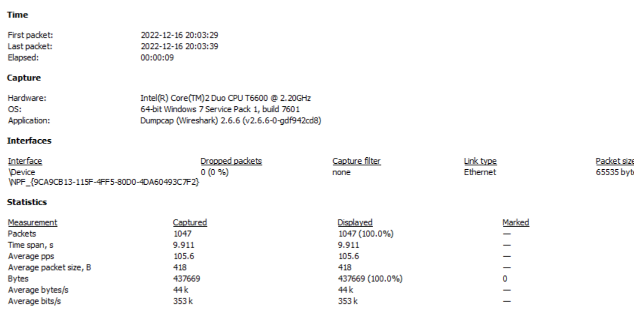
QoS parameters provide valuable insights, including the time a data packet takes to travel from the transmitter to the receiver (throughput), the variation in packet arrival times at the destination (delay/latency), the percentage of packets lost during transmission (packet loss), and the number of successfully received bits per second (jitter), which reflects the network's capacity for data transmission. These QoS tests, conducted using the Wireshark software, assess several key parameters such as throughput, packet loss, delay, and jitter, with the results detailed below.

Throughput Test Results. The Calculation Results **:**

$$\text{Throughput} = \left(\frac{received\ data\ packets}{available\ bandwidth}\right) [8]$$

Time Span = Hasil Bytes

Troughput = (Bytes : Time span, s ) x 8
= (437669 : 9.911) x 8
= 44159,923 bytes/s x 8
= 353279,386 Bytes
= 353,279386 Kilo Bytes (KB)



**Time**

First packet: 2022-12-16 20:03:29
Last packet: 2022-12-16 20:03:39
Elapsed: 00:00:09

**Capture**

Hardware: Intel(R) Core(TM)2 Duo CPU T6600 @ 2.20GHz
OS: 64-bit Windows 7 Service Pack 1, build 7601
Application: Dumpcap (Wireshark) 2.6.6 (v2.6.6-0-gdf942cd8)

**Interfaces**

| Interface | Dropped packets | Capture filter | Link type | Packet size |
|---|---|---|---|---|
| \Device \NPF_{9CA9CB13-115F-4FF5-80D0-4DA60493C7F2} | 0 (0 %) | none | Ethernet | 65535 byt |

**Statistics**

| Measurement | Captured | Displayed | Marked |
|---|---|---|---|
| Packets | 1047 | 1047 (100.0%) | — |
| Time span, s | 9.911 | 9.911 | — |
| Average pps | 105.6 | 105.6 | — |
| Average packet size, B | 418 | 418 | — |
| Bytes | 437669 | 437669 (100.0%) | 0 |
| Average bytes/s | 44 k | 44 k | — |
| Average bits/s | 353 k | 353 k | — |

Figure 15. Throughput Test Results

Packet Loss Test Results. The results calculation percentage of packet loss **:**

$$\text{Packet Loss} = \left(\frac{packages\ sent - packets\ received}{packages\ sent}\right) [8]$$

[((Packages sent – packets received): packages sent) x 100]
= ((1047 - 1047: 1047) x 100
= 0 % (0 packet loss)

Delay Test Results. To calculate this delay, data is taken from application testing using wireshark and data on wireshark is exported in CSV/excel form.

The formula for calculating the delay :

$$\text{Average delay} = \left(\frac{total\ packets\ delay}{total\ packets\ received}\right) [8]$$

time 2 - time 1 = delay result
total delay = sum(block all delay results then enter)
average delay = (total delay / number of packets then enter)
delay 1 = total delay of first line - last second line enter

Delay Calculation Results:
Total delays = 9.911229
Average delay = 9.911229 s x 1047 = 0.009475 ms

Jitter Calculation Results. To calculate this jitter, data is taken from application testing using wireshark and data on wireshark is exported in CSV/excel form.

jitter = delay 2 -delay 1 then enter

$$\text{Average jitter} = \left(\frac{total\ packets\ delay}{total\ packets\ received}\right) [8]$$

Total jitter = sum (block all jitter results then enter)
Average jitter = (total jitter / number of packets then enter)

Jitter Calculation Results:
Total jitter = 10.692214
Average jitter = 0.010231784 s = 10.23 ms

Analysis of QoS: (1) Throughput. In this throughput test, the value is 353.279386 Kbits/s; (2) Package Loss. Analysis of the packet loss test results obtained 0%, which is perfect. This can happen because of a stable internet network; (3) Delays. Analysis of the results of the average delay can be concluded that the test is included in the good category with an index number of 3. Because the delay of 9.47 ms is in the range of 0-75 ms. This is because the internet connection is fast, the domain hosting is premium, and also the conditional function of this laptop works well; (4) Jitters. Jitter analysis obtained an average result of 10.23 ms with a very good latency category. This can happen because the network capabilities are very good.

## IV. CONCLUSION

This automatic electricity token charging system successfully facilitates remote token charging with a relatively high success rate. The system provides notifications when the token is nearly depleted and allows users to recharge via a website with a reliable mechanism. Nonetheless, there are opportunities for improvement in power strength and network connection stability to further enhance the system's accuracy and reliability.

## REFERENCES


[1] Sudimanto, 2017. Pengisian Pulsa (Token) Listrik Menggunakan SMS (*Short Messages Services*), Bandung: Sekolah Tinggi Manajemen Informatika dan Komputer LIKMI

[2] Syafar, A. M. (2017). Sistem Pengisian Voucher Listrik Jarak Jauh Via SMS Berbasis Mikrokontroller. *Jurnal INSTEK (Informatika Sains dan Teknologi)*, *2*(2), 141-150.

[3] Samuel Manullang, 2020. Robot Pengisi Token Listrik, Politeknik Caltek Riau

[4] Arisandi, D., Lestari, S., & Ridho, R. (2018). Rancang Bangun Alat Pengisi Pulsa Otomatis Pada KwH Meter Digital Prabayar Satu Fasa Menggunakan Android Berbassis Mikrokontroler. *In Prosiding SNITP (Seminar Nasional Inovasi Teknologi Penerbangan)* (Vol. 2, No. 1).

[5] Naufal Giarniasih, H., & Kadarina, T. M. (2019). Smart Sistem Untuk Pemantauan Dan Pengisian Pulsa Listrik Pra Bayar Via Smartphone. *Jurnal Teknologi Elektro*, *10*(2), 97-102.

[6] Destiarini dan P. W. Kumara, “Robot Line Follower Berbasis Mikrokontroller Arduino Uno ATmega328,” *J. Informanika*, vol. 5, no. 1, hal. 18–25, 2019.

[7] 2011 Bruce, “Pengertian Mikrontroller,” *J. Chem. Inf. Model.*, vol. 53, no. 9, hal. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.

[8] D. Setiawan, H. Jaya, dan S. Nurarif, dkk, “Implementasi ESP32-Cam Dan Blynk Pada Wifi Door Lock,” *J. Sci. Soc. Res.*, vol. 5, no. 1, hal. 159–164, 2022.

[9] FEC, “Relay modules 1-channel features,” *Futur. Electron. Corp.*, no. 5 V, hal.1–2,2019,[Daring].Tersediapada:http://fecegypt.com/uploads/ dataSheet /1522335719_relay module.pdf.

[10] K. Wittek, S. Finke, N. Schelte, N. Pohlmann and S. Severengiz, "A Crypto-Token Based Charging Incentivization Scheme for Sustainable Light Electric Vehicle Sharing," *2021 IEEE European Technology and Engineering Management Summit (E-TEMS)*, Dortmund, Germany, 2021, pp. 136-140, doi: 10.1109/E-TEMS51171.2021.9524902

[11] T. Makonese, S. Mufukare and S. von Solms, "Perception of the use of pre-paid electricity meters in Chidzidzi, Mutoko, Zimbabwe," *2018 International Conference on the Domestic Use of Energy (DUE)*, Cape Town, South Africa, 2018, pp. 1-5, doi: 10.23919/DUE.2018.8384379

[12] D. Liu, X. Zeng and Y. Wang, "Edge-Computing-Driven Autonomous Ubiquitous Internet of Things in Electricity: Architecture and Challenges," *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*, Changsha, China, 2019, pp. 456-461, doi: 10.1109/EI247390.2019.9062125.

[13] Y. -J. Wu *et al*., "IoT Cloud-Edge Reconfigurable Mixed-Signal Smart Meter Platform for Arc Fault Detection," in *IEEE Internet of Things Journal*, vol. 10, no. 2, pp. 1682-1695, 15 Jan.15, 2023, doi: 10.1109/JIOT.2022.3210220.

[14] A. Vishwanath, V. Chandan and K. Saurav, "An IoT-Based Data Driven Precooling Solution for Electricity Cost Savings in Commercial Buildings," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7337-7347, Oct. 2019, doi: 10.1109/JIOT.2019.2897988

[15] M. U. Saleem, M. R. Usman, M. A. Yaqub, A. Liotta and A. Asim, "Smarter Grid in the 5G Era: Integrating the Internet of Things With a Cyber-Physical System," in *IEEE Access*, vol. 12, pp. 34002-34018, 2024, doi: 10.1109/ACCESS.2024.3372379.