ISSN 2355-3286

Light Emiting Diode Control System based on the Microcontroller and Smartphone Application

Fahmy Rinanda Saputri[1], Lucia Blanca Rahawarin[2]
[1,2] Engineering Physics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1]fahmy.rinanda@umn.ac.id, [2]lucia.rahawarin@student.umn.ac.id

Accepted 07 October 2022
Approved 22 November 2022

***Abstract*— Nowadays, fulfilling the needs of human life requires costs. If the daily needs are under control, then the expense spent to meet the daily needs will also be controlled. In everyday electricity usage, the lights may turn on when the user does not need them. This incident resulted in a waste of electricity and costs. This research designs a Light Emitting Diode (LED) light control system using a microcontroller and smartphone application. The features available in the lamp control application consist of Off and On, Timer, Schedule, Electrical Energy Monitoring, and Automation based on Electricity Billing Budgeting. The tools and software used in the design of the lamp control are: Node MCU ESP8266 version 2, SRD-05VDC-SL-C relay module, ACS712 current sensor, Arduino IDE, MIT App Inventor, and ThingSpeak. The time delay from pressing the ON/OFF button on the smartphone until the light turns on is about 1-6 seconds. The error value for calculating the lamp power with the ACS712 current sensor against the lamp power from the specifications on the light box is 1.3%.**

***Index Terms*— Arduino IDE; Light Emitting Diode (LED); microcontroller; MIT App Inventor; ThingSpeak.**

I. Introduction

Based on the State Electricity Company (PT. PLN)'s Electricity Tariff Adjustment, the electricity tariff from October to December 2021 is Rp. 1,444,70/kWh [1]. In the daily electricity, lights may turn on even when the lights are not needed. This incident resulted in a waste of electricity and costs because the user still had to pay for the electricity used.

Electrical energy consumption at Lucia's house was recorded every day for four weeks. The electrical energy consumption data is visualized in the electricity consumption graph. The graph of electricity consumption per day is the total electrical energy consumption in 1 day. The graph of electricity consumption per 7 days is the electrical energy total consumption in 7 days starting from 7 days before the date written on the graph's axis. Fig. 1 and Fig. 2 are the graph of electricity consumption per day and seven days for four weeks at the Lucia's house.

In the first week, the use of electrical energy in the Lucia's house was carried out without any saving efforts or called business as usual. In the second to fourth week, saving attempts were made by turning off unused lights and electronics without compromising the comfort of the house occupants. Fig. 1 shows the trend line of daily electricity consumption at Lucia's house. From Fig. 2, the trend line in the use of electrical energy from the first to the fourth week is decreasing. The total electricity consumption in September 2021 is 115.9 kWh. Compared to electricity usage in the Lucia’s house in August 2021, there is a decrease in the use of electrical energy by 15 kWh or 11.45%.

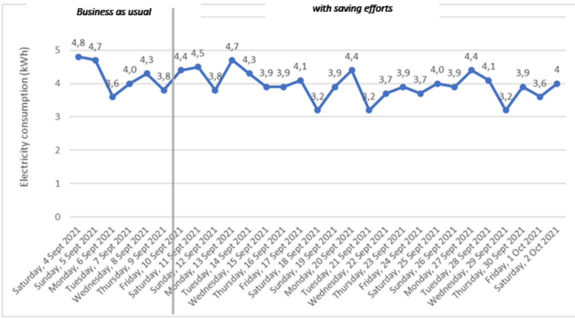

Fig. 1. Electricity Consumption per Day at Lucia's House

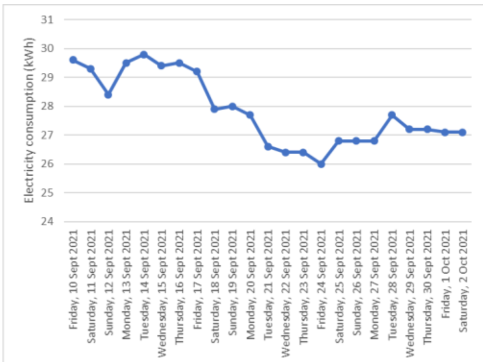

Fig. 2. Electricity Consumption per 7 Day at Lucia's House

Based on the previous research [2][3][4], Arduino and Bluetooth modules can control the lights on and off. ESP8266 Node MCU Wi-Fi module can also control light control via Wi-Fi [5]. In addition, Arduino can measure the use of electricity from an electronic device [6][7].

This research designs a light control system that can be set via a smartphone application to turn on and off automatically. In the smartphone application, there is a program to monitor the use of electrical energy for a light bulb and a program to calculate the duration of the lamp based on the user's electricity budget. Automation features based on monitoring energy and electricity budgets assist in knowing potential savings. In addition, no automation features for LED lamps have been found based on energy monitoring and electricity bill budgeting in the domestic market [8].

II. Materials and Tools

A. *Node MCU ESP8266 AMICA Wifi Module*

ESP8266 consists of a microcontroller (the part that controls) and TCP/IP (the part that connects to the internet). This chip can connect IoT devices to the internet via a Wi-Fi network by forming an embedded web server in the chip so that PCs and smartphones can connect to the ESP8266 chip [9].

In this research, the Node MCU used is the AMICA Node MCU (version 2) because it is smaller than Node MCU version 3. Node MCU is used to connect the circuit system to a wifi network so the user can control it via a wifi network.

B. *ACS712 20A Current Sensor*

The ACS712 Current Sensor is a sensor that can measure AC and DC from Allegro MicroSystems, Inc. The input voltage of this sensor is 5 V and will produce an output voltage in an analog form whose magnitude is proportional to the current being measured [10]. The current sensor ACS712 follows the working principle of the Hall Effect, where the sensor produces an output voltage that varies with the amount of current entering the sensor.

C. *Relay module SRD-05VDC-SL-C*

The relay module is an electrically operated electromagnetic switch. Like switches in general, a relay connects or disconnects the flow of electric current. Relay can be controlled by low voltage rated at 3.3 V and high voltage rated at 12 V, 24 V, and even mains voltage (230 V European standard or 120 V American standard) [11]. This research uses relay SRD-05VDC-SL-C due to this is the easiest to find in the market online during the making of this work.

D. *Others*

The other materials and tools used in this work are Light Emitting Diode (LED) 3W as the object to be controlled, transparent cable, bras plug, and FK 218 lighting as a lamp holder.

E. *Arduino IDE*

Arduino IDE software can run on Windows, Macintosh, OSX, and Linux operating systems. This software appears as an open-source device open to an experienced programmer's development. In this research, Arduino IDE software is essential in setting the Node MCU. Then, The LED can be controlled by it.

F. *MIT App Inventor*

This block-based programming software facilitates the creation of complex applications in much less time than in traditional programming environments. The MIT App Inventor project aims to develop software by empowering everyone, especially young people, to move from technology consumption to technology creation [12]. In this research, MIT App Inventor plays an essential role in designing and application coding on the smartphone for the light controller.

G. *ThingSpeak*

ThingSpeak is an IoT analytics platform that can aggregate, visualize and analyze data flows live from Cloud storage. In this research, ThingSpeak provides a fixed website link so that the user does not need to be periodically replaced the embedded web server link in the form of IP addresses that change frequently. ThingSpeak also visualizes sensor data in graphical and numerical form in real-time.

III. Design Methodology

A. *System Design Overview and Workflow*

Fig. 3 is the flowchart of lighting control design system’s overview and workflow.

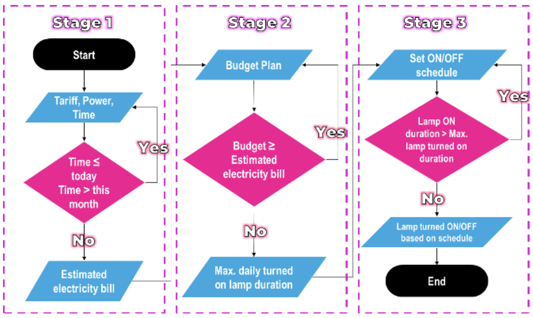

Fig. 3. System Design Flowchart

In stage 1, the user is directed to input the electricity tariff (Rp./kWh), lamp power (W), and automation period. Automation time cannot be less than today and more significant than this month. The user must re-enter the automation time range if the conditions are not met. The output of this stage is the estimated electricity bill when the lights are turned on continuously for 24 hours.

In stage 2, the user is directed to enter a budget plan. The user must re-enter the budget plan if the budget plan exceeds the estimated electricity bill calculated in the first stage. The output of the second stage is the maximum duration of the lights per day.

In stage 3, the user is directed to set a schedule for turning off the lights for 24 hours. This schedule will be applied daily within the automation time range inputted in stage 1. The lights' duration cannot be greater than the maximum duration of the lights, which is calculated in stage 2. If the schedule meets the requirements, the lights will start turning off according to the specified schedule.

B. System Block Diagram

The block diagram, as shown in Fig. 4, shows the main component of the control system design. The LED light is connected to the ACS712 current sensor and relay module SRD-05VDC-SL-C. The ACS712 current sensor sends input to the Node MCU in the form of a voltage which will be translated into LED lamp current. The relay module receives output from the Node MCU in the form of a voltage to regulate the LED lights on and off. The Node MCU sends on-off commands and reading of ACS712 sensor data to web server ThingSpeak via the internet. Commands and data stored in ThingSpeak are then sent to the app smartphone through the internet network.

Fig. 4. System Design Block Diagram

C. Testing Method

- LED On/Off Program via Smartphone Application

1) The Circuit

The ESP8266 Node MCU output voltage, which is 3.3V-5V DC, cannot be used to turn on a 3 Watt LED lamp whose input voltage is 170V-250V AC. Therefore, the first step is to make a simple electrical circuit that can power a 3 Watt LED lamp, consisting of the E27 lamp holder, cable, and an EU plug. Then the relay module must be added in circuit. Fig. 5 is the schematic of circuit with relay module.

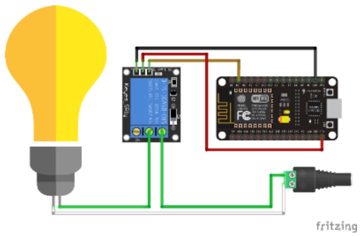

Fig. 5. Circuit Plus Relay Module Schematic

2) Application Display

The circle button labelled "OFF" or "ON" functions to turn the light off when pressed. If the button is dark gray and says “OFF” is pressed, the button will change color to light gray with the words “ON” and the light will turn on about 1-6 seconds later. As long as the “ON” button is light gray, the button cannot be pressed for 20 seconds. There is timer below the screen that says “Please wait for … seconds” to indicate how many seconds have passed since the button was pressed. After 20 seconds, the “ON” button will turn yellow.

When using an IP address as a web server, lights can immediately turn off or on after the ON / OFF button in the application is pressed. The time range of about 1-6 seconds when using ThingSpeak as the web server is because the path of sending commands to the IP address is shorter than to ThingSpeak, an intermediary platform. Using ThingSpeak, once the Node MCU is connected to the IP address, Node MCU commands are passed to the web server ThingSpeak and then forwarded to channel-specific target.

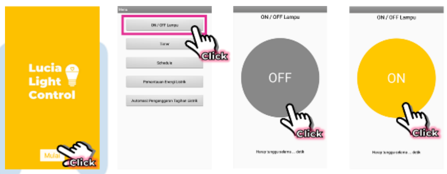

Fig. 6. Light’s ON/OFF Menu Display

- LED On/Off with Timer Program

The LED light can be turned ON/OFF by a timer. Hours, Minutes and Seconds settings by the user will be counted down. If the button Start is pressed, the countdown timer at the bottom of the app will continue to decrease every 1 second. If it reaches 00:00:00, the light will turn off if the circle button is set to “OFF”, or the light will turn on if the circle button is set to “ON”.

Fig. 7. Timer Menu Display

- LED On/Off with Schedule Program

The schedule is the LED on and off scheduling feature. First, press the “Select Time” button. After selecting the time, press the “Lamp Condition” button to set whether you want to schedule the lights to turn on or off at the selected time.

Fig. 8. Schedule Menu Display

After setting the time and condition of the lights, press the "Add" button then the scheduling will be carried out. The caption “Currently there are… scheduling” will be updated according to the many schedules that have been added after pressing the "Add" button or deleted after the schedule you want to delete is selected, and then the "Delete" button is pressed.

- Electrical Energy Monitoring Program

1) *The Circuit*

After the relay module turns off the lights, an ACS712 current sensor is added to the circuit. ACS712 current sensor is installed in series after the relay module. Fig. 9 is the schematic of the circuit after is added by the ACS712 current sensor.

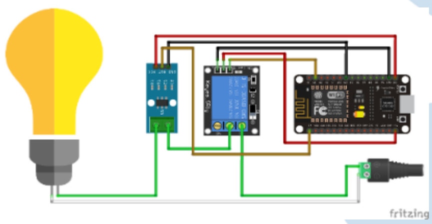

Fig. 9. Circuit Plus ACS712 Current Sensor Schematic

2) *Application Display*

This menu has a numeric display of energy used, a numeric display of lamp power, an energy graph, and a power graph that updates every 20 seconds. The ACS712 current sensor data is processed in the Arduino IDE and then sent to the ThinkSpeak web server. Next, ThingSpeak will visualize the ACS712 sensor data in graphic form and show the ACS712 sensor data in a numeric display. The display of numbers and graphs from ThingSpeak is then entered into the MIT App Inventor to view the data visualisation through the smartphone application.

Fig. 10 displays the Electrical Energy Monitoring menu when the lamp is on. metersis the total electrical energy that the lamp has used. Power Meter shows the power of the lamp when the lamp is on, which is 3.01 W. The Energy Graph line shows the energy value increase when the lamp is turned on. The line on the Power Graph indicates the power rating is around 3 Watts when the lamp is on.

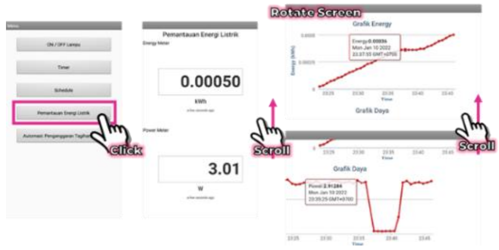

Fig. 10. Electrical Energy Monitoring Menu Display When LED ON

Then, Fig. 11 displays the Electrical Energy Monitoring menu when the lights are off. Energy Meter is the total electrical energy that the lamp has used. The power meter shows how much power the lamp has when the lamp is off, which is 0.28 W. The line on the Energy Graph tends to be flat when the lamp is turned off. The line on the Power Graph shows the power rating is around 0 Watt when the lamp is off.

Fig. 11. Electrical Energy Monitoring Menu Display When LED OFF

- Automation Feature Based on Electricity Bill Budgeting Program

1) *Stage 1: Input PLN Electricity Basic Tariff, Lamp Power, and Automation Time Range*

PLN's basic electricity tariff is inputted according to the ongoing PLN stipulation. The input lamp power figures can be referenced from the specifications on the lamp box or lamp power calculations in the application. Minimum automation time starts at 00:00. If 00:00 has passed, the automation can be started for the next day. The maximum automation time is the end of the same month as the automation start date. If "Select Start Date" and "Select End Date" do not match the conditions described previously, a warning message will appear, and the "Submit" button cannot be pressed.

ISSN 2355-3286

If the start date and end date are by the program provisions, then the "Submit" button can be pressed. When the "Submit" button is pressed, a "Check Input" will appear to ensure the user's input TDL, power, and automation time is correct. In addition, a "Calculation" will appear where users can see the duration of the day, the automation time, and the estimated electricity bill.

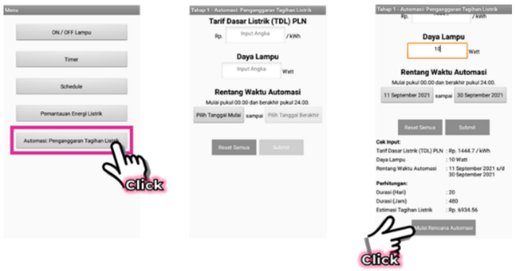

Fig. 12. Automation Menu Display - Stage 1

2) Stage 2: Input the Budget Plan

At the top of the application screen there is a summary of user input in previous stage. Next the user can enter the budget plan figures. If the user's budget plan input is greater than the estimated value of the electricity bill, a warning message will appear and the "Light Scheduling per Day" button will not appear. If the user's budget plan input is less than the estimated value of the electricity bill, a "Calculation" and "Light Scheduling per Day" button will appear to proceed to the next stage. In the “Calculation” section, users can see how long the lights are on per day according to the user's budget.

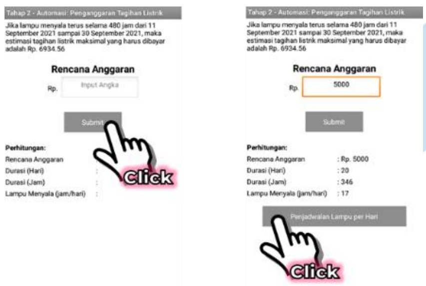

Fig. 13. Automation Menu Display - Stage 2

3) Stage 3: Light Scheduling per Day

At stage 3, you can see the buttons are rectangular in shape with the words "OFF". These buttons are worth 1 hour calculated from the hour on the top left of the buttons. If these buttons are pressed by the user, then the button pressed will read “ON” and the “Total Light On Time” will increase. If the buttons that say "ON" are pressed again, the writing on the button will change to "OFF" and "Total Light Time On" will decrease.

Fig. 14. Automation Menu Display - Stage 3

At the bottom of the ON/OFF buttons, there is a “Check” button. If “Total Lamp On Time” is less than “Max. Duration of Lights On/ Day”, there will be a message informing you of the difference in costs from the input of the previous budget plan. If “Total Lamp On Time” is greater than “Max. Duration of Lights On/Day”, there will be a warning message. The “Create Schedule” button can be pressed if “Total Light On Time” is less than or equal to “Max. Lamp Light Duration/Day”. After pressing the “Create Schedule” button, a schedule of lights will appear at the bottom of the screen. If we scrolldown, we will find the "Start Automation" button. After pressing the “Start Automation” button, the schedule that has been made in stage 3 of the Automation menu will be sent to the Schedule menu. It appears that there are now 24 scheduling lights that will be implemented. Thus, the lights will turn off according to the schedule.

IV. Result and Analysis

A. Measurements and Calculations

- NodeMCU Output Voltage: VIN, 3.3V, GPIO5 Pins

To select which Node MCU pins are suitable to use by the 05VDC-SL-C relay module and ACS712 current sensor, the output voltage of the Node MCU pins is measured. Table I shows the measurement results for the VIN, 3.3V, and GPIO5 Node MCU pins using a multimeter.

TABLE I. Node MCU Output Voltage Measurement

Pin Node MCU	Measurement to-										
	1	2	3	4	5	6	7	8	9	10	Mean
VIN (V DC)	4,75	4,75	4,76	4,76	4,76	4,76	4,76	4,76	4,76	4,76	4,76
3.3V (V DC)	3,28	3,30	3,30	3,30	3,30	3,30	3,30	3,30	3,30	3,30	3,30
GPIO (V DC)	3,28	3,28	3,28	3,28	3,28	3,27	3,27	3,27	3,27	3,27	3,28

Based on measurements, it is using a multimeter, the average output voltage of the VIN pin = 4.76V, the 3.3V pin = 3.3V, and the GPIO5 pin = 3.28V. The input voltage of the relay module 05VDC-SL-C and the current sensor ACS712 is 5VDC, while based on measurements, no Node MCU pin produces an output voltage of 5VDC. The solution is better to use a 5VDC relay module equipped with an optocoupler so that it can still be controlled by the Node MCU with a voltage of less than 5VDC. The selection of the Node MCU pins for the ACS712 current sensor will be calculated through the following calculations.

- Zero Current Output Voltage ACS712: VIN, 3.3V Pins

Zero current output voltage is generated by the ACS712 current sensor when no electricity is flowing to turn on the lamp.

TABLE II. ZERO CURRENT OUTPUT VOLTAGE MEASUREMENT

VCC	Tools	Unit	Measurement to-										
			1	2	3	4	5	6	7	8	9	10	Mean
VIN	Multimeter	VDC	2,37	2,38	2,37	2,38	2,38	2,35	2,35	2,36	2,36	2,36	2,37
	Arduino	Analog	801	802	801	800	800	797	797	797	798	798	799
	IDE	VDC	3,72	3,73	3,72	3,72	3,72	2,58	2,58	2,57	2,57	2,58	3,15
3.3V	Multimeter	VDC	1,63	1,63	1,63	1,63	1,63	1,62	1,62	1,62	1,62	1,62	1,63
	Arduino	Analog	549	547	547	548	549	545	545	544	544	545	546
	IDE	VDC	1,77	1,76	1,76	1,77	1,77	1,76	1,76	1,76	1,76	1,76	1,76

Next is to compare the zero current output voltage measurement results from Arduino IDE with zero current output voltage based on the formula. According to the theory, if no current flows through IP+ and IP- or the current is equal to zero, then the output voltage of the ACS712 sensor will show VIOUT = VCC × 0.5 [13]. The error in this calculation is how far the zero current output voltage value is based on Arduino IDE from the zero current output voltage value based on the VIOUT formula. The following is the calculation of the theoretical VIOUT and the error of measured VIOUT with ACS712 against theoretical VIOUT:

1) $$VIOUT_{VIN\ theoretical} = VCC_{VIN\ [Table\ 4]} \times 0{,}5$$
$$= 4{,}76 \times 0{,}5$$
$$= 2{,}38V$$

$$VIOUT_{VIN\ measured\ [Table\ 5\ with\ Arduino\ IDE]} = 3{,}15V$$

$$Error = \frac{|VIOUT_{meas.} - VIOUT_{theo.}|}{VIOUT_{theo.}} = \frac{|3{,}15 - 2{,}38|}{2{,}38} = 0{,}32$$

2) $$VIOUT_{3.3V\ theoretical} = VCC_{3.3V\ [Table\ 4]} \times 0{,}5$$
$$= 3{,}3 \times 0{,}5$$
$$= 1{,}65V$$

$$VIOUT_{VIN\ measured\ [Tabel\ 5\ with\ Arduino\ IDE]} = 1{,}76V$$

$$Error = \frac{|VIOUT_{meas.} - VIOUT_{theo.}|}{VIOUT_{theo.}} = \frac{|1{,}76 - 1{,}65|}{1{,}65} = 0{,}06$$

According to the specification sheet, the ACS712 current sensor's input voltage is 5V. The VIN=4.76V pin voltage is closer to 5V than the 3.3V=3.3V pin. However, the error of VIOUT in the 3.3V pin is smaller than the VIOUT in the VIN pin. By looking at this calculation, the 3.3V pin is decided to be the voltage source for the ACS712 current sensor.

B. System Design Overview and Workflow

- Electrical Energy Monitoring Menu

The ACS712 current sensor measures the electric circuit's current flow by calculating the sensor's potential difference or output voltage. The output voltage of the current sensor mounted on pin A0 Node MCU is first displayed in analog form bit 0-1024 by Arduino IDE. The mathematical equation for converting analog data into sensor voltage (V sensor) is:

$$V\ sensor = Analog\ Data\ (bit) \times \frac{3{,}3\ V}{1024\ (bit)} \tag{5}$$

Analog data is obtained from analogRead(A0), 3.3 V from the maximum ADC of the Node MCU, and 1024 is the maximum analog data that the Node MCU can measure. Next, the V sensor is translated into sensor current (I sensor) using the following mathematical equation:

$$I\ sensor = \frac{V\ sensor\ (V)}{0{,}1\ V/A} \tag{6}$$

V sensor is obtained from equation (5), and 0.1V/A is obtained from the current sensor sensitivity of ACS712 20A. If using the ACS712 5A, then the sensitivity is 0.185 V/A. If you use the ACS712 30A, the sensitivity is 0.66 V/A.

After getting the circuit current, the next step is to find the lamp voltage. Inside the 3W LED, there is an MB10F rectifier. The MB10F rectifier specification shows that the voltage drop for every 0.4A is 1V [14]. So, the mathematical equation for the lamp voltage is as follows:

$$V\ Lamp = I\ Sensor\ (A) \times \frac{1V}{0{,}4A} \tag{7}$$

The lamp power calculation is the circuit current multiplied by the lamp voltage. Here is the mathematical equation:

$$Power\ (W) = I\ Sensor\ (A) \times V\ Lamp\ (V) \tag{8}$$

Finally, the calculation of energy is power times time. This series updates energy calculation every 20 seconds. Here is the mathematical equation:

$$Energy(kWh) = Power(W) \times \frac{1kW}{1000W} \times 20s \times \frac{1h}{3600s} \tag{9}$$

- Automation Menu:Electricity Bill Budgeting - Stage 1

The first input is TDL (Rp./kWh), lamp power (W), and automation period (days). If the next buttonSubmitpressed, it will display “Check Input” to ensure the user input is correct and “Calculation” will also appear. At this stage, the application calculates the duration (days), duration (hours), and estimated electricity bills. Here is the mathematical equation of the calculations in the first stage:

$$Duration\ (Day) = End\ Date - Start\ Date \tag{10}$$

$$Duration\ (Hour) = Duration\ (Day) \times \frac{24\ Hour}{1\ Day} \tag{11}$$

$$Estimated\ Electricity\ Bill = \frac{TDL\ (Rp./kWh) \times Duration(h) \times Power(W) \times 1\ kW}{1000W} \tag{12}$$

- Automation Menu:Electricity Bill Budgeting - Stage 2

The budget plan is entered into the column provided, then click Submit. Next is the calculation of duration (days), duration (hours), and the length of time the lights are on (hours/day).

$$Duration\ (Hour) = \frac{Budget\ Plan(Rp.)}{TDL\ (Rp./kWh)} \times \frac{Power(W) \times 1kW}{1000\ W} \tag{13}$$

$$LED\ ON\ \left(\frac{Hour}{Day}\right) = \frac{Duration\ (Hour)}{Duration\ (Day)} \tag{14}$$

- Automation Menu:Electricity Bill Budgeting - Stage 3

At this stage, the user enters a schedule for turning off the lights as many times as the lights are on per day calculated in the previous stage. The total duration of the lights on must be less than or equal to the length of time the lights are on per day. Here is the mathematical equation for this condition:

$$LED\ ON\ Duration\ Total\ \leq Max. LED\ ON\ per\ Day\ Duration \tag{15}$$

- Energy Monitoring with ThingSpeak

Current measurements from current sensors and lamp voltage calculations, lamp power, and lamp electrical energy come from the Arduino IDE and are then sent by the Node MCU to ThingSpeak. Fig. 15 is a graph of the current, voltage, power, and electrical energy of the lamp when it is turned on.

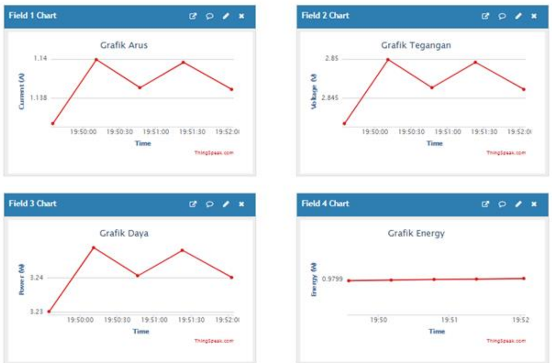

Fig. 15. Current, Voltage, Power, and Energy When LED ON Graphs

TABLE III. CURRENT, VOLTAGE, POWER, ENERGY WHEN LAMP IS ON

Lamp is ON	Measurement to-										
	1	2	3	4	5	6	7	8	9	10	Mean
Current (A)	1,14	1,14	1,14	1,14	1,14	1,14	1,14	1,05	0,98	1,02	1,10
Voltage (V)	2,85	2,85	2,85	2,84	2,84	2,84	2,85	2,62	2,46	2,34	2,73
Power (W)	3,25	3,24	3,25	3,24	3,22	3,23	3,24	2,74	2,41	2,57	3,04
Energy (Wh)	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02

When the light is on, the average power calculated by the program is 3.04W. The lamp power based on the specifications on the box is 3W. If the calculated power of the program is compared with the lamp power of the specification, the error value of the calculated power is:

$$Power\ Calculation\ Error = \frac{|3{,}04 - 3{,}00|}{3{,}00} \times 100\% = 1{,}3\% \tag{16}$$

This value shows the difference in the measurement of the lamp power value using the ACS712 current sensor with the lamp power value in the lamp specification. Fig. 16 is a graph of the current, voltage, power, and electrical energy of the lamp when it is turned off.

When the light is off, the average power calculated by the program is about 0.01W. The current, voltage and power graph drops near 0 when the lamp is turned off. However, the energy graph does not go down because the energy graph is the total energy since the first time the lamp is turned on.

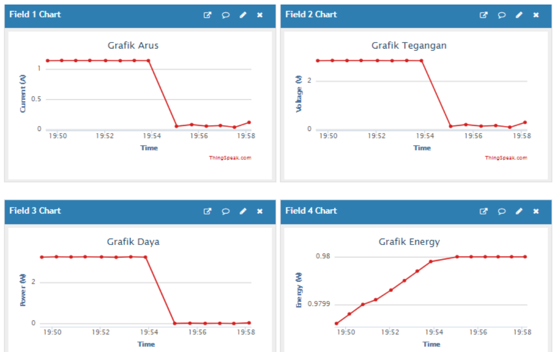

Fig. 16. Current, Voltage, Power, and Energy When LED OFF Graphs

TABLE IV. CURRENT, VOLTAGE, POWER, ENERGY WHEN LAMP IS OFF

Lamp is OFF	Measurement to-										
	1	2	3	4	5	6	7	8	9	10	Mean
Current (mA)	53,71	81,52	57,00	66,19	39,38	39,38	19,06	49,67	87,83	90,50	58,42
Voltage (mV)	134,28	203,80	142,50	165,47	98,45	98,48	47,65	124,18	219,57	226,26	146,06
Power (mW)	7,21	16,61	8,12	10,95	3,88	3,88	0,91	6,17	19,29	20,48	9,75
Energy (kWh)	0	0	0	0	0	0	0	0	0	0	0

V. CONCLUSION

The lamp turned on and off via Node MCU through a smartphone application designed from web-based software programming, MIT App Inventor. The Node MCU is connected to the smartphone application through the ThinkSpeak web server. Period from ON/OFF button pressed on the smartphone until the light is ON/OFF for about 1-6 sec. Lamp's power calculation error using the current sensor against the power of lamp specification is 1.3%.

ACKNOWLEDGEMENT

We thank Universitas Multimedia Nusantara, which provides the whole necessary for this research.

REFERENCES

[1] PT PLN (Persero), "Electricity Tariff - PT PLN (Persero)," 2021.[Online].Available:https://web.pln.co.id/pelanggan/tari

f-energi-listrik/tariff-adjustments. [Accessed September 20, 2021].

[2] F. Rahmadayanti, "Arduino-Based LED Lamp Android Application," 2016.

[3] M. Hudori and Y. Paisal, "Design of Automatic Lighting Control System in Residential Houses to Increase Electricity Consumption Efficiency," 2019.

[4] E.T. Setiawan, "Controlling Home Lights Based on Arduino Microcontroller Using Android Smartphone".

[5] J. Chandramohan, R. Nagarajan, K. Satheeshkumar, N. Ajithkumar, PA Gopinath and S. Ranjithkumar, "Intelligent Smart Home Automation and Security System Using Arduino and Wi-fi," 2017.

[6] M.I.L. Wahyu, Syaifurrahman and M. Saleh, "Design of a Digital kWh Meter as a Cost Calculator of Electrical Energy Usage Based on Arduino UNO R3".

[7] G. Herandy and M. Suprianto, "Monitoring of Costs and Measurement of Electric Power Consumption Based on Arduino Mega2560 Using the Web," 2019.

[8] L.B. Rahawarin, "Comparative Study of Data and Innovation Ideas for Automation Features of Smart Home Products PT. BARDI Automation Solutions," UMN Knowledge Center, Tangerang, 2021.

[9] E. Ouyang, "Introduction," in *Hands-On ESP8266: Mastering Basic Peripherals*, Hands On Embedded, 2018, pp. 1-4.

[10] A. Roman, “A Simple Approach to Measure ACDC,” March 25, 2020. [Online]. Available: https://www.romn.io/2020/05/a-simple-approach-tomeasure-acdc.html. [Accessed September 21, 2021].

[11] R. Santos, "ESP8266 NodeMCU Relay Module – Control AC Appliances (Web Server)," Random Nerd Tutorials, December 18, 2019. [Online]. Available: https://randomnerdtutorials.com/esp8266-relay-module-ac-webserver/. [Accessed September 22, 2021].

[12] MIT App Inventor, "About Us - MIT App Inventor," Massachusetts Institute of Technology, [Online]. Available: https://appinventor.mit.edu/about-us. [Accessed September 22, 2021].

[13] Allegro MicroSystem, "Fully Integrated, Hall-Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor," [Online]. Available: https://pdf1.alldatasheet.es/datasheetpdf/view/205336/ALLEGRO/ACS712.html. [Accessed September 21, 2021].

[14] PACELEADER [PACELEADER INDUSTRIAL], "Miniature Glass Passivated Single-Phase Surface Mount Flat Bridge Rectifier," [Online]. Available: https://pdf1.alldatasheet.com/datasheet pdf/view/310355/PACELEADER/MB10F.html. [Accessed September 24, 2021].