Comparison of Bluetooth Technology Implementation using BLE and Bluetooth Classic Protocols in Pelican Crossing

Januarius Dharma Anugerah Putra[1], Muhammad Syirajuddin Suja'i[2*], Ahmad Wilda Yulianto[3]

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

[2] Telecommunication Engineering Study Program, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia

[1]januariusdharma@gmail.com , [2]syirajuddin@polinema.ac.id , [3]ahmadwildan@polinema.ac.id

***Abstract*—To address the 34.60% surge in Indonesian traffic accidents recorded by Korlantas Polri, which resulted in over 19,000 fatalities, this research developed a smart pelican crossing prototype comparing Bluetooth Classic and Bluetooth Low Energy (BLE) protocols to enhance pedestrian safety. Utilizing the Science and Technology Development Research method, the study evaluated both protocols across distances ranging from 30 cm to 450 cm under both unobstructed and obstructed conditions. Experimental results demonstrated that BLE offers a superior detection range, achieving average success rates of 92.67% in unobstructed environments and 78.67% with obstacles. Conversely, Bluetooth Classic exhibited higher average received power values of -83.22 dBm and -89.56 dBm, respectively. Crucially, Bluetooth Classic demonstrated significantly greater operational efficiency for users, requiring only 6 seconds and 4 touches for application access and installation, whereas BLE required 86 seconds and 8 touches. Although BLE provides a broader detection range, this study concludes that Bluetooth Classic is more suitable for public implementation by government authorities due to its rapid connection speed and superior user accessibility. This technological intervention serves as a vital infrastructure solution for protecting vulnerable road users, ensuring greater safety for pedestrians with specific restrictions, and reducing fatalities at high-risk highway crossings.**

***Keywords*—Bluetooth, Bluetooth Classic, BLE, Pelican Crossing, Protocol Comparison.**

I. INTRODUCTION

Traffic accidents on highways are increasing over time. Despite being equipped with various markings and signs, the accident rate remains high. In 2016, the World Health Organization (WHO) reported that the death toll from road accidents reached 1.36 million [1]. In Indonesia, in 2022, specifically from January to September, the Traffic Corps of he National Police recorded 94,617 accident cases, which represents an increase of 34.60% from the previous year. This was accompanied by 19,054 fatalities, an increase of approximately 3.72% from the previous year [2].

One of the causes of this is due to drivers who neglect the conditions and presence of pedestrians, especially those crossing the road on highways. Therefore, a technologically advanced traffic system is needed to help improve safety for pedestrians.

Pelican crossing (pedestrian light-controlled crossing) is one of the solutions found to reduce accident rates. Pelican crossing, or in Indonesian, "penyeberangan pelikan" is a pedestrian crossing facility equipped with traffic lights to cross the road safely and comfortably [3]. Although pelican crossings have standards in their operation, most pelican crossings still use physical buttons for activation [4]. During the pandemic era, the public is advised to minimize physical contact with others and public objects, which is contrary to the current pelican crossing system technology. On the other hand, public participation is a key factor in preventing the spread of the Covid-19 pandemic, which is still not completely over. The Ministry of Health also recorded 14 confirmed cases of the Omicron subvariant CH.1.1 or Orthrus variant in Indonesia [5]. The government, as the party advising the public to implement social distancing and physical distancing, seems to be lacking in applying the latest technology as a solution to this problem. The current pelican crossing system also has limitations for users with height constraints to press the button. This is felt by children or people with disabilities. The static placement of physical buttons makes it difficult for certain users to use them, and they end up crossing directly or at other locations with greater risk. Cases of theft and damage, such as the theft of 10 pelican crossing boxes in England and their resale, and similar issues in Indonesia, such as in Banjarmasin on Jalan Jenderal Sudirman and Jalan Pangeran Samudera, where the crossing were damaged or even missing, and similar incidents in Jalan Ahmad Yani, Karawang Barat, West Java, where the crossings were also damaged [6] [7] [8]. These issues certainly need attention and resolution.

*Corresponding author

E-ISSN: 2654-6531 P- ISSN: 2407-0807

With technological advancements, various models of pelican crossings are becoming increasingly diverse. Some examples include the application of YOLO in crossings by Chyi-Ren Dow, the use of FaceNet and AgeNet by Purnawarman Musa, and the use of sensors instead of buttons in crossings by Rahmad Ahmad [9] [10] [11]. However, despite these various developments, there is still room for improvement, such as with Bluetooth technology. Bluetooth, which has existed since 1994, the emergence of the BLE protocol, and its continued development up to version 5, and its ongoing usage, indicates that Bluetooth still has potential for application [12] [13] [14]. Not only from Bluetooth, but the ESP32 also has the potential for further development, especially with its capabilities that still support Bluetooth technology [15] [16].

Therefore, the application of Bluetooth technology as an alternative solution for the pelican crossing system is very interesting in addressing this issue. This study will analyze the performance differences between the pelican crossing system based on Bluetooth technology with the BLE protocol and the same system but using the classic Bluetooth protocol. This analysis will compare the efficiency, range, performance, and durability of both protocols to provide the best solution for implementing an efficient and safe pelican crossing system.

II. METHOD

A. Research Stages

The design of the system that will be created will first go through several paths. This is intended so that the research is carried out in detail so that the results obtained can achieve maximum results. The research stages that will be carried out in creating this system are shown in Figure 1.

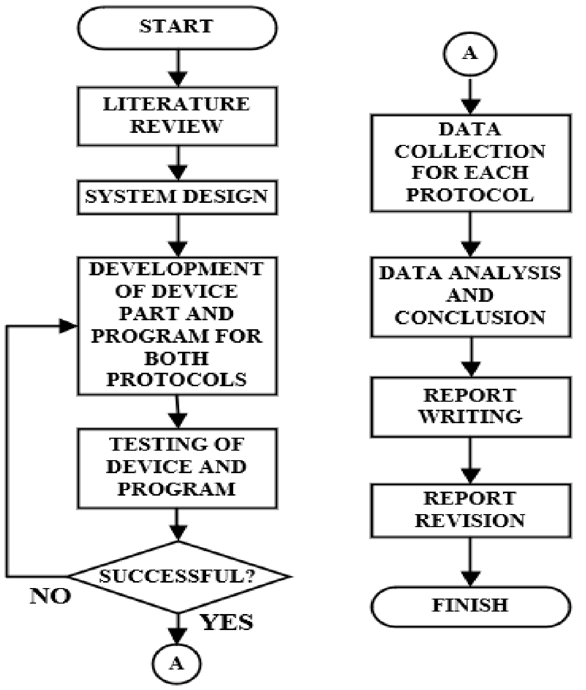

Figure 1. Research Stages

The following is a description of Figure 1:
The first stage carried out by the author was a literature study regarding wireless sensor networks, Bluetooth technology including its protocols, ESP32 WROOM, and Arduino IDE. The second stage is system design where the author will determine how smartphones activate traffic lights using Bluetooth. The third stage is making tool parts in the form of prototypes using ESP32 WROOM. Making the tool will come first which will then be programmed using the Arduino IDE application on the laptop. The process of making each part of each protocol will be carried out in stages to avoid fatal errors and make repairs easier. The fourth stage is testing the tools and programs for both protocols. Here what is tested is the activation of the tool and the success of the system. Testing is carried out periodically for each change made to find out any errors that may occur. If an error occurs, the author will change the tool design or programming again to correct the error until it is successful. The fifth stage is data collection. Systems on each protocol that are ready to use will be tested. What is tested is the success of the system and its smoothness which is more focused on the access time of the smartphone and the response of the system. The sixth stage is data analysis. The data that has been collected will be analyzed according to previously determined parameters. The data analysis process will be adjusted to the basic theory used. Then conclusions will be made based on the results of the analysis. The seventh stage is making a report. If the research has been successful and a conclusion has been reached, it will continue with making a report which will later be submitted as evidence of the research. This report will be created periodically so that in the end there are not too many things that need to be done while avoiding data loss in the previous process. The eighth stage is report revision. Making a report definitely has errors that need to be corrected, such as wording and other things, so it requires the help of other parties such as supervisors in correcting it to get the actual report results.

B. Block Diagram

The system design to be created is divided into four parts, namely system block diagram, system flowchart, hardware design and software planning.

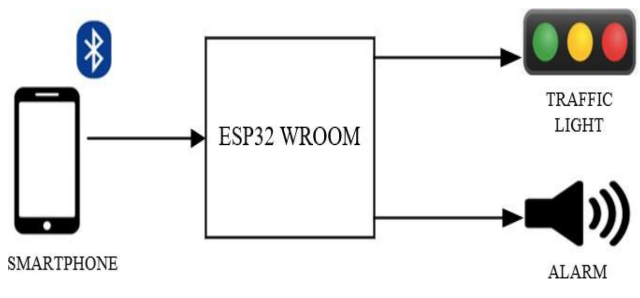

Figure 2. System Block Diagram

Figure 2 is a brief overview of input, process and output. The input of this system is the user's smartphone which uses Bluetooth to be detected by the ESP32. The processing stage will be carried out on the pelican crossing which is supported by the ESP32 WROOM module. Finally, the output from the system will be a colored light that depends on road conditions and an alarm sound as a warning to road users.

C. System Flowchart

Figure 3 shows the flowchart of the processing system that occurs on the ESP32. The Bluetooth owned by the ESP32 which is located at the pelican crossing will be the input for this system. The following is a system flowchart of the design that will be created.

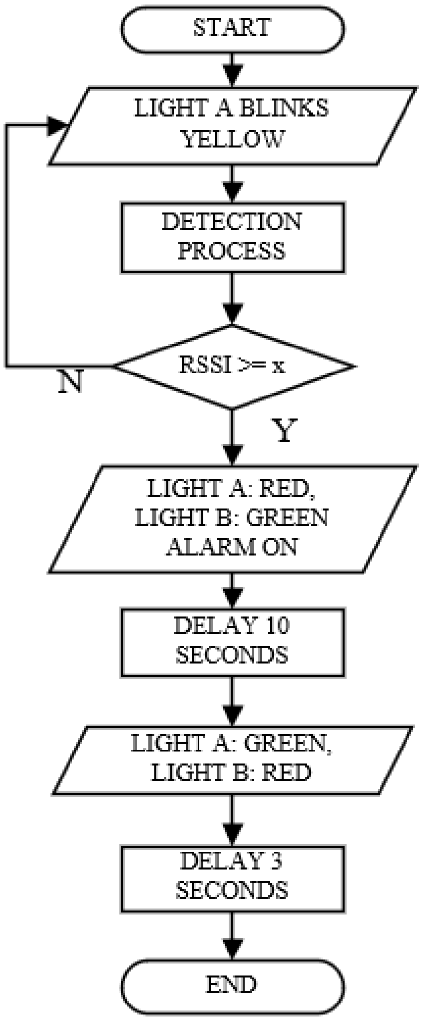

Figure 3. System Flowchart

The initial condition is that light A (driver's light) is flashing yellow. This coincides with the ESP32 scanning nearby Bluetooth devices. ESP32 will calculate the RSSI level as a condition for system activation. If there is a Bluetooth device with an RSSI level greater than -90 dBm, then light A will light red, light B will light green and the alarm will turn on. The RSSI value limit is determined by a distance of 2 meters from the location of the pelican crossing detection. The duration of the crossing is obtained from calculations using formula 2.1 with a crossing width of 2.5 meters, a crossing length of 30 meters, and a crossing volume of 1 person per cycle so that the result is 26 seconds. After that, lamp A will turn green and lamp B will turn red. At this time, the rider can continue the journey. Then the process is complete and it will return to the beginning.

D. System Design

The following is an illustration of the system design on the prototype.

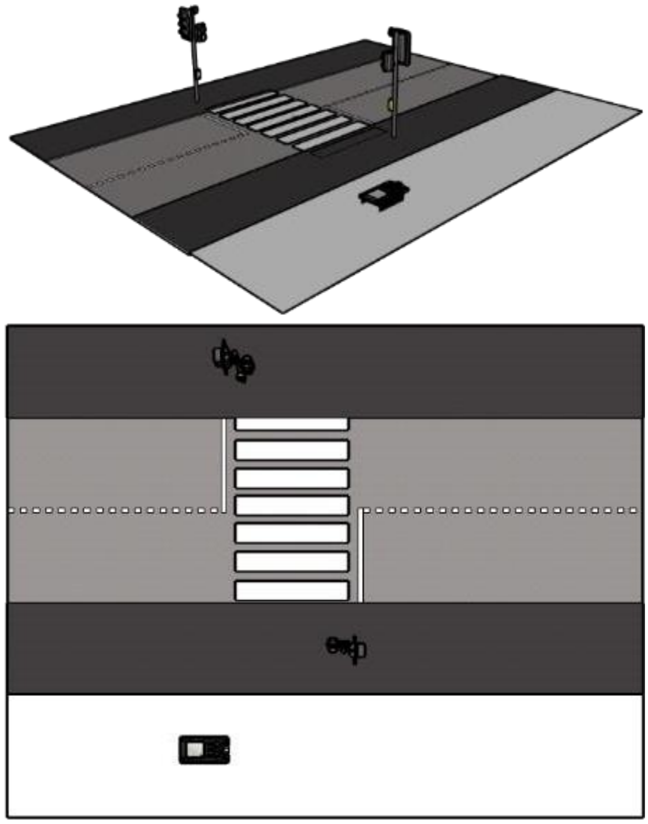

Figure 4. System Design

Figure 4 is an illustration of the prototype hardware that will be made. The system will be used on roads with a two-way concept. The lights and alarms for drivers are placed on the left side of each side of the road facing the direction the driver is coming so that they can be seen and heard by passing motorists. The lights for pedestrians are placed on both sides of the road, right next to the drivers' lights, facing the direction of the crossing, which is intended for pedestrians on each side. The ESP32 will be placed on the side to make it easier to replace the system and connect to a power source. The lights for motorists consist of red, yellow and green lights like traffic lights in general, while the traffic lights for pedestrians only have red and green lights.

Figure 5 and Figure 6 show the schematic and wiring diagram of the Pelican crossing prototype connected to the ESP32 microcontroller component leg. This circuit uses 5 mm LED lights. If you want to use a lamp with greater power, you can connect a relay so that it can be connected to an additional external power source and then connect it to the ESP32. A list of pins and components used can be seen in Table I.

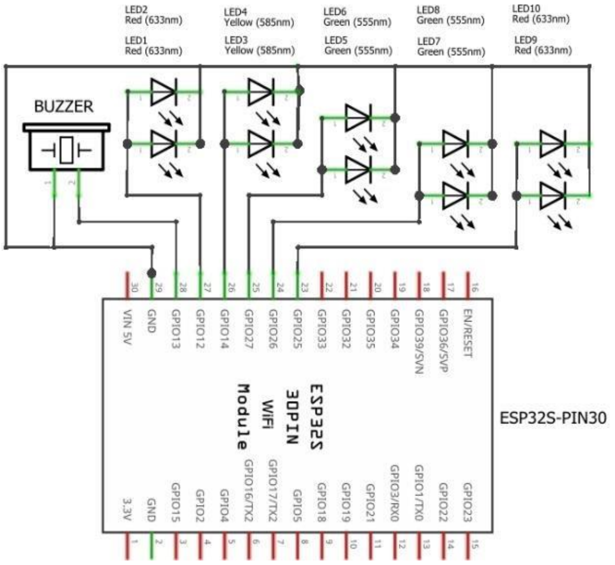

Figure 5 Schematic Diagram of the Entire System

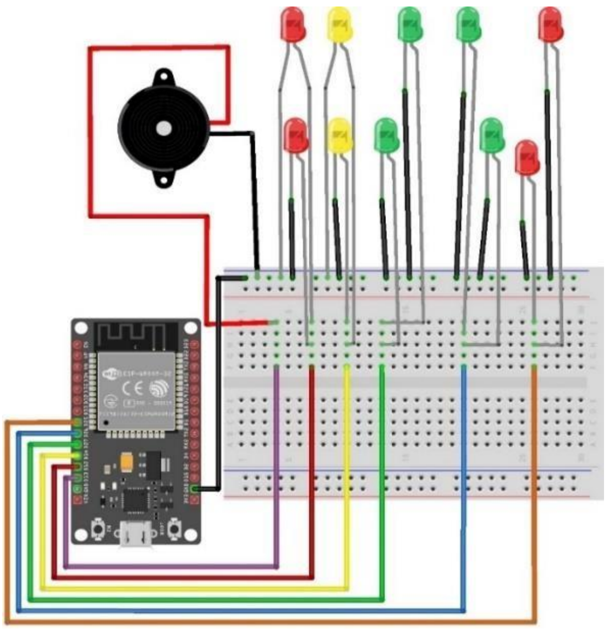

Figure 6. Overall System Wiring

TABLE I
ESP32 PIN USAGE IN THE SYSTEM

Component		ESP32 Pins
Name	Pin	
Buzzer	GND VCC	GND GPIO13
Red LED	GND VCC	GND GPIO12
Yellow LED	GND VCC	GND GPIO14
Green LED	GND VCC	GND GPIO27
Green LED	GND VCC	GND GPIO26
Red LED	GND VCC	GND GPIO25

E. Software Planning

1) Classic Bluetooth Software Planning:

The software planning of classic Bluetooth consists of several parts.

```
#include <BluetoothSerial.h>

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
#endif

BluetoothSerial SerialBT;
```

Figure 7. Use of the Classic Bluetooth Library

Figure 7 shows the use of the BluetoothSerial.h library in the Arduino IDE for classic Bluetooth control on the ESP32 Wroom.

```
if (btScanSync) {
  BTScanResults *pResults = SerialBT.discover(BT_DISCOVER_TIME);
  if (pResults) {
    pResults->dump(&Serial);
    int deviceCount = pResults->getCount();
    int highRSSICount = 0;
    for (int i = 0; i < deviceCount; i++) {
      int rssi = pResults->getDevice(i)->getRSSI();
      if (rssi > rssithreshold) {
        highRSSICount++;
      }
    }
```

Figure 8. Device Detection Process with Classic Bluetooth

Figure 8 shows how to use BluetoothSerial.h to control classic Bluetooth to detect nearby devices. In this process, it also retrieves RSSI value data from the detected device.

```
if (highRSSICount > 0) {
  for (int i = 0; i < 52; i++) {
    digitalWrite(redmain, HIGH);
    digitalWrite(greensec, HIGH);
    digitalWrite(pinalarm, i % 2 == 0 ? HIGH : LOW);
    delay(500);
  }
  digitalWrite(redmain, LOW);
  digitalWrite(greensec, LOW);
  digitalWrite(greenmain, HIGH);
  digitalWrite(redsec, HIGH);
  delay(3000);
  digitalWrite(greenmain, LOW);
  digitalWrite(redsec, LOW);
```

Figure 9. Controlling the Output of Fulfilled Conditions

Figure 9 shows the output results that occur after the system detects a device that has exceeded the minimum RSSI limit. The programming shows that if there is a device that meets the conditions, then for 26 seconds the red light for drivers will come on, the green light for pedestrians will come on and the alarm will sound on and off alternately every second. After that, the driver's red light will turn off, the pedestrian's green light will turn off at the same time as the driver's green light will turn on and the pedestrian's red light will turn on. This will last three seconds. Then the driver's green light and the pedestrian's red light will turn off.

```
} else {
  for (int i = 0; i < 2; i++) {
    digitalWrite(yellowmain, i % 2 == 0 ? HIGH : LOW);
    delay(1000);
```

Figure 10. Standby State

Figure 10 shows the output results when the conditions are not met. The system will show the driver's yellow light flashing every second until the conditions are met.

2) BLE Software Planning:

BLE software planning consists of several parts.

```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>
```

Figure 11. Use of BLE Libraries

Figure 11 shows the use of the BLEAdvertisedDevice.h library in the Arduino IDE to activate the BLE advertising feature on the ESP32 Wroom.

```
class MyAdvertisedDeviceCallbacks: public BLEAdvertisedDeviceCallbacks {
  void onResult(BLEAdvertisedDevice advertisedDevice) {
    Serial.printf("Advertised Device: %s \n", advertisedDevice.toString().c_str());
    lastrssi = advertisedDevice.getRSSI();
  }

BLEScanResults foundDevices = pBLEScan->start(scanTime, false);
int deviceCount = foundDevices.getCount();
```

Figure 12. Device Detection Process with BLE

Figure 12 shows how to use BLEAdvertisedDevice.h to control BLE to detect nearby devices. In this process, it also retrieves RSSI value data from the detected device.

```
if (lastrssi > rssithreshold) {
  for (int i = 0; i < 52; i++) {
    digitalWrite(redmain, HIGH);
    digitalWrite(greensec,HIGH);
    digitalWrite(pinalarm, i % 2 == 0 ? HIGH : LOW);
    delay(500);
  }
  digitalWrite(redmain, LOW);
  digitalWrite(greensec,LOW);
  digitalWrite(greenmain,HIGH);
  digitalWrite(redsec,HIGH);
  delay(3000);
  digitalWrite(greenmain,LOW);
  digitalWrite(redsec,LOW);
```

Figure 13. Controlling the Output of Fulfilled Conditions

Figure 13 shows the output results that occur after the system detects a device that has exceeded the minimum RSSI limit. The programming shows that if there is a device that meets the conditions, then for 26 seconds the red light for drivers will come on, the green light for pedestrians will come on and the alarm will sound on and off alternately every second. After that, the driver's red light will turn off, the pedestrian's green light will turn off at the same time as the driver's green light will turn on and the pedestrian's red light will turn on. This will take 3 seconds. Then the driver's green light and the pedestrian's red light will turn off.

```
} else {
  digitalWrite(yellowmain, HIGH);
  delay(1000);
  digitalWrite(yellowmain, LOW);
  delay(1000);
```

Figure 14. Standby State

Figure 14 shows the output results when the conditions are not met. The system will show the driver's yellow light flashing every second until the conditions are met.

III. RESULTS AND DISCUSSION

A. Analysis of Unobstructed Reach Test Results

Based on data in terms of unobstructed range, it can be concluded that the effective range of classic Bluetooth can only reach 360 cm. This can be seen at a distance of 300 cm. Classic Bluetooth begins to experience a decrease in its ability to receive full power and only reaches 80%. The instability continues until at a range of 390 cm classic Bluetooth reaches a percentage below 50%, to be precise 20%. In addition, BLE has a longer effective range of up to 420 cm. The stable state occurs up to 390 cm and begins to decrease at a distance of 420 cm to 70%. However, at a distance of 450 cm BLE has lost its effective range because it only has a percentage below 50%, to be precise at a value of 20%. From this data it can be concluded that the BLE protocol has a better range than classic Bluetooth when there are no obstacles, plus it is supported by a better average percentage of BLE success with a value of 92.67% compared to classic Bluetooth which has a value of 82%.

To make it easier to read the data, it can be seen from the graph in Figure 15.

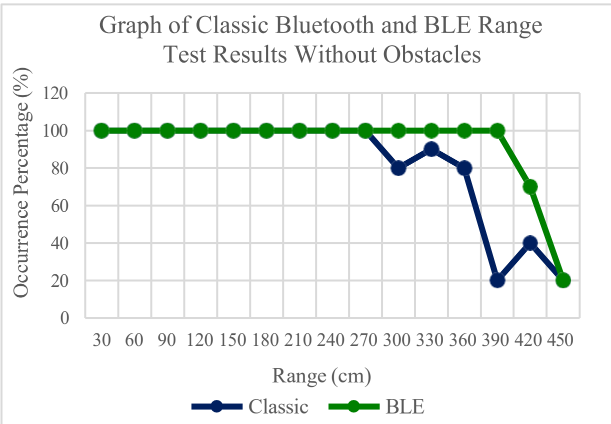

Figure 15. Unobstructed Reach Graph

B. *Analysis of Reach Test Results with Obstacles*

Based on data in terms of range with obstacles, it can be concluded that the effective range of classic Bluetooth can only reach 330 cm. This can be seen at a distance of 270 cm. Classic Bluetooth begins to experience a decrease in its ability to receive full power and only reaches 70%. This instability continues until the 360 cm range of classic Bluetooth reaches a percentage below 50%, to be precise 30%. In addition, BLE has a longer effective range of up to 390 cm. Even though it had decreased at a distance of 180 cm to a value of 90%, this value peaked again until it reached 270 cm. Then a gradual decrease occurs starting at a distance of 300 cm to 90%. However, at a distance of 420 cm BLE has lost effective range because it is unable to detect data at all or 0%.

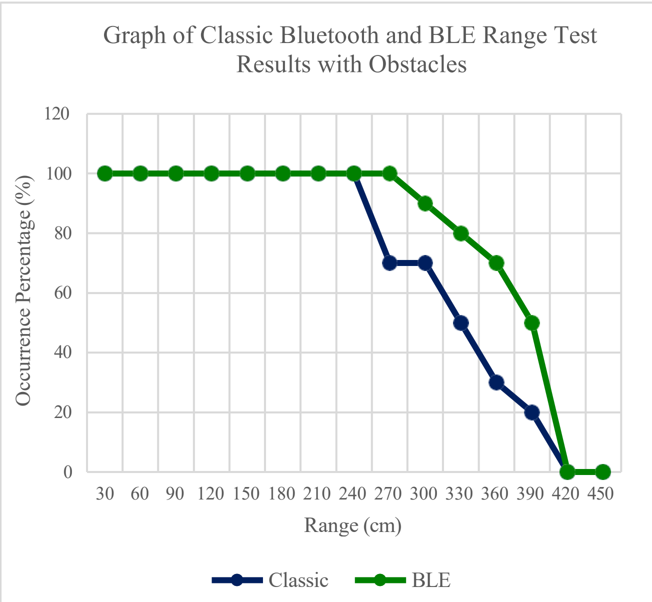

Figure 16. Reach Graph with Obstacles

From this data it can be concluded that the BLE protocol has a better range than classic Bluetooth when there are obstacles, supported by a better average percentage of BLE success with a value of 78.67% compared to classic Bluetooth which has a value of 69.33%. To make it easier to read the data, it can be seen from the graph in Figure 16.

C. *Analysis of Unobstructed Power Test Results*

Data from power testing results from both protocols shows that as the distance increases, the average RSSI value received becomes smaller with the average RSSI value for classic Bluetooth being higher than BLE in unobstructed conditions. This can be seen from both protocols that the distance of 30 cm has the highest RSSI value. Classic Bluetooth has a value of -59.6 dBm or the equivalent of 1.09×10^{-6} mW, while for BLE it has a value of -69.6 dBm or the equivalent of 1.1×10^{-7} mW. Apart from that, it can be seen that the average RSSI value for classic Bluetooth has a higher RSSI value reception with a value of -83.22 dBm compared to BLE which has a value of -87.98 dBm. To make it easier to read the data, you can see Figure 17.

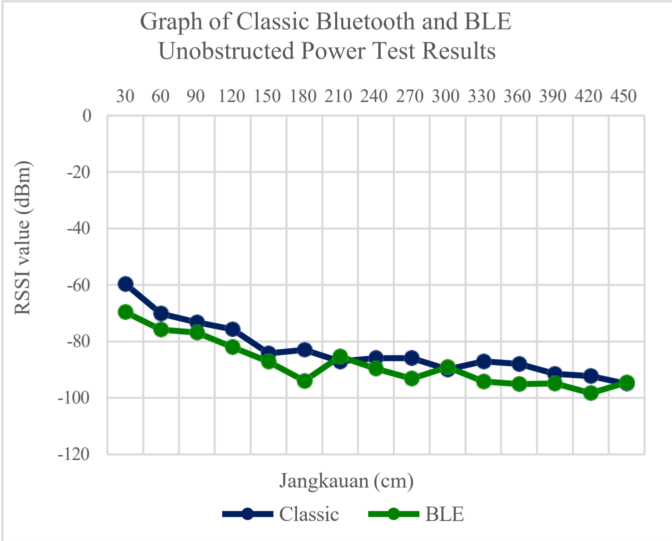

Figure 17. Unobstructed Power Chart

D. *Analysis of Power Test Results with Obstacles*

Data from power testing results from both protocols shows that as the distance increases, the average RSSI value received becomes smaller with the average RSSI value for classic Bluetooth being higher than BLE in conditions with obstacles. This can be seen from both protocols that the distance of 30 cm has the highest RSSI value. Classic Bluetooth has a value of -67.1 dBm or the equivalent of 1.95×10^{-7} mW, while for BLE it has a value of -75 dBm or the equivalent of 3.16×10^{-8} mW. Apart from that, it can be seen that the average RSSI value for classic Bluetooth has a higher RSSI value reception with a value of -89.56 dBm compared to BLE which has a value of -91.63 dBm. To make it easier to read the data, you can see Figure 18.

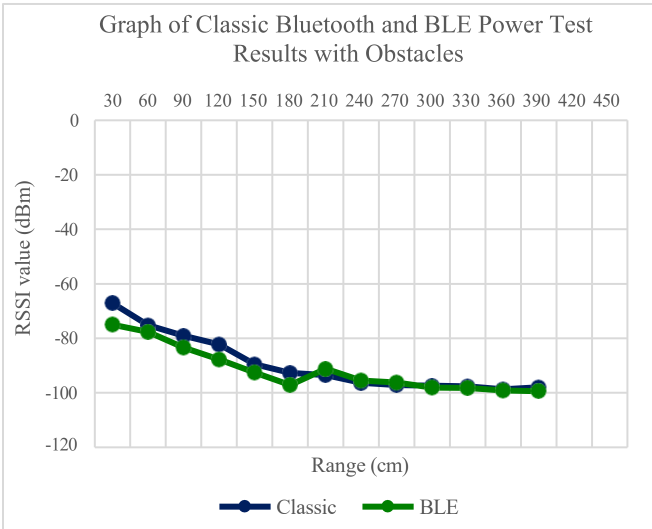

Figure 18. Power Graph with Obstacles

E. *Analysis of Test Results Comparing Duration and Number of Touches*

The results of comparing the duration and number of touches from the two protocols can be seen in Subsection

4.2.3.5. This data shows that the classic Bluetooth protocol is faster to access than BLE. This can be seen from classic Bluetooth which requires time to prepare applications to cross which is faster in terms of time and requires less interaction when using a smartphone. Classic Bluetooth has a total time for application access of 6 seconds. This consists of zero seconds of installation, which means it does not require time to install third-party applications and 6 seconds to activate the application, which is the smartphone's built-in Bluetooth feature. On the other hand, BLE has a total time of 86 seconds with 55 seconds for third-party application installation and 31 seconds for application activation. Apart from that, in terms of the number of smartphone interaction touches, classic Bluetooth has fewer interactions, namely 4 when activating the application before crossing over, while BLE has a total of 8 interactions calculated when the application is installed. To make it easier to read the data, you can see Figure 19 and Figure 20.

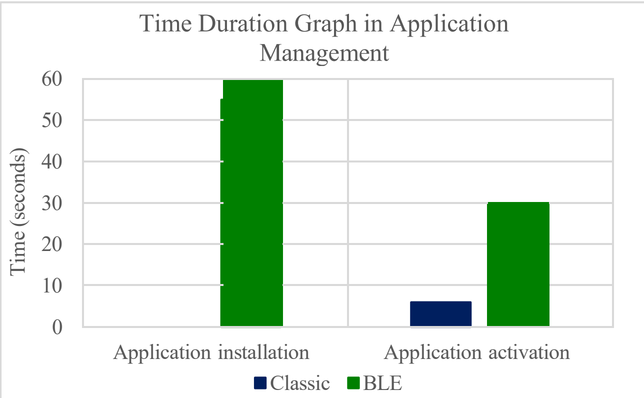

Figure 19. Time Duration Graph in Application Management

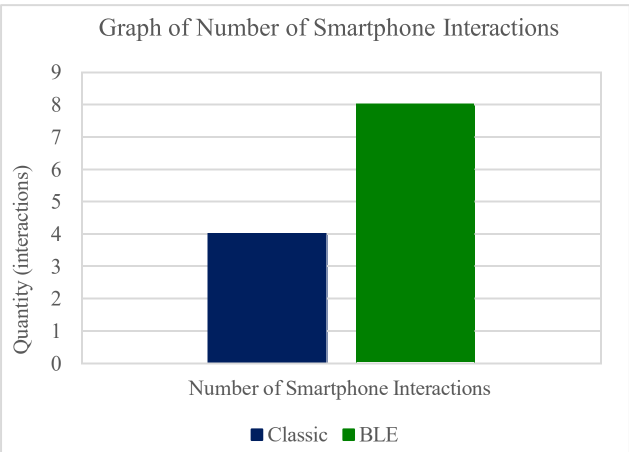

Figure 20. Graph of Number of Smartphone Interactions

The time required and the number of access touches using the classic Bluetooth protocol has less value than BLE. This is because classic Bluetooth does not require additional third-party applications and only requires activating the Bluetooth feature on the smartphone. This is different from the BLE protocol which requires a third-party application to activate the advertising feature on BLE smartphones. Required applications such as EFR Connect, nRF Connect or similar applications.

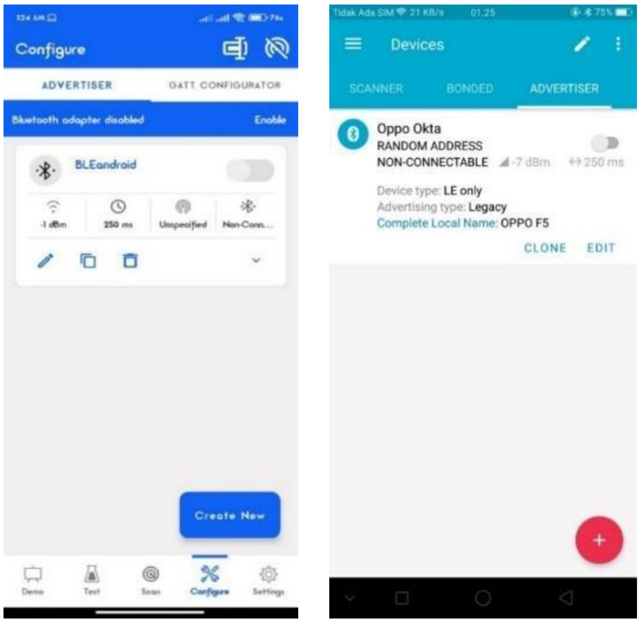

Figure 21. EFR Connect (left) and nRF Connect (right) displays

IV. CONCLUSION

The BLE protocol has a better range than classic Bluetooth when there are no obstacles or when there are obstacles. Supported by the average success percentage, BLE is better with a value of 92.67% in an unobstructed state and 78.67% in an obstructed state compared to classic Bluetooth which has a value of 82% in an unobstructed state and 69.33% in an obstructed state. As the distance increases, the average RSSI value received becomes smaller with the average RSSI value of classic Bluetooth being higher than BLE in both unobstructed and obstructed conditions. Supported by an average RSSI value for classic Bluetooth, it has a higher reception RSSI value with a value of -83.22 dBm in an unobstructed state and -89.56 dBm in an unobstructed state compared to BLE which has a value of -87.98 dBm in an unobstructed state. blocked and -91.63 dBm in the blocked state. In terms of duration and number of touches, the classic Bluetooth protocol has less duration and number of touches than BLE. This is because classic Bluetooth does not require using third-party applications and making certain settings. Classic Bluetooth only has a total time of 6 seconds for installing the application and activating the application compared to BLE, which is 86 seconds. Classic Bluetooth also has a smaller number of interactions, namely 4 when the application is activated, while BLE has a total of 8 interactions.

REFERENCES

[1] Espressif Systems, "Espressif," 2 February 2023. [Online]. Available: https://www.espressif.com/sites/

default/files/documentation/esp32-wroom-32_datasheet_en.pdf. [Accessed 30 March 2023].

[2] R. Ahmad, R. Riai Oktavianus Sasue, D. Wahyu Hidayat, A. Soimun, A. Prima Gilang Rupaka, Y. Oktopianto and A. Budi Sulistyo, "Pedestrian User-Friendly Intelligent Crossing Advance for Improved Safety," *Jurnal Keselamatan Transportasi Jalan,* vol. 9, no. 1, pp. 71-79, 2022.

[3] M. Azzam, "Pelican Crossing Depan Lapangan Karangpawitan Dirusak, Masyarakat Kesulitan Menyeberang Jalan," TribunBekasi.com, 23 October 2021. [Online]. Available: https://bekasi.tribunnews.com/2021/10/23/pelican-crossing-depan-lapangan-karangpawitan-dirusak-masyarakat-kesulitan-menyeberang-jalan. [Accessed 26 March 2023].

[4] D. Bazaraa, "Light-Fingered Thieves Steal 10 Signal Boxes from Same Pelican Crossing That They Then Sell on EBay as Novelty Doorbells in £32,000 Spree," DailyMail, 25 November 2022. [Online]. Available: https://www.dailymail.co.uk/news/article-11469141/Thieves-steal-10-signal-boxes-pelican-crossing-sell-eBay-32-000-spree.html. [Accessed 25 March 2023].

[5] C.-R. Dow, H.-H. Ngo, L.-H. Lee, P.-Y. Lai, K.-C. Wang and V.-T. Bui, "A Crosswalk Pedestrian Recognition System by Using Deep Learning and Zebra-Crossing Recognition Techiniques," *Software: Practice and Experience,* vol. 50, no. 5, pp. 630-644, 2019.

[6] M. Helmi, "PARAH...!! Fasilitas Pelican Crossing Penyebrang Jalan Tak Ada yang Berfungsi," Radar Banjarmasin, 22 July 2022. [Online]. Available: https://radarbanjarmasin.jawapos.com/banua/15/07/2022/parah-fasilitas-pelican-crossing-penyebrang-jalan-tak-ada-yang-berfungsi/. [Accessed 24 March 2023].

[7] Muliadi, A. Imran and M. Rasul, "Pengembangan Tempat Sampah Pintar Menggunakan ESP32," *Jurnal Media Elektrik,* vol. 17, no. 2, pp. 73-79, 2020.

[8] N. Polri, "World Day of Remembrance for Road Traffic Victim, Korlantas: Fatalitas Kecelakaan Bermula dari Sebuah Pelanggaran," 19 November 2022. [Online]. Available: https://ntmcpolri.info/world-day-of-remembrance-for-road-traffic-victim-korlantas-fatalitas-kecelakaan-bermula-dari-sebuah-pelanggaran/. [Accessed 16 February 2023].

[9] M. Purnawarman, E. Prasetyo Wibowo, S. Bahri Musa and I. Baihaqi, "Pelican Crossing System for Control A Green Man Light With Predicted Age," *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer,* vol. 21, no. 2, pp. 293-306, 2022.

[10] Rokom, "Kasus COVID-19 Varian Baru Masih Ditemukan, Perkuat dengan Vaksinasi," Kementerian Kesehatan Republik Indonesia, 21 February 2023. [Online]. Available: https://sehatnegeriku.kemkes.go.id/baca/rilis-media/20230221/3042441/kasus-covid-19-varian-baru-masih-ditemukan-perkuat-dengan-vaksinasi/. [Accessed 1 April 2023].

[11] W. Sahari and S. Widodo, "Kajian Kebutuhan Fasilitas Penyeberangan Pada Ruas Jalan Di Area Komersial Kota Pontianak (Studi Kasus: Jl. Teuku Umar – Jl. Hos Cokroaminoto, Kota Pontianak)," *JeLAST : Jurnal PWK, Laut, Sipil, Tambang,* vol. 2, no. 2, 2016.

[12] D. P. U. D. J. B. M. D. B. Teknik, Tata Cara Perencanaan Fasilitas Pejalan Kaki di Kawasan Perkotaan, Jakarta, 1995.

[13] K. Townsend, C. Cufi, A. and R. Davidson, "Protocol Basics," in *Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power Networking*, Sebastopol, CA, O'Reilly Media, Inc., 2014, pp. 15-34.

[14] M. Zhang, J. Zhao, S. Chen and W. Gong, "Reliable Backscatter with Commodity BLE," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, 2020.

[15] WHO, Global Status Report on Road Safety, Switzerland: WHO, 2018.

[16] M. Rusdi and A. Yani, "Sistem Kendali Peralatan Elektronik Melalui Media Bluetooth Menggunakan Voice Recognition," *Journal of Electrical Technology,* vol. 3, no. 1, pp. 27-33, 2018.