Proceeding of EECSI 2018, Malang - Indonesia, 16-18 Oct 2018

UUID Beacon Advertisements For Lecture Schedule Information

Wiwin Agus Kristiana
Faculty of Computer Science
Narotama University
Surabaya, Indonesia
wiwin.agus@narotama.ac.id

Mochamad Mizanul Achlaq
Faculty of Computer Science
Narotama University
Surabaya, Indonesia
mochamad.mizanul@narotama.ac.id

Benediktus Anindito
Faculty of Computer Science
Narotama University
Surabaya, Indonesia
benediktus.anindito@narotama.ac.id

Aryo Nugroho
Faculty of Computer Science
Narotama University
Surabaya, Indonesia
aryo.nugroho@narotama.ac.id

Cahyo Darujati
Faculty of Computer Science
Narotama University
Surabaya, Indonesia
cahyo.darujati@narotama.ac.id

Moh Noor Al Azam
Faculty of Computer Science
Narotama University
Surabaya, Indonesia
noor.azam@narotama.ac.id

***Abstract*—Smartphone users are increasingly diverse in using their phones. Some tasks that monitored through the bulletin boards or computer screens, lately it can be done anywhere with a mobile phone while on the move. Similarly, the features in smartphones are increasingly following the development of communication technology. One of them is Bluetooth version 4, which currently can always be available on all types of smartphones. Even for entry-level phones that commonly used by students, nowadays are equipped with the new version of Bluetooth. In this paper discussed the application of BLE or Bluetooth Low Energy, which is part of Bluetooth version 4, to provide the information about availability, lecture schedule, and lecture room at the University Narotama. By using this BLE communication technology all smartphones equipped with BLE, enabling the NARO-MOBILE application and residing in the campus environment, will receive all the latest information provided by SIMNARO - Narotama University Management Information System, in a real-time.**

Keywords—Internet of Things, IoT, Internet, Bluetooth, Bluetooth Low Energy (BLE)

I. INTRODUCTION

Narotama University is one of the private universities in Indonesia located in Surabaya City. Pawiyatan Gita Patria Foundation in 1981 developed this university for the first time, by beginning the establishment of law faculty in cooperation with lecturers from the Airlangga University.

As one of the 3,276 private universities in Indonesia [1], Narotama has relatively rapid development. Since the late the 1990s, Narotama has launched information technology-based services for all its stakeholders. SIMNARO (Sistem Informasi Manajemen Narotama or Narotama Management Information System) is the name of the developed system which currently has some essential services.

The SIMNARO essential services as follows:

- Academic information system, which is the primary foundation of services to the students. In this system, the students can plan their lectures, choose their classes and get much information about the course that followed.
- ELINA or E-Learning Narotama is an application that uses the Content Management System (CMS) Moodle to use as a basis for online learning over the Internet [2].
- Sistem Informasi Sumberdaya Terintregasi (SISTER or Integrated Resource Information System), which is a system from the Ministry of Research Technology and Higher Education (Kementerian Riset Teknologi dan Pendidikan Tinggi - Ristekdikti) that is unity in SIMNARO services.
- Quality Assurance Information System, also as one of the primary system because Narotama University has complied with ISO 9001: 2008

As the development of SIMNARO services, this paper has piloted a way to broadcast information to all SIMNARO users in a campus environment via Bluetooth Low Energy (BLE) directly to their smartphone. This method is thought to be quite effective rather than placing information on a bulletin board or asking students to access a server to get the information they need all the time.

II. BLUETOOTH LOW ENERGY

Bluetooth Special Interest Group (SIG) has defined BLE as part of Bluetooth version 4 which has lower power requirements than previous classical versions of Bluetooth. BLE uses a frequency that is also used by Bluetooth classic, which is 2.4 GHz. Nevertheless, BLE is not compatible with standard Bluetooth in its communications [3].

BLE is primarily designed to transfer data with a small size of data and sparse sending, usually is a simple data, so that the power required is minimal. This design is of course very different from the traditional Bluetooth design, which commonly used earlier. To accommodate the connection between the old model and the new one, Bluetooth SIG then introduces two kinds of trademarks. The first is Bluetooth Smart (BS) for devices that only support BLE, and the second is Bluetooth Smart Ready (BSR) for tools that can work either in Bluetooth classic mode or Bluetooth Low Energy mode.

Currently, a lot of different types of accessories that collaborate with mobile devices such as smartphones, tablets, and notebooks have long used BLE in their standards. Personal devices like smartwatches or others wearable devices are also equipped with Bluetooth 4.0 or later also it has BLE functionality.

Currently, a lot of different types of accessories that collaborate with mobile devices such as smartphones, tablets, and notebooks have used BLE in their standards. Personal devices like smartwatches, smart shoes or others wearable devices are also equipped with Bluetooth 4.0, or later to make

978-1-5386-8402-3/18/$31.00 ©2018 IEEE

it has BLE functionality. Even on medical devices, BLE also used to help people to live more comfortable and healthy.

The iBeacon is the famous technological name that uses BLE. It introduced by Apple Inc. in middle of 2013. BLE on this device is used to broadcast unique information to the nearest mobile device that has also been equipped with BLE [4]. Equalized to the classical Bluetooth, the iBeacon signal using BLE is expected to have the same coverage area but with lower power consumption.

The application of Indoor LBS or Location Based Services is one of the examples of comprehensive implementation of BLE in iBeacon. The applications inside the BLE observer device (usually smartphone devices) will calculate the strength of RSSI (Received Signal Strength Indication) the BLE signal it receives and then compared to the signal strength during the transmission performed by BLE Broadcaster[3]. The relative distance between the BLE observer and the BLE broadcaster could calculate by using the RSSI or Isotropic loss of free-space radiation calculation (equation 1)[3].

$$\text{RSSI (dBm)} = -10n \log10 (d) + A \quad (1)$$

The BLE observer receives all signals transmitted by all nearby BLE broadcasters. Therefore, it is necessary to identify a specific signal received from the BLE broadcaster. Thus, each BLE broadcaster continually sends data in its particular Universal Unique Identifier (UUID) form.

In addition to sending the data in the form of UUID as above, BLE broadcaster also transmits its signal strength. Of all these identifications, the application on the BLE observer can calculate the relative distance of all BLE broadcasters around it.

Currently, most smartphones, with the latest iPhone, Windows Mobile, Blackberry and Android operating systems, are equipped with Bluetooth version 4 compatible with BLE technology. Therefore, all smartphones can perform collaborative operations with iBeacon [3] [5].

A. The BLE Stack

BLE has two complementary layers to accommodate the highest performance (Figure 1) and is attributed to the lower layer and the upper layer. These two layers are separated by an adjustable control function from the higher layer to do something on the lower layer.

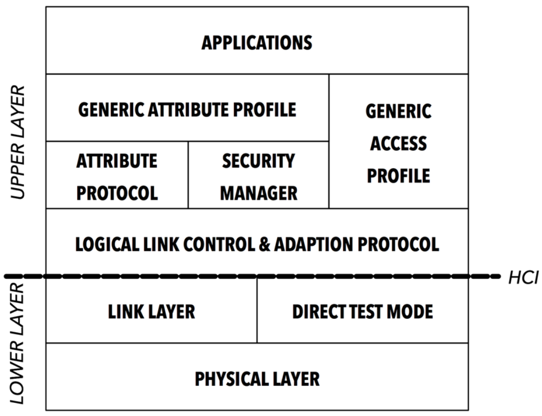

Fig. 1. Bluetooht Low Energy Stack[3-5]

Physical is the most bottom layer, contains an analog communication circuit, and capable of doing the modulating and demodulating analog to digital and vice versa. The radio frequency used is the ISM (Industrial, Scientific, and Medical) band of 2.4 GHz, and partitioned into 40 channels, ranging from 2.4000 GHz to 2,4835 GHz.

Link Layer is the part that interacts directly with the physical layer. This layer is the only real-time hard drive constrained layer of the entire protocol stack since this layer must be responsible for meeting all the time requirements defined by the Bluetooth 4.0 Core specification. Therefore, this layer is usually isolated from the higher layer of the protocol stack by providing an interface that hides the complexity and real-time requirements of the layers above it.

The Link Layer is the element that interacts instantly with it is below layer, that is the physical layer. This layer is the only real-time hard drive layer of the entire protocol stack because this layer must be responsible for meeting all the time requirements specified by the Bluetooth 4.0 core specification. Consequently, this layer usually is isolated from the higher layers of the protocol stack by providing an interface that hides the complexity and real-time requirements of the layers beyond it.

The Logical Link Control and Adoption Protocol have two primary functions, namely multiplexer protocol that gets data from layers above it and then packs it in standard BLE format and vice versa. This layer also performs the fragmentation -if the packet obtained from the above it is more than 27 bytes and then forwarded to the layer below it, and conducts recombination -if it is received from the layer below it some packets and then made a whole package for the layer above it [6-9].

The Attributes Protocol is a manageable stateless client/server protocol used by a BLE device. In the BLE, each device is a client, server, or also perhaps a combination of both, no matter whether it is a master or a slave. A client is a machine that requests data from the server, and a server is a device that gives the data to the client. Each server contains data organized in the form of properties, each of which is assigned a 16-bit attribute, a UUID or Universally Unique Identifier, a pair of permissions information, and multiple of values [9].

The Security Manager Protocol is a combination of a protocol and a series of security algorithms that will conduct the protocol stack capabilities to deliver and exchange security keys. This protocol would enable the device to transfer data securely through an encrypted communication, to ascertain the trust of the identity of the remote device, as well as to mask the Bluetooth Address if it needed.

Generic Attribute Profile is above of ATT. The job is adding hierarchy and data abstraction model above ATT. From the one side, this protocol can be avowed the backbone of BLE data transfer because it describes how the data is prepared and transferred within the applications. GATT represents a general data object that can frequently be used by multiple applications profiles that are known as GATT-based profiles. GATT keeps the same client/server architecture on the ATT layer, but the data is packaged in services, consisting of one or more characteristics. Each characteristic can be recognized as the integration of a piece of user data along with metadata, which includes the description information such as properties, usernames, sections, et cetera. [6-9]

The Generic Access Profile determines how the devices interact with each other at a lower level, beyond the actual protocol stack. GAP can be considered to define the top BLE control layer, given that it determines how the device carry out the control methods such as the discovery of the equipment, connection, the establishment of security, and more, to ensure interoperability and to enable data communication to take place between devices from various vendors. GAP assign different device rules and concepts for organizing and standardizing low-level operation of devices, such as roles and interactions, operational and transition modes, operational procedures for achieving consistent communication, and security aspects, including security modes and processes, also the additional data formats for data non-protocol.

B. *Universal Unique Identifier (UUID)*

iBeacon is the first BLE Beacon technology released by Apple. Therefore most beacons take inspiration from the iBeacon data format. In general, the beacon will broadcast four data similar to figure 2. [8][9]

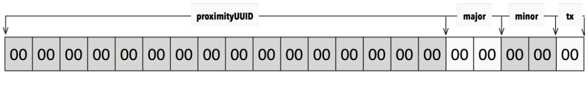

Fig. 2. BLE UUID Format[3-5]

- UUID: 16 bytes of unique data that identifies which beacon that transmit.
- Major: 2 bytes decimal number that could be used to determining the subset number of a large group of beacons within an area.
- Minor: 2 bytes decimal number that could be used to identifying a specific beacon.
- PWR: 1 byte TX power level, indicating the signal strength of the BLE device when transmitted.

The UUID, Major, and Minor data can be used as the location information where is the BLE broadcaster located. Meanwhile, using the PWR information, the application can be calculated relative distance between BLE broadcaster and BLE observer with the calculation of RSSI in equation 1.

By using the calculation of triangulation, the data from several BLE broadcasters that received by the BLE observer can be a clue the current location of the BLE observer somewhere in the room.

III. UUID Encode Design

This research modifies UUID, Major, and Minor contents to broadcast much variety of information through the BLE, especially the information about the status of the courses required by lecturers and students within the Narotama campus.

Meanwhile, the width of the UUID is only 16 bytes in single broadcasting. That is why the data must be encoding first before it broadcasted. The apps that exist within the smartphone as a BLE observer then must first translate the UUID data before it is displayed on the screen.

From some studies, the distance of the BLE signal is limited to only about 20 meters from the broadcaster[3] and will be shorter if there are obstacles around the location. Therefore, broadcasting information with BLE requires multiple BLE broadcasters in one area. This situation occurs in the possibility of overlapping information received by BLE observer from more than one BLE broadcaster. Therefore, the UUID design should be able to anticipate this possibility, so the app does not need to store and display previously received information, which can make the CPU and memory smartphone exhausted.

Figure 3 shows the block diagram of the UUID format used.

Fig. 3. Modified UUID

- DEPT, a 1 byte, is a code of the Department corresponding to the UUID. This information will use by apps to determine whether the UUID data transmission needs to be saved and displayed or forget. This data must be compared to the lecturer or student identification number based on the settings that made by the user.
- CIDN, 3 bytes, is representing the Course Code Identification Number just like the course code in SIMNARO. The name of the course will be stored in SQLite at mobile devices and must be updated by the application if needed.
- LID, 4 bytes, is the Lecturer ID. The LID is representing the staff number ID of Narotama University that all Narotama lecturer also has it. Just like the CIDN, the full name of lecturer will be stored at SQLite and updated by the application if required.
- ROOM, 2 bytes or 4 digit hexadecimal, representing the classroom number. The standard of Narotama in room number is one digit alphabet (A to Z) as the building code and followed by floor and room number. As an example, there is a room on building A at 7th floor and room number 3, so it has called A-701. Therefore the UUID is encoded in 2 forms, the first hexadecimal is the ASCII code for the building alphabet ("A" or 0x41 to "Z" or 0x5A) and then followed by the next three hexadecimal as the numerical (0 through 4095) that representing floor and room number.
- CODE, 1 byte, is representing the Status Code of the class. The code are "0x10" for the class is available, "0x11" class is pending, "0x12" the class is moved, and "0x1F" the class already begins.
- The rest of 5 bytes UUID not used.

Meanwhile, the Major and Minor data fields, which previously used to identify of the grouping of the beacons within a broad group, in this model of information it will use as the data ID of the broadcasted information. This data ID is required to anticipate the possibility if the data is received more than once by the BLE observer at the smartphone. When it happens, the apps do not necessarily process it as the new information.

In the making of the data ID, it uses two digits of the year followed by two digits of the month and then followed by two digits of the date, and at the end, there will be a sequences number from 0 to 999. This data format is in the decimal form, of course, it has to convert in hexadecimal before its use as the data major and minor. For example, the 157th data that submitted on June 1, 2018, will have the data ID 180601157 or 0xAC3C145 in hexadecimal. So it will be written 0x0AC3 as major and 0xC145 as a minor.

IV. The Broadcaster

This research used some prototype computer Raspberry Pi 3 which already equipped with Bluetooth version 4 for the broadcasters (figure 4).

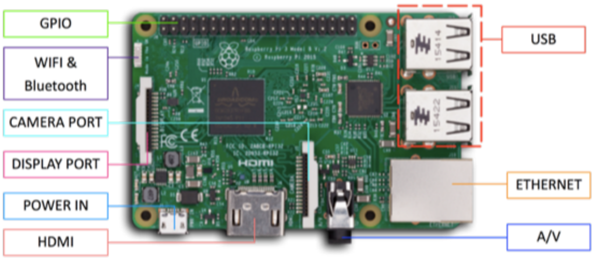

Fig. 4. Raspberry Pi 3 model B

A. The Hardware and Network

Raspberry Pi is a single-board prototype computer that low-cost, and high-performance that developed by the UK-based Raspberry Pi Foundation. This foundation concentrates on placing digital power in the hands of people around the world so that the entire world community can understand and shape the ever-growing digital world, can solve their problems, and can use the digital power for their future work.

One that the foundation does is to promote the teaching of basic computer science in schools, especially in developing countries, using this low-cost computer.

The newest version is the Raspberry Pi series 3 model B+, which equipped with:

- A 64-bit quad-core ARM Cortex-A53 CPU with speed at 1.4GHz;
- 802.11ac WiFi LAN;
- Bluetooth 4.2;
- Gigabit Ethernet;
- Power-over-Ethernet support;
- Improved PXE network and USB mass-storage booting; and
- Advanced thermal management

This new Raspberry Pi still carry out 1 GBytes of Random Access Memory (RAM) and also still has four USB slots, HDMI output, and 3.5 mm jack for audio. Besides, for digital communication, this board is provided with 40 pins General-Purpose Input/Output (GPIO) that supports standard communication protocols such as Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI).[10] [11].

The network topology that established in this research using the scheme like figure 5. All BLE broadcasters (Raspberry Pi) not connected as one network of the server farm. It protected from another network to direct access. The access to the BLE broadcaster will be allowed only from the broadcast management server.

The UUID and data ID are constructed on the broadcast management server. It calculates based on the information give by SIMNARO server, the time of the server, and the sequence number of the constructed data broadcast.

The constructed UUID and data ID then will be kept by the broadcast management server to broadcast several times and to several BLE broadcasters as long as the data has permission to transmit. One of the parameters is when the constructed data do not exceed the schedule of the lecture.

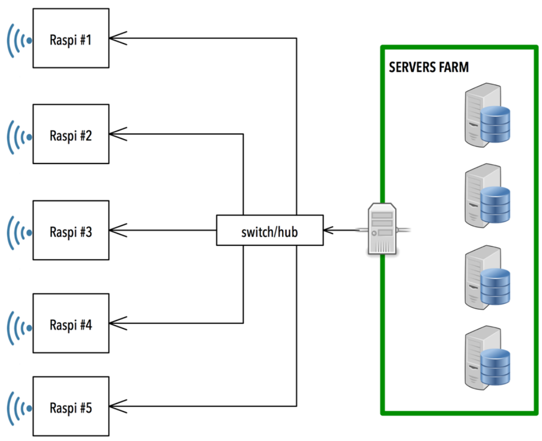

Fig. 5. Experiment Network Topology

B. The Logical

In this study, each Raspberry Pi -that acting as a BLE broadcaster, are installed the Application Programming Interface (API) that uses the Python programming language, as the standard programming language for Raspberry Pi [12], and comes with the Django framework for APIs could be accessed by methods communication Hypertext Transfer Protocol (HTTP) [13]. For security reasons, this API is created only accessible by the IP address of the broadcast management server via the Transport Control Protocol (TCP) connection using port 8080.

As described above, the broadcast management server responsible for creating the UUID and the data ID before it sent to Raspberry Pi and then broadcast via BLE. Therefore, the API on Raspberry Pi will only get only three parameters, which are the UUID, the data ID, and the MD-5 hash of all parameters as the security and error protection. The data ID that received by API then will be translated into Major and Minor fields in the data beacons. Meanwhile for the transmitting signal data field will be filled with the value of 0x00.

When all data ready, these data beacons are broadcasted by Raspberry Pi via the Blue-Z library -the Linux kernel standard library for Bluetooth communication.

The flowchart application for API inside the Raspberry Pi is in figure 6.

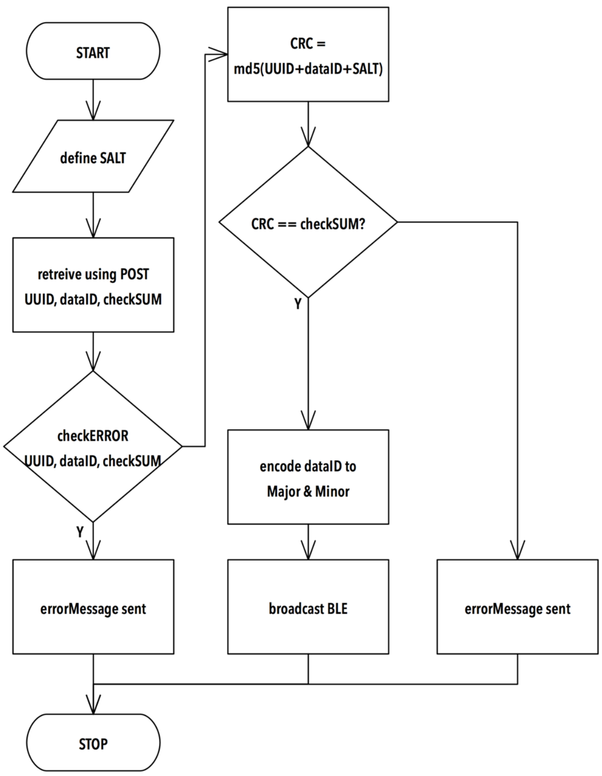

Fig. 6. The Flowchart of BLE Broadcaster

The API will start working when receiving access requests from the broadcast management server. The first API will define a static SALT variable that contains a case-sensitive token. This token must be the same as that in the broadcast management server. Next, the API will perform data retrieval via HTTP with the POST method. API does not access with other methods.

Then the data that has been taken by the API then checked on the variables named UUID, dataID, and CheckSum. All of these variables must be filled with value and should not be empty or contain spaces.

The next step is to calculate the MD-5 hash from the combined the value of UUID, dataID, and the SALT, which is then compared to the contents of the CheckSum variable. If the value is the same, then the process can proceed, but if not then an error message is sent to the sender server, and then the program stops.

If the process passes on, then the dataID separated into 2 bytes data Major and 2 bytes data Minor. As mentioned earlier, Major and Minor data will be as the primary key whether the BLE observer has received the same information or not.

The process then will be ended by broadcasting all the information above and the API back standby receiving requests from the broadcast management server.

V. The Observer

In this experiment, we do testing a BLE observer using Xiaomi Redmi 5A smartphone that already equipped with Bluetooth 4.1, using the Android 7.1.2 (Nougat) operating system[14]. These experiments include receiving BLE data signals that are UUID, Major, Minor, and TX strength, as well as translation of UUID into the information displayed on the screen

On the application development side of this research using hybrid programming with Cordova framework[15] and assisted with some plugins that have been available in the market. The main reason for the use of hybrid programming in this experiment is the speed of application development, and the testing does not require advanced programming[16],[17].

A. The Logical

Figure 7 shows the application logic flowchart on the smartphones.

This application starts by checking on the Bluetooth device. If the device is not active, then the application will show the permission request to the user to activate it, and if the user then allows it then the next step is to turn the Bluetooth device on and then back to the first step, but if the user does not allow it, then the application will stop.

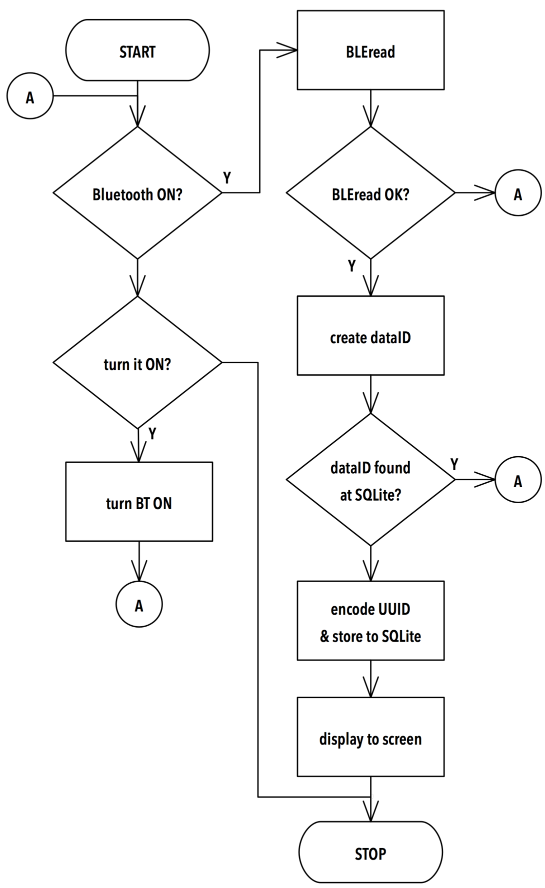

Fig. 7. The Flowchart of BLE Observer

Once the Bluetooth is detected, the next is application execute the method of BLE data read. If this method fails to get the data, it is mean there is no BLE signal captured. Then the application will repeat to the first step. If the method succeeds, then the return value of this method would contain BLE data. The application will get a set of information related to UUID, Major, Minor, and RSSI.

Next, the Major and Minor fields will be translated into the dataID. The results of this translation then use to query in SQLite. If the query found the same of the dataID, it is mean the data is already in the SQLite so then the application will return to the beginning at the origin of the application. If the dataID is not found, then the application will continue in the next step.

When the dataID is not found in SQLite, it means that UUID is not received by BLE observer yet. Therefore UUID needs to decoded to get the required information and store it in the SQLite.

The final step in this application is to show everything that is in SQLite to the screen display. Moreover, the last, application will return to the first step.

B. *The Test Result*

In this experiment, the testing of the system is performed by broadcasting several times of data when the smartphone at some locations that covered by the BLE signal. Then observed the BLE signal receiving by the device both when the smartphone is in still position or a move.

The Testing proves that the BLE broadcaster can send the information well, both when smartphones are in a state of immobile, or when the smartphone is in a state of reasonable moves. In all condition, the smartphone can receive the information that broadcasted by Raspberry Pi, decode the UUID to the information, and then display it on the screen.

Figure 8 is the screenshot of smartphones acceptance of the information.

Fig. 8. The Screenshot of Smartphone

Besides testing the concept of broadcasting data via BLE, in this testing also recorded the average RSSI of BLE signal received by the smartphone as the BLE observer and the estimated distance between the smartphone to the Raspberry Pi as a BLE broadcaster without considering the obstacles that exist between them (table 1).

As can be seen in Table 1, the signal strength will decrease if the smartphone away from the Raspberry Pi location. However, in some locations, the declining of the signal can also be caused by an obstacle between the smartphone and Raspberry Pi.

TABLE I. RSSI AND APPROXIMATE DISTANCE

Location	BLE #1	BLE #2	BLE #3	BLE #4	BLE #5
#1	10 m -66dBm	15 m -85dBm	20 m -90dBm	> 30 m N/A	> 30 m N/A
#2	5 m -57dBm	10 m -80dBm	20 m -87dBm	> 30 m N/A	> 30 m N/A
#3	10 m -70dBm	10 m -88dBm	10 m -84dBm	> 30 m N/A	> 30 m N/A
#4	10 m -79dBm	5 m -56dBm	10 m -89dBm	> 30 m N/A	> 30 m N/A
#5	15 m -87dBm	10 m -83dBm	5 m -73dBm	> 30 m N/A	> 30 m N/A
#6	15 m -81dBm	10m -85dBm	15 m -90dBm	20 m -89dBm	20 m N/A
#7	20 m -80dBm	15 m -87dBm	20 m -95dBm	20 m -87dBm	20 m N/A
#8	>30 m N/A	20 m N/A	20 m N/A	15 m -78dBm	15 m -70dBm
#9	>30 m N/A	> 30 m N/A	>30 m N/A	15 m -80dBm	15 m -78dBm
#10	>30 m N/A	> 30 m N/A	>30 m N/A	10m -78dBm	15 m -74dBm
#11	>30 m N/A	> 30 m N/A	>30 m N/A	5 m -55dBm	20 m -85dBm
#12	>30 m N/A	> 30 m N/A	>30 m N/A	10 m -63dBm	5 m -58dBm

VI. CONCLUSION AND FUTURE WORKS

In this experiment, we have created the mechanism of broadcasting the status and schedule of lectures by using BLE media at Narotama University. This method is sufficient for lecturers and students to know their course schedule without having to move from their positions. With the regular basis broadcasting, no information not received by the students or lecturers as long as they stay within the campus - at an affordable location with BLE broadcaster signals.

Also, because of there is no human involvement required concerning sending or receiving information in this system, the possibility of information being undelivered for the reasons of forgetfulness -as the human nature, can be avoided.

Broadcasting systems that use BLE can load with various data for various purposes, but because of UUID's limited capacity, it is necessary to design the UUID as encoded information as flexible as possible, for all information broadcasting purposes.

The future work of this research is to change the UUID design so that it can be more flexible for the whole purpose of broadcasting information inside the Narotama campus. Because the information required is not just about the schedule and status of the course, but also for other information that supports university roadmaps to become leaders as universities using information and communication technology.

It even better if we could utilize this system for indoor LBS system too. It because naturally, BLE can use to show the location of a device relative to something around it. So we can have two systems on the same devices.

Other future works that can be developed from this paper are using this encoded UUID concept to broadcast all types of information in a limited location. For example, this encoded UUID concept can be used for communication between sensors and array actuators in a large data center.

REFERENCES

[1] Kemristekdikti, Statistik Pendidikan Tinggi Tahun 2017, PT 17, Indonesia, pp. 15, 2017. https://ristekdikti.go.id/epustaka/buku-statistik-pendidikan-tinggi-2017/

[2] L. de la Torre, M. Guinaldo, R. Heradio, and S. Dormido, “The Ball and Beam System: A Case Study of Virtual and Remote Lab Enhancement With Moodle,” IEEE Transactions on Industrial Informatics, vol. 11, no. 4, pp. 934–945, Aug. 2015. http://ieeexplore.ieee.org/document/7120976/

[3] M. N. Al-Azam, B. Anindito. "BLE Observer Device Menggunakan Raspberry Pi 3 untuk Menentukan Lokasi BLE Broadcaster." [BLE Observer Device Using Raspberry Pi 3 to Determine the BLE Broadcaster Location], Prosiding SNATIF, Indonesia, 2016,pp. 181-188, 2016. http://jurnal.umk.ac.id/index.php/SNA/article/view/646 [in Bahasa Indonesia]

[4] S.M.H. Sharhan, S. Zickau. “Indoor mapping for location-based policy tooling using bluetooth low energy beacons.” Wireless and Mobile Computing, Networking and Communications (WiMob) Proceedings, 19–21 October 2015 (Abu Dhabi, United Arab Emirates, 2015). IEEE, pp. 28–36, 2015. http://ieeexplore.ieee.org/document/7347937/

[5] Y. Yang, Zhouchi Li, K. Pahlavan. “Using iBeacon for intelligent in-room presence detection”. Cognitive Methods in Situation Awareness and Decision Support (CogSIMA) Proceedings, 21–25 March 2016, San Diego, CA, USA, 2016. IEEE, pp. 187–191, 2016. http://ieeexplore.ieee.org/document/7497808/

[6] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, C. Gomez. “IPv6 over BLUETOOTH(R) low energy.” Internet Engineering Task Force (IETF), RFC7668, 2015. https://tools.ietf.org/html/rfc7668

[7] R. Tabata, A. Hayashi, S. Tokunaga, S. Saiki, M. Nakamura, S. Matsumoto, “Implementation and evaluation of BLE proximity detection mechanism for pass-by framework.” International Conference on Computer and Information Science (ICIS), 26,Äì29 June 2016, Okayama, Japan, 2016. IEEE, pp. 1-6, 2016. http://ieeexplore.ieee.org/document/7550872/

[8] M. N. Al-Azam, M. M. Achlaq, A. Nugroho, A. Gabriel Sooai, A. Winaya, and Maftuchah, “Broadcasting the Status of Plant Growth Chamber using Bluetooth Low Energy,” MATEC Web of Conferences, vol. 164, p. 01029, 2018. https://doi.org/10.1051/matecconf/201816401029

[9] K. Townsend, R. Davidson, Akiba, and C. Cufí, “Getting started with Bluetooth low energy: tools and techniques for low-power networking”, Revised First Edition. Sebastopol, CA: O’Reilly, 2014. http://shop.oreilly.com/product/0636920033011.do

[10] M. M. Achlaq, M. N. Al-Azam, A. Winaya, Maftuchah, “Nawala Mangsa 2.0–Weather Station with BLE Broadcaster”, Journal of Telematics and Informatics (JTI), Vol 6, No 2, June 2018, 2018. http://section.iaesonline.com/index.php/JTI/article/view/448

[11] E. Upton, G. Halfacree. “Raspberry Pi user guide.” New Jersey: Willey Publishing. pp. 4. 2014. https://www.wiley.com/en-us/Raspberry+Pi+User+Guide%2C+4th+Edition-p-9781119264361

[12] S. Monk, “Programming the Raspberry Pi: getting started with Python, Second Edition.” New York: McGraw Hill Education, 2016. https://www.amazon.com/Programming-Raspberry-Pi-Second-Getting/dp/1259587401

[13] C. Darujati, M. Hariadi. “Facial motion capture with 3D active appearance models”. The 3rd International Conference Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 7–8 November 2013 (Bandung, Indonesia, 2013). IEEE, pp. 59–64, 2014. http://ieeexplore.ieee.org/document/6698465/

[14] A. Nugroho, S. Sumpeno, M. H. Purnomo, “Visualizing Interaction in Catfiz Indonesian Messenger Using Graph Coloring”, NICOGRAPH International 2015, pp. 1234-1237, 2015. https://www.researchgate.net/publication/279200763_Visualizing_Interaction_in_Catfiz_Indonesian_Messenger_Using_Graph_Coloring

[15] J.M. Wargo. PhoneGap essentials: Building cross-platform mobile apps. Boston: Addison-Wesley (2012). pp. 45. https://www.amazon.com/PhoneGap-Essentials- Building-Cross-platform-Version/dp/0321814290

[16] Cordova. Cordova plugin for ibeacon. Mobile apps with HTML, CSS & JS [online] from https://www.npmjs.com/package/cordova-plugin-ibeacon. [Accessed on 7 December 2017]

[17] T. S. Sollu, Alamsyah, M. Bachtiar, and A. G. Sooai, “Patients’ Heart Monitoring System Based on Wireless Sensor Network,” *IOP Conference Series: Materials Science and Engineering*, vol. 336, p. 012009, Apr. 2018.