

BLE OBSERVER DEVICE MENGGUNAKAN RASPBERRY Pi 3 UNTUK MENENTUKAN LOKASI BLE BROADCASTER

Moh. Noor Al Azam^{1*}, Benediktus Anindito²

¹Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Narotama

²Program Studi Sistem Komputer, Fakultas Ilmu Komputer, Universitas Narotama

Jl. Arief Rachman Hakim no. 51, Sukolilo, Surabaya 60117

*Email: noor.azam@narotama.ac.id

Abstrak

Penggunaan Bluetooth Low Energy (BLE) digunakan dalam layanan berbasis lokasi atau Location Based Services (LBS) di dalam ruangan -baik dalam format connected topology (CT) maupun broadcast topology (BT). Dalam format CT, peripheral device akan ditempatkan dalam tempat yang statis sementara central device (seperti smartphone) akan menterjemahkan data-data yang diterima dari peripheral device tersebut menjadi koordinat dimana dia berada saat itu. Sedangkan dalam format BT, BLE broadcaster lebih banyak diposisikan sebagai perangkat yang bergerak dan observer device cenderung berada dalam tempat yang statis - yang harus menterjemahkan BLE broadcaster yang mendekatinya dan mengambil sebuah keputusan. Pada penelitian ini digunakan Raspberry Pi 3 -yang sudah embedded BLE, sebagai observer device dan Cubeacon sebagai BLE broadcaster serta sebuah server sebagai pencatat posisi BLE broadcaster berada. Dari ujicoba yang dilakukan, penggunaan Raspberry Pi 3 dan Cubeacon dapat memberikan penentuan lokasi sampai dengan jarak kurang dari 1 meter sampai dengan maksimal 5 meter dari lokasi BLE broadcaster berada.

Kata kunci: Beacon, Bluetooth Low Energy (BLE), Bluetooth Smart Ready, proximity, RSSI

1. PENDAHULUAN

Bluetooth Low Energy (BLE) adalah bagian dari spesifikasi utama bluetooth versi 4. Spesifikasi tersebut pada mulanya dibuat sedemikian rupa sehingga diharapkan BLE bisa hadir lebih kecil dan dengan versi yang lebih optimal dari pendahulunya -bluetooth klasik. Tetapi ketika benar-benar hadir, BLE memiliki turunan dan desain yang sama sekali berbeda dari tujuan awal tersebut.

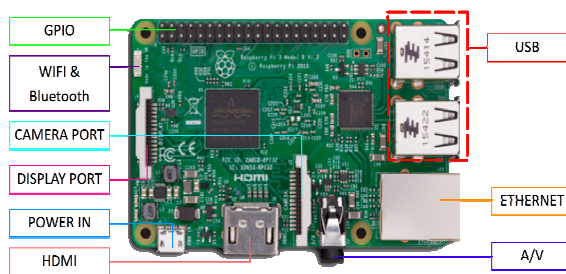
Yang membuat BLE menarik untuk dipelajari adalah bahwa itu teknologi yang tepat, dan dengan kompromi yang tepat serta dibuat pada waktu yang tepat. Untuk standar yang relatif baru (awalnya didisain oleh Nokia sebagai Wibree sebelum diadopsi oleh bluetooth Special Interest Group -bluetooth SIG, pada tahun 2010), BLE telah terlihat memiliki tingkat adopsi yang sangat cepat, dan dengan jumlah produk yang sudah BLE enable menempati peringkat terdepan dibandingkan teknologi nirkabel lainnya pada titik waktu yang sama dalam siklus rilis mereka.

Apple inc. -utamanya, telah berupaya secara signifikan dalam memproduksi modul BLE yang handal dan memberikan informasi tentang desain BLE-nya tersebut ke masyarakat. Strategi ini -pada gilirannya, mendorong vendor lain untuk ikut menyediakan sumber daya mereka pada teknologi yang diperkirakan akan sukses atau berkembang dalam jangka panjang. Teknologi yang dikembangkan oleh Apple inc. ini, saat ini dikenal sebagai iBeacon. Berbagai vendor telah membuat pemancar hardware iBeacon kompatibel -biasanya disebut beacon. Perangkat BLE yang dipancarkan sebagai pengenalan mereka untuk perangkat elektronik portabel di sekitarnya. Teknologi ini memungkinkan smartphone, tablet dan perangkat lain melakukan suatu tindakan ketika di dekat sebuah beacon. Salah satu beacon buatan anak bangsa dikenal dengan nama Cubeacon.

2. METODOLOGI

Dalam penelitian ini dibuat sebuah format broadcast topology (BT) dengan menggunakan Raspberry Pi 3 model B (RPI), sebagai observer devices dan Cubeacon, sebagai beacon atau BLE broadcaster.

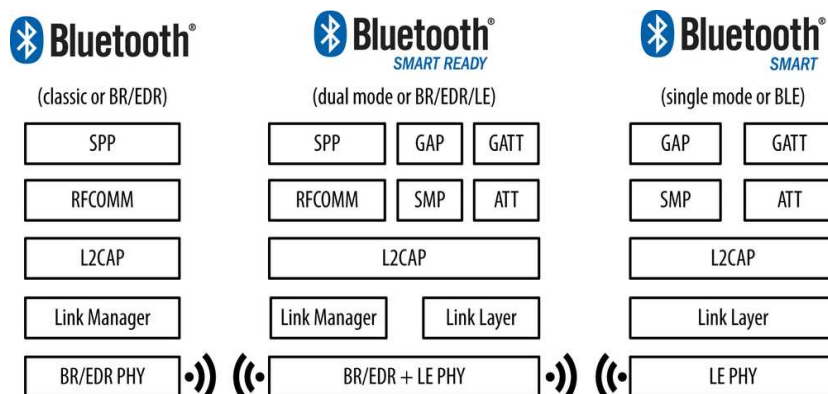
2.1. Raspberry Pi 3



Gambar 1. Raspberry Pi 3 dan port-portnya

Raspberry Pi adalah sebuah *single-board computer* yang dikembangkan di Inggris oleh *Raspberry Pi Foundation* untuk mempromosikan pengajaran ilmu komputer dasar di sekolah-sekolah dan negara-negara berkembang. Beberapa generasi *Raspberry Pi* ini telah dirilis. Generasi pertama (*Raspberry Pi 1*) dirilis pada bulan Februari 2012 di model dasar A dan lebih tinggi spesifikasi model B. Model A+ dan B+ yang dirilis setahun kemudian. Raspberry Pi 2 Model B dirilis pada bulan Februari 2015 dan Raspberry Pi 3 model B pada bulan Februari 2016 (gambar 1).

Semua model *Raspberry Pi* menggunakan sistem *Broadcom system-on-chip* (SoC), yang meliputi unit ARM kompatibel CPU dan pada *chip graphics processing unit* (GPU) ditanamkan VideoCore IV. Kecepatan CPU mulai dari 700 MHz sampai 1,2 GHz pada *Raspberry Pi 3* dan memiliki memori RAM dari 256 MB sampai dengan 1 GB. *SD Card* digunakan untuk menyimpan sistem operasi dan program baik dalam *SDHC* atau ukuran *MicroSDHC*. Beberapa tipe *Raspberry Pi* memiliki antara satu dan empat slot USB, *output* video komposit dan HDMI, dan 3,5 mm *jack* untuk audio. Untuk keperluan komunikasi dengan digital disediakan oleh sejumlah pin *General-Purpose Input/Output* atau GPIO yang mendukung protokol umum seperti I²C. Pada model B memiliki *port Ethernet* dan *Raspberry Pi 3* sudah dilengkapi dengan Wi-Fi 802.11n serta *Bluetooth*.



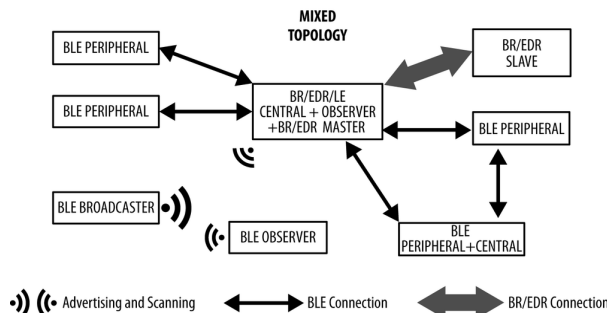
Gambar 2. Versi Bluetooth dan komunikasi antar tipe perangkat

Sebagai sistem operasi, pihak *Raspberry Pi* menyediakan *Raspbian*, sebuah distribusi Linux turunan Debian yang bisa diunduh. Beberapa pihak ketiga lainnya juga menyediakan sistem operasi yang bisa digunakan di *Raspberry Pi* -seperti *Ubuntu*, *Windows 10 IoT Core*, *RISC OS*.

Sebagai bahasa pemrograman, pihak *Raspberry Pi* mendorong penggunaan bahasa *Python* dan *Scratch* sebagai bahasa pemrograman utama. Meskipun demikian, beberapa bahasa pemrograman lain juga dapat digunakan pada sistem yang sama.

Pada tanggal 29 Februari 2016, *Raspberry Pi* meluncurkan *Raspberry Pi 3* yang di dalamnya sudah ditanamkan *Bluetooth 4.1 Low Energy*. Oleh karenanya *bluetooth* pada *Raspberry Pi 3* digunakan sebagai *Basic Rate/Enhanced Data Rate* (BR/EDR) atau yang lebih dikenal sebagai

classic bluetooth maupun sebagai *Low Energy (LE)*. Kemampuan ini lebih dikenal dengan sebutan *Bluetooth Smart Ready* -seperti pada gambar 2.

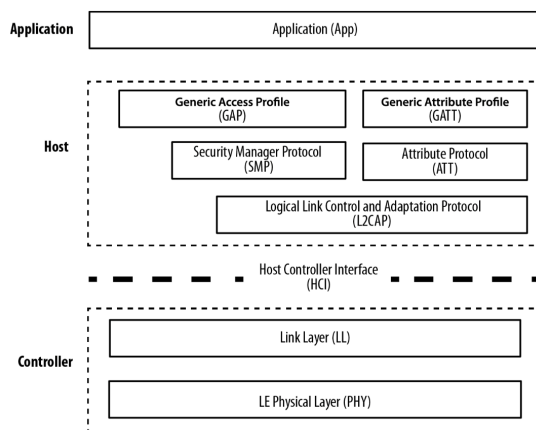


Gambar 3. Mixed Topology

Dengan kemampuan ini, maka *bluetooth* dalam *Raspberry Pi 3* dapat digunakan dalam *Mixed Topology* (gambar 3) baik sebagai komunikasi antar perangkat melalui *classic bluetooth*, maupun digunakan sebagai *BLE broadcaster* dan *observer device*, pada saat yang bersamaan.

2.2. *Bluetooth Low Energy*

Hampir sama dengan OSI Layer, dalam satu buah perangkat BLE terbagi dalam 3 layer tugas: *controller*, *host*, dan *application* (gambar 4).



Gambar 4. BLE Protocol Stack

PHY atau *LE Physical Layer* adalah layer terbawah yang berisi sirkuit komunikasi analog, memiliki kemampuan melakukan modulasi dan de-modulasi sinyal analog dan mengubahnya ke sinyal digital. Frekuensi radio yang digunakan adalah frekuensi ISM (*Industrial, Scientific, and Medical*) 2.4 Ghz dibagi dalam 40 *channel* dari 2.4000 Ghz sampai dengan 2.4835 Ghz.

LL atau *Link Layer* adalah bagian yang berinteraksi secara langsung dengan *PHY*. Layer ini biasanya diimplementasikan dalam kombinasi dari hardware khusus dan perangkat lunak. Layer ini adalah satu-satunya *hard real-time constrained layer* dari seluruh stack protokol yang dimiliki, karena layer ini harus bertanggung jawab untuk memenuhi semua persyaratan waktu yang didefinisikan oleh spesifikasi *Bluetooth 4.0 Core*. Oleh karenanya, layer ini biasanya diisolasi dari *layer* yang lebih tinggi dari *stack protocol* dengan cara menyediakan antarmuka yang menyembunyikan kompleksitas dan real-time persyaratan dari *layer-layer* di atasnya.

L2CAP atau *Logical Link Control and Adoption Protocol* memiliki dua fungsi utama, yaitu sebagai *protocol multiplexer* yang mendapatkan data dari beberapa protokol di atasnya dan kemudian mengemasnya dalam format standar BLE dan sebaliknya. Dalam layer ini juga dilakukan *fragmentation* -jika paket yang didapat dari layer di atasnya lebih dari 27 *bytes* dan kemudian

diteruskan ke layer di bawahnya, dan *recombination* -jika yang diterima dari layer di bawahnya beberapa paket dan kemudian dijadikan satu paket utuh untuk layer di atasnya. Dari sisi pengembang aplikasi perlu diperhatikan, bahwa *data payload 27 bytes* ini adalah *payload* maksimal yang akan dikirimkan melalui BLE. Sementara L2CAP sendiri memerlukan 4 *bytes header*. Oleh karenanya perlu diatur sedemikian rupa sehingga data yang dikirimkan akan efektif untuk dipecah dalam 23 *bytes* (27 *bytes* - 4 *bytes*).

ATT atau *Attributes Protocol* adalah sebuah protokol *client/server stateless* sederhana yang digunakan oleh perangkat BLE. Dalam BLE, setiap perangkat adalah *client*, *server*, atau mungkin gabungan dari keduanya, tidak peduli apakah perangkat itu *master* atau *slave*. *Client* adalah perangkat yang meminta data dari *server*, dan *server* adalah perangkat yang mengirimkan data ke *client*. Setiap server berisi data dalam bentuk *attribute*, yang masing-masing diberi 16-bit *attribute*, *Universally Unique Identifier* atau UUID, sebuah pasangan informasi hak akses, dan beberapa nilai. *Attribute* digunakan untuk menangani pengenalan yang digunakan untuk mengakses nilai *attribute*. UUID menentukan jenis dan sifat data yang terdapat dalam nilai ini. Ketika *client* ingin membaca atau menulis nilai *attribute* dari/ke *server*, *client* akan melakukan *request* membaca atau menulis ke *server*. *Server* akan merespon dengan nilai *attribute* atau *acknowledgement*. Dalam operasi baca, terserah *client* untuk mengurai nilai dan memahami jenis data berdasarkan UUID dari *attribute*. Di sisi lain, selama operasi menulis, klien diharapkan dapat memberikan data yang konsisten dengan jenis *attribute* dan server bebas untuk menolak operasi jika itu tidak terjadi.

SMP atau *Security Manager Protocol* adalah gabungan dari sebuah protokol dan serangkaian algoritma keamanan yang akan menyediakan *stack protocol* kemampuan untuk menghasilkan dan bertukar *security keys*. Protokol ini akan memungkinkan perangkat berkomunikasi secara aman melalui link yang dienkripsi, untuk menjamin kepercayaan identitas perangkat *remote*, dan juga untuk menyembunyikan *Bluetooth Address* jika diperlukan. SMP dibatasi pada dua peran: pertama sebagai *Initiator* -berhubungan dengan LL *master* dan GAP *central*. Kedua, sebagai *responder* -berhubungan dengan LL *slave* dan GAP *peripheral*.

GATT atau *Generic Attribute Profile* dibangun di atas ATT, menambahkan hirarki dan model data abstraksi di atas ATT. Di satu sisi, hal ini dapat dianggap sebagai tulang punggung transfer data BLE karena mendefinisikan bagaimana data diorganisasi dan dipertukarkan antara aplikasi. GATT mendefinisikan objek data generik yang dapat digunakan berulang oleh berbagai profil aplikasi (dikenal sebagai *GATT-based profiles*). GATT menjaga arsitektur *client/server* yang sama pada *layer* ATT, namun data dikemas dalam *services*, yang terdiri dari satu atau lebih karakteristik. Setiap karakteristik dapat dianggap sebagai kesatuan dari sepotong data pengguna bersama dengan metadata (informasi deskripsi tentang nilai yang seperti *properties*, nama pengguna, bagian, dan banyak lagi).

GAP atau *Generic Access Profile* menentukan bagaimana perangkat berinteraksi satu dengan yang lain pada tingkat yang lebih rendah, di luar *stack protocol* yang sebenarnya. GAP dapat dianggap mendefinisikan BLE kontrol lapisan paling atas, mengingat bahwa itu menentukan bagaimana perangkat melakukan kontrol prosedur-prosedur seperti penemuan perangkat, koneksi, pembentukan keamanan, dan lain-lain untuk memastikan interoperabilitas dan untuk memungkinkan pertukaran data berlangsung antara perangkat dari berbagai vendor. GAP menentukan peraturan perangkat dan konsep yang berbeda untuk mengatur dan standarisasi operasi tingkat rendah perangkat, seperti peran dan interaksi, mode operasional dan transisi, prosedur operasional untuk mencapai komunikasi yang konsisten, aspek keamanan -termasuk mode dan prosedur keamanan, dan format data tambahan untuk data *non-protocol*.

2.3. Broadcast Topology

GAP adalah kunci utama yang memungkinkan perangkat BLE untuk beroperasi dengan satu sama lain. *Profile* ini menyediakan kerangka kerja yang setiap implementasi BLE harus diikuti agar perangkat dapat menemukan satu dengan yang lain, seperti *broadcast data*, berkoneksi dengan *secure*, atau melakukan operasi dasar lainnya dalam standar dan dengan cara yang dipahami secara universal.

GAP memiliki 4 buah pilihan tipe peran sebuah perangkat bergabung dalam jaringan BLE: *Broadcaster*- digunakan untuk perangkat yang hanya mengirimkan data yang didistribusikan secara periodik. Secara teori, sebuah perangkat *broadcaster* dapat dilakukan oleh *transmit-only radio*,

namun pada penerapannya tetap dilakukan oleh radio yang memiliki kemampuan dua arah. *Observer* -digunakan untuk perangkat yang hanya menerima data dari perangkat *broadcaster*. Perangkat ini menerima semua paket data yang dikirimkan dan kemudian menterjemahkannya sesuai dengan aturan yang dimilikinya. *Central* -adalah peran perangkat yang menjadi *LL master*. Perangkat ini dapat membangun beberapa sambungan dengan perangkat-perangkat yang lain. Peran *central* ini selalu sebagai inisiator dari sebuah koneksi dan dia menjadi pengatur dari sebuah perangkat untuk terhubung dalam sebuah jaringan BLE, dan. *Peripheral* -berlawanan dengan peran *central*, peran ini berhubungan dengan *LL slave*. Peran ini menggunakan paket yang ditransmisikan agar *central* menemukannya, dan kemudian melakukan koneksi kepadanya.

Setiap perangkat tertentu dapat beroperasi dalam satu atau lebih peran di atas pada satu waktu, dan spesifikasi BLE tidak melakukan pembatasan pada hal ini. Dalam tabel 1 dapat dilihat mode GAP dan prosedur yang berlaku bagi mereka

Tabel 1. Mode dan prosedur yang memungkinkan

MODE	APPLICABLE ROLE	PEER APPLICABLE
BROADCAST	BROADCASTER	OBSERVATION
NON-DISCOVERABLE	PERIPHERAL	N/A
LIMITED DISCOVERABLE	PERIPHERAL	LIMITED & GENERAL DISCOVERY
GENERAL DISCOVERABLE	PERIPHERAL	GENERAL DISCOVERY
NON-CONNECTABLE	PERIPHERAL, BROADCASTER, OBSERVER	N/A
ANY CONNECTABLE	PERIPHERAL	ANY CONNECTION ESTABLISHMENT

Sebaliknya, dalam tabel 2 memperlihatkan mode yang dibutuhkan oleh lawan dalam melakukan prosedur GAP yang ada.

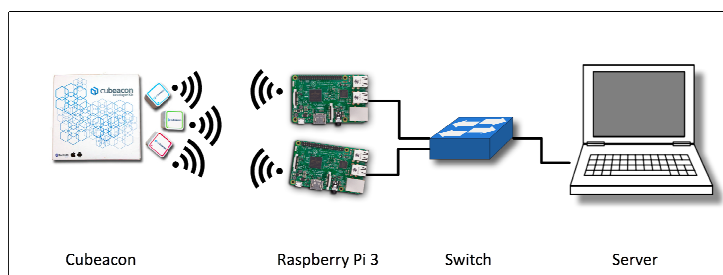
Tabel 2. Prosedur dan mode yang diperlukan

PROCEDURE	APPLICABLE ROLES	APPLICABLE PEER MODE
OBSERVATION	OBSERVER	BROADCAST
LIMITED DISCOVERY	CENTRAL	LIMITED DISCOVERABLE
GENERAL DISCOVERY	CENTRAL	LIMITED & GENERAL DISCOVERABLE
NAME DISCOVERY	PERIPHERAL, CENTRAL	N/A
ANY CONNECTION ESTABLISHMENT	CENTRAL	ANY CONNECTABLE
CONNECTION PARAMETER UPDATE	PERIPHERAL, CENTRAL	N/A
TERMINATE CONNECTION	PERIPHERAL, CENTRAL	N/A

Mode *broadcast* dan prosedur *observation* didefinisikan dalam GAP sehingga menciptakan sebuah kerangka kerja di mana perangkat dapat mengirim paket data secara *uni-direction* -sebagai *broadcaster*, agar satu atau lebih *observer* “mendengarkannya”. Perlu diperhatikan bahwa *broadcaster* tidak memiliki cara untuk mengetahui apakah paket data yang dikirimkan benar-benar mencapai *observer*. Paket data yang dikirim oleh *broadcaster* berisi data untuk pengguna, bersama dengan beberapa item metadata -seperti alamat perangkat *Bluetooth*, yang dimasukkan oleh LL.

Dengan menggunakan perangkat *broadcaster-only* ini, maka data yang ingin di-*sharing*-kan kepada publik disiarkan dan perangkat yang ada dalam jangkauan akan bisa mengambilnya, bisa hanya satu *device* atau ribuan *device*. Sebagai contoh peralatan ini adalah, *Apple iBeacon* atau *Cubeacon* yang menggunakan mode *broadcast* untuk secara konstan mengirimkan data paket tertentu yang memungkinkan perangkat yang ada dalam jangkauan untuk mendeteksi keberadaannya dan menerima informasi data yang dikirimkan. *Beacon* tidak perlu memikirkan berapa banyak *device* yang mendengarkan; dia hanya terus memberitahu dunia bahwa dia ada dan mengirimkan *payload* terbatas untuk siapa pun yang peduli untuk mendengarkan.

3. HASIL DAN PEMBAHASAN



Gambar 5. Topologi Penelitian

Dalam penelitian ini, digunakan 3 buah beacon dari Cubeacon sebagai perangkat *broadcaster-only* dan 2 buah Raspberry Pi 3 yang akan dibuat sebagai *observer* serta sebuah server sebagai penyimpan data yang dibaca oleh *observer* sebelum diproses lebih lanjut. Secara garis besar gambaran topologi jaringannya adalah seperti pada gambar 5.

Dalam penelitian ini isi data yang ditransmisikan Cubeacon tidak dilakukan pengecekan. Identifikasi dari *broadcaster* hanya dilihat dari MAC Address yang dimiliki oleh masing-masing BLE. Sementara pada Raspberry Pi 3, BLE digunakan mode *observer* yang secara terus menerus mendengarkan transmisi BLE yang ada di sekitarnya. Dari penerimaan sinyal BLE ini, didapatkan *Received Signal Strength Indication* atau RSSI -yaitu sebuah nomor yang mengindikasikan *power level* sinyal yang diterima oleh antena Raspberry Pi. Rumus sederhana penghitungan RSSI seperti pada persamaan 1 di bawah:

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

dimana n adalah *path loss exponent* dan untuk 2.4 GHz di *free space* digunakan angka 2. d adalah jarak dalam meter. Sedangkan A adalah referensi *signal strength* (dBm) yang diukur dalam jarak 1 meter.

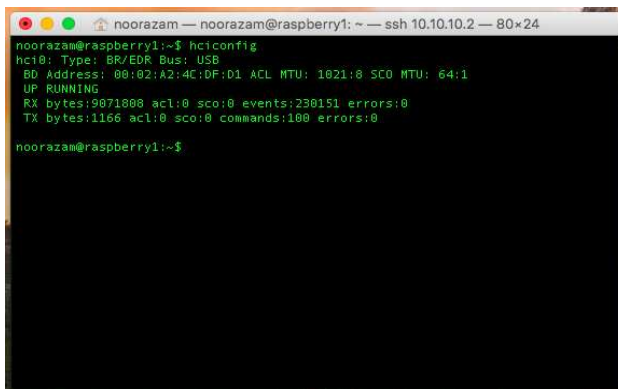
3.1. Instalasi Raspberry Pi

Dalam penelitian ini, digunakan sistem operasi Raspbian Jessie Lite (basis Linux Debian Jessie dengan minimal image) untuk Raspberry Pi 3 yang akan digunakan (https://downloads.raspberrypi.org/raspbian_lite_latest).

```
noorazam@raspberrypi1:~$ ssh 10.10.10.2 - 80x24
2 ls -l
3 passwd
4 ping 10.10.10.1
5 ping www.rad.net.id
6 ping www.narotama.ac.id
7 traceroute 8.8.8.8
8 ls -l /var/log
9 ls -l /cvar/log
10 cd /var/log
11 cat syslog.log
12 dmesg | more
13 sudo apt-get update && sudo apt-get upgrade
14 sudo apt-get update && sudo apt-get upgrade
15 sudo apt-get install libusb-dev
16 sudo apt-get install libglib2.0-dev --fix-missing
17 sudo apt-get install libudev-dev
18 sudo apt-get install libical-dev
19 sudo apt-get install libreadline-dev
20 sudo apt-get install libdbus-glib-1-dev
21 sudo apt-get install bluetooth bluez blueman
22 sudo apt-get install python-bluez
23 clear
24 history
noorazam@raspberrypi1:~$
```

Gambar 6. Beberapa aplikasi dan modul yang perlu diinstall

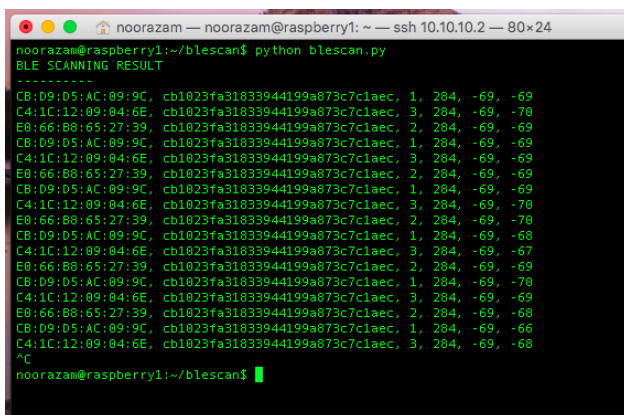
Setelah sistem operasi bisa diinstalasi, pertama kali yang harus dilakukan adalah melakukan *update* agar didapatkan aplikasi terbaru yang ada di *repository server Raspbian*. Setelah itu beberapa fungsi dan modul perlu ditambahkan agar kita bisa melakukan kontrol pada *bluetooth*. Urutan dan fungsi-fungsi yang perlu ditambahkan bisa dilihat pada gambar 6. Setelah semua selesai dilakukan instalasi, maka *bluetooth* sudah bisa dilakukan akses dengan perintah *hciconfig*, seperti pada gambar 7.



Gambar 7. mengakses bluetooth device dengan hciconfig

3.2. Hasil Pengukuran RSSI

Setelah semua persiapan sudah dilakukan, maka dilakukan pengukuran RSSI dari 3 *Cubeacon* pada 2 *Raspberry Pi* yang ada. Gambar 8 adalah contoh *output* aplikasi yang digunakan pada salah satu *Raspberry Pi*.



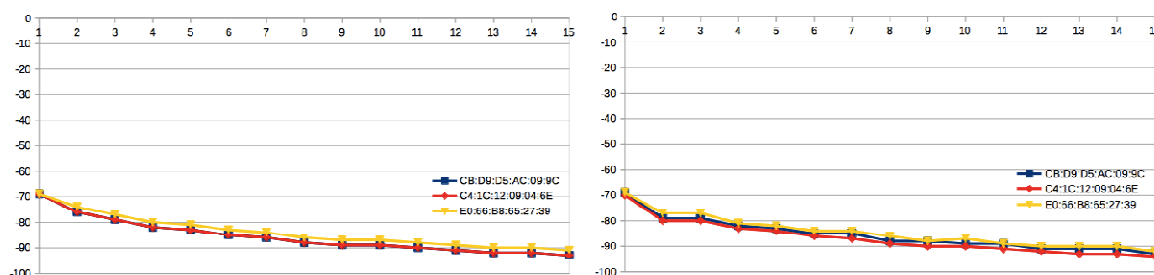
Gambar 8. Pengukuran RSSI

Dari pengamatan sepintas bisa dilihat bahwa RSSI yang dibaca pada *observer* tidak konstan meski jarak dengan *broadcaster* tidak berubah. Oleh karenanya, pengukuran ini dilakukan oleh software setiap 5 detik selama 100 kali pengukuran dan hasilnya kemudian dilakukan rata-rata. Hasil rata-rata RSSI yang terbaca pada masing-masing *Raspberry Pi* untuk setiap *Cubeacon* ada pada tabel 3.

Tabel 3. Hasil rata-rata pengukuran RSSI

d	CB:D9:D5:AC:09:9C		C4:1C:12:09:04:6E		C4:1C:12:09:04:6E	
	RASPI #1	RASPI #2	RASPI #1	RASPI #2	RASPI #1	RASPI #2
1	-69	-69	-69	-70	-67	-67
2	-76	-79	-76	-80	-74	-77
3	-79	-79	-79	-80	-77	-77
4	-82	-82	-82	-83	-80	-81
5	-83	-83	-83	-84	-81	-82
6	-85	-85	-85	-86	-83	-84
7	-86	-85	-86	-87	-84	-84
8	-88	-88	-88	-89	-86	-86
9	-89	-88	-89	-90	-87	-88
10	-89	-89	-89	-90	-87	-87
11	-90	-89	-90	-91	-88	-89
12	-91	-91	-91	-92	-89	-90
13	-92	-91	-92	-93	-90	-90
14	-92	-91	-92	-93	-90	-90
15	-93	-93	-93	-94	-91	-92

Hasil rata-rata RSSI ini bila dibandingkan dalam bentuk grafik, maka akan bisa dilihat seperti pada gambar 9. Dari dua gambar grafik tersebut bisa dilihat bahwa adanya fluktuasi nilai RSSI yang dibaca juga terjadi pada saat pembacaan nilai RSSI oleh masing-masing *observer*. Jadi meski satu *broadcaster* yang sama dengan jarak yang sama namun dibaca oleh dua *observer* yang berbeda, akan menghasilkan nilai yang mungkin berbeda.



Perbedaan ini sangat masuk akal karena nilai RSSI sangat dipengaruhi oleh lingkungan, baik itu lingkungan eksternal radio -misalkan adanya halangan, maupun lingkungan internal radio, misalkan arah antenna.

4. KESIMPULAN

Penggunaan *Raspberry Pi 3* sebagai *observer* dalam jaringan BLE sangat dimungkinkan dan dengan mengukur tingkat RSSI dari beberapa *observer* maka dapat dihitung jarak *broadcaster*. Namun pengukuran RSSI ini memiliki fluktuasi yang harus dipertimbangkan dalam perhitungan.

Untuk dapat menentukan posisi lebih *valid*, maka diperlukan beberapa *observer* yang bisa menerima sinyal dari *broadcaster* yang sama. Dengan menggunakan perhitungan *triangular* atau menghitung irisan di jarak masing-masing *observer*, kesalahan bisa lebih dikoreksi.

Penggunaan *broadcaster* dan *observer* dari vendor yang sama sangat disarankan, agar RSSI yang dihasilkan lebih seragam dan akan membuat perhitungan lebih mendekati kondisi lapangan.

Pengukuran dengan menggunakan *MAC Address* sebagai acuan akan membebani *observer*. Karena itu disarankan dilakukan *filtering* dengan memanfaatkan *UUID* dan *stack protocol* BLE sebelum informasi keberadaan *broadcaster* dilaporkan ke layer aplikasi.

DAFTAR PUSTAKA

- Anonimus. *Calculate Distance from RSSI - Electrical Engineering Stack Exchange.*, <http://electronics.stackexchange.com/questions/83354/calculate-distance-from-rssi>. Diakses: 10 Agustus 2016 jam 10:27.
- Sharhan, S.M.H., and Zickau, S., (2015). *Indoor mapping for location-based policy tooling using Bluetooth Low Energy beacons*. IEEE, doi:10.1109/WiMOB.2015.7347937, pp. 28–36.
- SwitchDoc Labs, *iBeacon Raspberry Pi Scanner in Python.*, <http://www.switchdoc.com/2014/08/ibeacon-raspberry-pi-scanner-python/>. Diakses: 13 Agustus 2016 jam 13:14.
- Tabata, R., Hayashi, A., Tokunaga, S., Saiki, S., Nakamura, M., and Matsumoto, S., (2016). *Implementation and evaluation of BLE proximity detection mechanism for Pass-by Framework*. IEEE, doi:10.1109/ICIS.2016.7550872. pp. 1–6.
- Townsend, K., Davidson, R., Akiba, Cufí, C., (2014), *Getting started with Bluetooth low energy: tools and techniques for low-power networking*, Revised First Edition. ed. O'Reilly, Sebastopol, CA., pp. 9-73.
- Vermesan, Ovidiu., and Friess, Peter., (2014,) *Internet of Things: From Research and Innovation to Market Deployment*, River Publishers, Aalborg, pp. 8-112.
- Xu, Jiuqiang., Liu, Wei., Lang, Fenggao., Zhang, Yuanyuan., and Wang, Chenglong., (2010), *Distance Measurement Model Based on RSSI in WSN*. *Wireless Sensor Network 2*, no. 8. doi:10.4236/wsn.2010.28072. pp. 606-611.