

Implementasi Sistem *Pervasive* Pada *Smart Home* Berbasis *Bluetooth* Versi 4.0 Menggunakan Modul *BLE HM-10* dan Sensor

I Wayan Bobby Astagina Naghi¹, Sabriansyah Rizqika Akbar², Barlian Henryranu Prasetio³

Program Studi Teknik Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹wayannaghi@gmail.com, ²sabrian@ub.ac.id, ³barlian@ub.ac.id

Abstrak

Smart Home dengan konsep *pervasive* mampu menemukan perangkat *Bluetooth* dan layanan yang ditawarkan. *Smart Home* dirancang dapat mengontrol banyak *node* sehingga diperlukan penghematan energi. Maka dari itu, *Bluetooth* yang tepat untuk *Smart Home* berkonsep *pervasive* ini adalah *Bluetooth* 4.0 (*BLE*). *Bluetooth* 4.0 adalah turunan dari teknologi *Bluetooth* klasik yang berbiaya murah, usia baterai panjang karena minimnya konsumsi energi dan pengembangannya yang mudah. Penulis melakukan implementasi sistem *Smart Home* berkonsep *pervasive* untuk dengan *smartphone* sebagai pengontrol melalui media komunikasi *BLE*. *Smart Home* terdiri atas *node* lampu (*Arduino Nano*, LED, dan modul *BLE HM-10*), *node* suhu dan kelembaban (*Arduino Nano*, sensor *DHT11*, dan modul *BLE HM-10*), dan *node* kipas (*Arduino Uno*, kipas DC, dan modul *BLE HM-10*). Aplikasi *Android* untuk pengontrol dirancang menggunakan *MIT App Inventor 2*. *Scanning* dilakukan oleh aplikasi lalu dilanjutkan dengan memilih *node BLE HM-10* dan menghubungkannya melalui tombol *CONNECT*. Saat sudah terjadi koneksi, tombol *SERVICES* ditekan untuk mengetahui layanan yang akan diberikan oleh *node BLE HM-10* tersebut. Jika layanan sudah ditampilkan pada *smartphone*, pengguna dapat memasukkan perintah angka untuk mengoperasikan *node BLE HM-10*. *Feedback* hasil operasi akan ditampilkan pada *smartphone*. Tombol *DISCONNECT* ditekan untuk memutuskan koneksi. Performa *BLE HM-10* juga diuji dalam hal kualitas sinyal dan perbandingan konsumsi energi dengan *Bluetooth HC-05*. Dari hasil pengujian, karena berkonsep *pervasive*, saat proses *scanning* aplikasi dapat mengenali nama-nama *node BLE HM-10* dan layanan dapat ditampilkan ketika pengguna meminta. *Feedback* dan layanan ditampilkan dalam bentuk kalimat dengan jumlah maksimal 20 karakter karena maksimum ukuran *payload BLE HM-10* adalah 20 *byte*, dan kalimat yang melebihi 20 karakter akan ditampilkan bertahap. *BLE HM-10* memiliki kualitas sinyal maksimal 98% dan minimal 20% serta juga unggul dalam segi penghematan energi (5,69 mW) dibanding *Bluetooth HC-05* (13,56 mW).

Kata kunci: *smart home*, *pervasive*, *bluetooth* versi 4.0 (*BLE*)

Abstract

Smart Home with pervasive concept can find Bluetooth device and the services. Smart home constructed can control many node so energy saving is required. Thus, appropriate Bluetooth for this pervasive Smart Home is Bluetooth 4.0 (BLE). Bluetooth 4.0 is derivated from Classic Bluetooth technology which low-cost, long-life battery caused by minimum energy consumption, and easy development. Author implements the pervasive smart home system with smartphone as the controller through BLE communication. Smart Home consists of lamp node (Arduino Nano, LED, and modul BLE HM-10), temperature and humidity node (Arduino Nano, sensor DHT11, and modul BLE HM-10), and fan node (Arduino Uno, kipas DC, dan modul BLE HM-10). Android application as the controller constructed using MIT App Inventor 2. Scanning done by application then continued by choosing BLE HM-10 node and connect them through CONNECT button. When connection formed, SERVICES button pushed to knowing the services given by BLE HM-10 node. If the service has been displayed on smartphone, user can input number instruction to operating the BLE HM-10 node. Feedback of operation result will be displayed on the smartphone. DISCONNECT button pushed to disconnect the connection. BLE HM-10 performance also tested in case of signal quality and energy consumption comparison with Bluetooth HC-05. Based on the test result, caused of pervasive concept, when scanning process, application can

recognize BLE HM-10 node and the services displayed when requested by user. Feedback and service displayed in the form of sentence with maximum amount of 20 characters because maximum payload size of BLE HM-10 is 20 byte, and the sentence more than 20 characters will be displayed one by one. BLE HM-10 has maximum signal quality of 98% and minimum quality of 20% and better in term of energy saving (5,69 mW) than Bluetooth HC-05 (13,56 mW).

Key words: smart home, pervasive, bluetooth version 4.0 (BLE)

1. PENDAHULUAN

Kemajuan pesat di bidang teknologi jaringan kini tidak hanya digunakan sebatas pada media komunikasi antar manusia. *Smart Home* merupakan salah satu contoh hasil pesatnya perkembangan teknologi jaringan yang dapat didefinisikan sebagai konsentrator atau penyebar informasi maupun layanan yang bertujuan untuk mencakup seluruh area fungsional rumah. Fungsi operasional tidak hanya untuk elemen-elemen tertentu saja yang digunakan di dalam rumah untuk meningkatkan level kenyamanan dan kualitas tapi dapat juga menyediakan antarmuka eksterior seperti interaksi dengan *Smart Grid* dan *Smart City* (D. P. Mendes, et al., 2015).

Smart Home yang menjadi primadona di masa kini dapat diimplementasikan dengan konsep komputasi *pervasive*. Komputasi *pervasive* atau disebut juga komputasi dimana-mana bertujuan membuat manusia dapat terhubung dengan komputer tanpa harus menyadari bahwa mereka sedang melakukan komputasi. Komputasi *pervasive* membuat akses dan pemrosesan informasi menjadi lebih tersedia dan mudah didapatkan oleh setiap orang baik dimanapun dan kapanpun. Pengguna dapat bertukar maupun menerima informasi yang mereka inginkan dengan cepat, efisien, tidak perlu usaha yang banyak, tidak peduli dimana mereka berada (Hansmann, et al., 2003).

Fitur *pervasive* dapat diaplikasikan pada *Smart Home* yang menggunakan media komunikasi *Bluetooth* dimana fitur ini mampu menemukan perangkat-perangkat *Bluetooth* yang terdeteksi dan layanan apa yang diberikan oleh perangkat *Bluetooth* tersebut (Munoz-Organero, et al., 2012). *Smart Home* dirancang untuk dapat mengontrol banyak *node* sehingga diperlukan energi yang besar. Maka dari itu, *Bluetooth* yang tepat digunakan sebagai *Smart Home* berkonsep *pervasive* ini adalah *Bluetooth* versi 4.0 atau yang biasa disebut dengan *Bluetooth Low Energy (BLE)*. *BLE* adalah teknologi nirkabel

turunan dari *Bluetooth* klasik dimana ia bekerja pada *bandwidth* 2.4 GHz dan teknologi *Frequency Hopping*. Teknologi ini kompatibel pada banyak perangkat komunikasi seperti *smartphone* dan tablet. *Bluetooth Low Energy* banyak digunakan dalam berbagai bidang seperti kesehatan, olahraga, dan hiburan karena sifatnya yang berbiaya murah, usia baterai yang panjang karena konsumsi arus listrik yang minim dan pengembangannya yang mudah (Rouse, 2014). Dengan demikian, fitur *pervasive* pada *Smart Home* ini akan memudahkan pengguna karena pengguna sudah mengetahui layanan apa yang diberikan sehingga pengguna dapat mengoperasikan perangkat-perangkat *Smart Home* tersebut.

Pada penelitian ini, penulis melakukan implementasi sistem *Smart Home* untuk melakukan 3 pekerjaan yaitu menyalakan maupun mematikan lampu dan mengetahui suhu dan kelembaban dalam rumah, serta menyalakan dan mematikan kipas. 1 buah *smartphone* dengan aplikasinya sebagai pengontrol akan menerima layanan dari *node BLE HM-10* ketika sudah terkoneksi. Dan aplikasi pada *smartphone* dapat mengirim perintah kepada *node BLE HM-10* untuk melakukan 3 pekerjaan diatas. Setelah itu, *smartphone* akan menampilkan data mengenai hasil proses yang telah dilakukan oleh ketiga pekerjaan tersebut. Media koneksi yang digunakan menggunakan teknologi nirkabel yaitu *Bluetooth* versi 4.0 yang dikenal sebagai *Bluetooth Low Energy (BLE)*. Sistem ini bekerja secara *pervasive* dimana masing-masing layanan untuk melakukan 3 pekerjaan tersebut akan dikirim kepada *smartphone* oleh 3 *node BLE HM-10* apabila sudah dalam kondisi terkoneksi. Selain itu, akan dilakukan penelitian terhadap segi kualitas sinyal, jarak jangkauan dan konsumsi energi dari *BLE HM-10* jika dibandingkan dengan media komunikasi nirkabel lain seperti *Bluetooth* klasik *HC-05*.

2. DASAR TEORI

2.1 Bluetooth Low Energy

Pada penelitian tentang protokol Bluetooth v4.0 (*Bluetooth Low Energy*) yang berjudul “*Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology*” dari Carles Gomez, et al., dijelaskan mengenai pengertian *Bluetooth Low Energy*, tujuan, dan manfaat dari protokol ini. *Bluetooth Low Energy (BLE)* adalah teknologi nirkabel hemat energi terbaru yang dikembangkan oleh *Special Interest Group (SIG)* untuk kontrol jarak dekat dan digunakan dalam perangkat monitoring aplikasi yang diharapkan dapat diintegrasikan ke milyaran perangkat di masa depan. (Gomez, et al., 2012)

Seperti teknologi *Bluetooth* klasik, stack protokol pada *BLE* terdiri atas 2 bagian utama, yaitu *Controller* dan *Host*. Pada *Controller* terdapat *Physical Layer* dan *Link Layer*, dan diimplementasikan sebagai *System-on-Chip (SoC)* dengan sebuah radio terintegrasi. *Host* berjalan pada sebuah prosesor dan terdapat beberapa layer fungsional seperti *Logical Link Control and Adaptation Protocol (L2CAP)*, *Attribute Protocol (ATT)*, *Generic Attribute Profile (GATT)*, *Security Manager Protocol (SMP)*, dan *Generic Access Profile (GAP)*. Komunikasi antara *Host* dan *Controller* distandarisasi sebagai *Host Controller Interface (HCI)*.

Saat teknologi *BLE* hadir, teknologi nirkabel hemat energi lainnya seperti *ZigBee*, *6LoWPAN*, atau *Z-Wave*, telah lebih dulu hadir pada beberapa segmen pasar. Akan tetapi, mereka tidak punya peluang yang tinggi untuk dikembangkan pada perangkat-perangkat seperti ponsel pintar. Sehingga *BLE* memiliki peluang yang kuat untuk mengatasi kelemahan tersebut. Berikut ini adalah karakteristik *BLE* dibandingkan dengan teknologi-teknologi nirkabel lainnya.

BLE sangat berpeluang menjadikan sebuah perangkat ponsel pintar sebagai *remote control* universal untuk perangkat-perangkat domestik di rumah. Selain itu, *BLE* dapat digunakan sebagai aplikasi yang *contactless*, contohnya *mobile payment*, pertiketan, dan kontrol akses. *BLE* pun sangat memadai bila diimplementasikan pada lingkungan industri, dimana *BLE* yang merupakan turunan *Bluetooth* klasik, beroperasi pada *adaptive frequency hopping* yang dapat mengatasi *multipath fading*

dan interferensi radio pada mesin. (Gomez, et al., 2012)

2.2 Pervasive

Pada penelitian tentang komputasi *pervasive* yang berjudul “*Pervasive Computing: Vision and Challenges*” dari M. Satyanarayanan, dijelaskan mengenai pengertian dari komputasi *pervasive* dimana komputasi *pervasive* adalah suatu lingkungan komunikasi dan komputasi yang terintegrasi dengan pengguna sehingga seolah-olah teknologi tersebut menjadi “tidak tampak”. Secara spesifik, komputasi *pervasive* menggabungkan 4 unsur penelitian antara lain *effective use of smart spaces*, *invisibility*, *localized scalability*, dan *masking uneven connection*.

Pada penelitian tentang protokol *pervasive* yang berjudul “*Pervasive Device and Service Discovery Protocol in Xbee Sensor Network*” dari Sabriansyah Rizqika Akbar, et al, dijelaskan bahwa ada beberapa protokol komunikasi seperti *Zigbee*, *Xbee*, *6LoWPAN*, dan lain-lainnya yang dapat melakukan fitur *pervasive* pada *layer network* dan *link* seperti *neighbour discovery*, *dynamic addressing*, *routing*, dan *collision advance* (Akbar, et al., 2016).

2.3 Modul BLE HM-10

Modul ini bekerja pada voltase 3.3 V atau 5 V. *Board HM-10* memiliki voltase yang terintegrasi (*DC-DC*) dan *Logic Level Converters (LLC)* sehingga pin-pinnya dapat langsung dihubungkan ke *Arduino*.

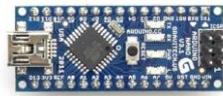


Gambar 2.1 Modul BLE HM-10

HM-10 mengimplementasikan koneksi serial pada pin 1 (*TXD*) dan pin 2 (*RXD*) yang terhubung secara logika ke layanan dan koneksi *BLE*. Setiap data yang diterima melalui pin *RXD* dikirim melalui notifikasi ke perangkat pusat. Setiap data yang ditulis oleh perangkat pusat dilewatkan melalui pin *TXD*. Mekanisme ini menjadikan koneksi *BLE* sebagai koneksi serial standar untuk mikrokontroler yang terkoneksi (*Arduino*, *Raspberry Pi*).

2.4 Arduino Nano dan Arduino Uno

Arduino Nano adalah salah satu varian mikrokontroler *Arduino* yang kecil dengan berat sekitar 5 gram, lengkap, dan *breadboard-friendly* yang berbasis *Atmega328* (*Arduino Nano 3.x*) atau *Atmega168* (*Arduino Nano 2.x*). *Arduino Nano* beroperasi dengan voltase 5 Volt, dengan *input* yang direkomendasikan yaitu sekitar 7-12 Volt. *Nano* memiliki 14 pin digital I/O (6 untuk *PWM*) dan pin analog sebanyak 8. Memori yang digunakan yaitu *FlashMemory* 16 KB untuk *Atmega168* dan 32 KB untuk *Atmega328*, *SRAM* 1 KB untuk *Atmega168* dan 2 KB untuk *Atmega328*, dan kecepatan *clock* yaitu 16 MHz.



Gambar 2.2 *Arduino Nano*

Arduino Uno adalah salah satu varian mikrokontroler *Arduino* seberat 25 gram berbasis *Atmega 328p*. Memiliki 14 pin digital untuk *input/output* (6 untuk *output PWM*), 6 *input* analog, sebuah kristal 16 MHz, koneksi *USB*, *power jack*, *header ICSP*, dan tombol *reset*. Beroperasi pada voltase 5 V, dengan *input* yang direkomendasikan yaitu 7-12 V.



Gambar 2.3 *Arduino Uno*

2.5 Sensor DHT11

Sensor *DHT11* terdiri dari elemen *polimer kapasitif* (digunakan untuk mengukur kelembaban), sensor suhu. Didalamnya juga terdapat memori kalibrasi yang digunakan untuk menyimpan koefisien kalibrasi hasil pengukuran sensor. Data hasil pengukuran dari *DHT11* ini berupa *digital logic* yang diakses secara serial. *DHT11* merupakan sensor digital untuk suhu dan kelembaban sekaligus yang memiliki kisaran pengukuran dari 20-90% RH dan 0-50 derajat celsius. Sensor ini bekerja dengan 2 kabel (data dan *SCK*) (Mandarani, 2014). Data yang diperoleh berupa data pengukuran suhu

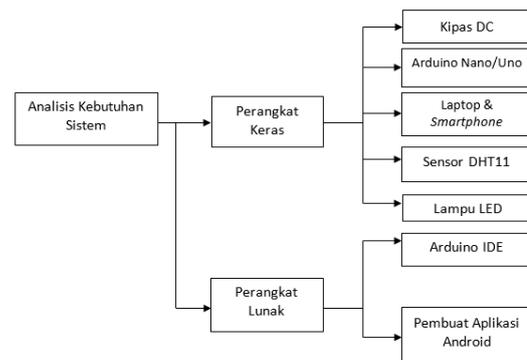
dari lingkungan, jika sensor membaca suhu makin rendah maka tegangan *pull down* yang di alirkan menjadi lebih besar, sehingga akan menghasilkan VCC data yang semakin besar. Data yang dihasilkan dari sensor ini sudah berupa data digital.



Gambar 2.4 Sensor *DHT11*

3. PERANCANGAN DAN IMPLEMENTASI

Perancangan dilakukan pada 2 bagian, yakni perancangan pada sisi aplikasi *Android*, dan perancangan pada sisi *appliances* atau perangkat keras *node BLE HM-10*.



Gambar 3.1 Perancangan Analisis Kebutuhan Sistem

Diagram alir dibawah ini dirancang untuk menjelaskan bagaimana alur pengoperasian sistem.

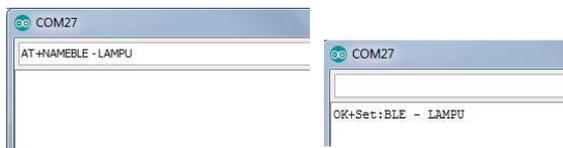
3.1. Perancangan Aplikasi

Aplikasi yang digunakan sebagai pengontrol dalam sistem ini dibuat menggunakan *MIT App Inventor 2*. Aplikasi dirancang untuk memenuhi kebutuhan fungsional antara lain mampu melakukan *scanning node BLE HM-10*, melihat daftar *node BLE HM-10*, memilih *node BLE HM-10*, melakukan *connect* dengan *node BLE HM-10*, menampilkan layanan dari *node BLE HM-10*, mengirim perintah menuju *node BLE HM-10*, menampilkan *feedback* dari *node BLE HM-10*, melakukan *disconnect* dari *node BLE HM-10*. Berikut adalah rancangan aplikasi pada *MIT App Inventor 2*.

3.2. Perancangan Node BLE HM-10

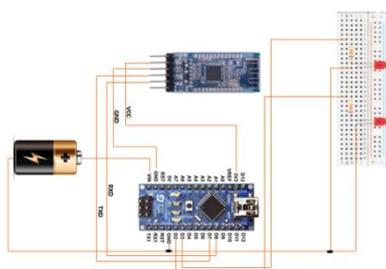
Pada perancangan *node*, 3 buah *node BLE HM-10* yang dirancang antara lain *node* lampu, *node*

suhu dan kelembaban, dan *node* kipas. Terlebih dahulu dilakukan pemberian nama modul *BLE HM-10* agar pengguna dapat mengenali *node* yang ingin digunakan. Pemberian nama dilakukan menggunakan *AT Command* pada *Serial Monitor Arduino* dan *USB-to-TTL*.



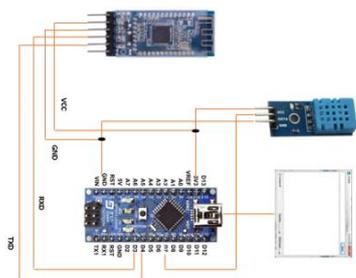
Gambar 3.2 Pemberian Nama *Node BLE HM-10* Menggunakan *AT Command*

Node lampu dibawah ini dirancang menggunakan *Arduino Nano*, lampu LED, dan modul *BLE HM-10*.



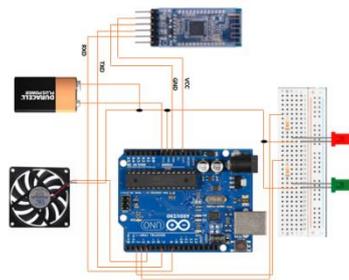
Gambar 3.3 Rancangan *Node BLE HM-10* Lampu

Node suhu dan kelembaban dirancang menggunakan *Arduino Nano*, sensor *DHT11*, dan modul *BLE HM-10*



Gambar 3.4 Rancangan *Node BLE HM-10* Suhu dan Kelembaban

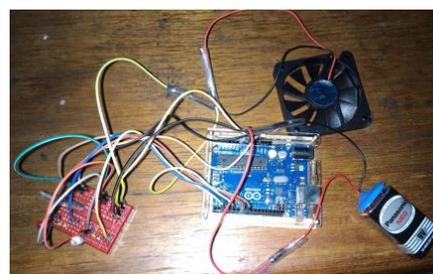
Node kipas dirancang menggunakan *Arduino Uno*, kipas *DC*, lampu *LED*, dan modul *BLE HM-10*.



Gambar 3.5 Rancangan *Node BLE HM-10* Kipas

3.3. Implementasi Perangkat Keras

Implementasi perangkat keras dilakukan pada 3 *node BLE HM-10*, yaitu *node* lampu, *node* suhu dan kelembaban, dan *node* kipas. Nama-nama dari *BLE HM-10* untuk masing-masing *node* diatur memakai *AT-Command* menggunakan perangkat kabel *USB-to-TTL* agar nama tersebut dapat dikenali oleh pengguna dan aplikasi.



Gambar 3.6 Implementasi *Node BLE HM-10*

3.4. Implementasi Perangkat Lunak

Implementasi perangkat lunak dilakukan menggunakan 2 perangkat lunak, yaitu *Arduino IDE* dan *MIT App Inventor 2*. *Arduino IDE* menyediakan *source code* berbahasa *C* yang digunakan untuk melakukan pemrograman pada *appliances*, yaitu 3 *node BLE HM-10* tersebut.



Gambar 3.7 *Arduino IDE*

Sedangkan aplikasi diimplementasikan menggunakan *web-based Android Builder* yaitu *MIT App Inventor 2*. *MIT App Inventor 2* menyediakan penyusunan antarmuka

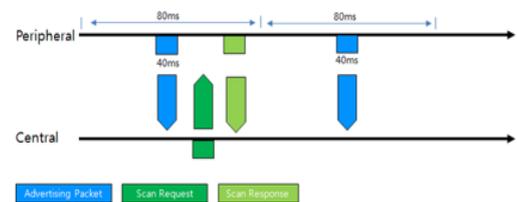
aplikasi dan blok-blok pemrograman yang dapat dirangkai untuk membentuk sebuah aplikasi.



Gambar 3.8 Pembuatan Aplikasi Pengontrol

layanan apa yang diberikan oleh *node BLE HM-10* tersebut hanya jika *smartphone* mereka telah terkoneksi dengan *node BLE HM-10* yang mereka pilih. Aplikasi tidak akan dapat digunakan maupun memberi informasi apa-apa baik layanan maupun *feedback* selama belum terkoneksi dengan *node BLE HM-10*.

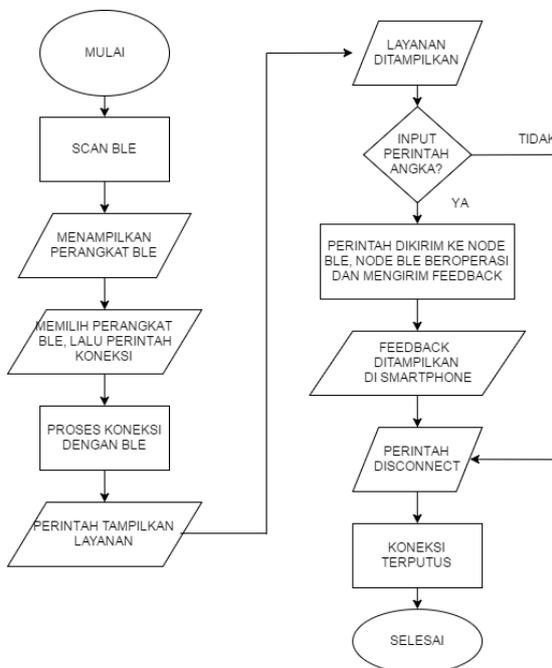
Pengujian pada sisi aplikasi diawali dengan menampilkan *node BLE HM-10* ketika menekan tombol SCANNING. Proses scanning terjadi ketika *smartphone* menerima *advertise* dari *node BLE HM-10*, lalu *smartphone* mengirim *scan request*, dan *smartphone* mengirim balasan berupa *scan response*.



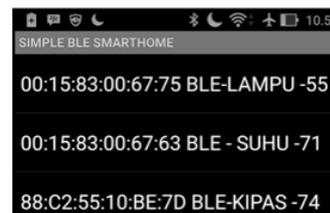
Gambar 4.2 Mekanisme Scanning

4. PENGUJIAN DAN ANALISIS

Diagram alir dibawah ini menjelaskan bagaimana pengguna mengoperasikan sistem.



Gambar 4.1 Diagram Alir Penggunaan Sistem



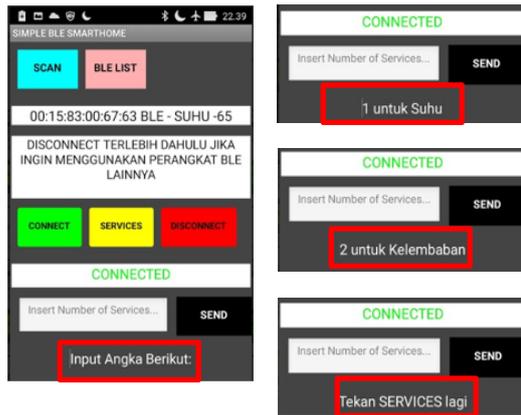
Gambar 4.3 Hasil Scan Node BLE HM-10

Selanjutnya, setelah menampilkan nama-nama *node BLE HM-10* hasil *scanning*, nama *node BLE HM-10* tersebut dapat dipilih dan akan ditampilkan pada aplikasi. Jika menekan tombol CONNECT, maka tulisan CONNECTED akan muncul yang menandakan bahwa *node BLE HM-10* sudah terkoneksi.

Lalu, pengguna dapat mengetahui layanan yang diberikan dengan menekan tombol SERVICES. Layanan akan dikirimkan dan ditampilkan pada bagian paling bawah aplikasi. Layanan ditampilkan dalam bentuk kalimat, dan ditampilkan bertahap per 20 karakter karena batas maksimum *payload* atau data yang dikirim melalui modul *BLE HM-10* adalah 20 karakter atau 20 *byte* per sekali transmisi. Setiap kalimat layanan yang dikirim diberi jeda *delay* selama 2 detik.

4.1. Pengujian di Sisi Aplikasi

Sistem ini berkonsep *pervasive* pada bagian layanan, yaitu pengguna dapat menemukan *node BLE HM-10* yang ingin dicari dan mengetahui



Gambar 4.4 Contoh Layanan yang Ditampilkan

Layanan berupa kalimat yang memerintahkan pengguna untuk memasukkan angka sesuai dengan pekerjaan yang pengguna inginkan untuk dilakukan oleh *node BLE HM-10* tersebut. Dibawah ini adalah contoh menggunakan *node BLE HM-10* suhu dan kelembaban.

Setelah pengguna mengetahui layanan, angka dapat dimasukkan sesuai dengan layanan tersebut pada kotak yang telah disediakan. Angka ini bertipe data string dan akan dikirim menuju *node BLE HM-10* tersebut. Ketika tombol SEND ditekan, maka proses *write* akan dilakukan terhadap *UUID Services* dan *UUID Characteristics* yang dimiliki oleh modul *BLE HM-10*, dan *value* atau angka perintah yang dimasukkan oleh pengguna.



Gambar 4.5 Contoh Input Perintah Angka

Setelah angka dikirim, *node BLE HM-10* akan melaksanakan tugasnya, lalu mengirim *feedback* hasil operasi *node* tersebut. *Feedback* ditampilkan pada sisi paling bawah aplikasi. Jika ingin berhenti menggunakan *node BLE HM-10*, pengguna dapat menekan tombol DISCONNECT untuk memutuskan koneksi.

Kata DISCONNECTED akan muncul untuk menandakan bahwa koneksi telah terputus.



Gambar 4.6 Contoh Hasil Feedback yang Ditampilkan



Gambar 4.7 Contoh Tampilan Disconnect

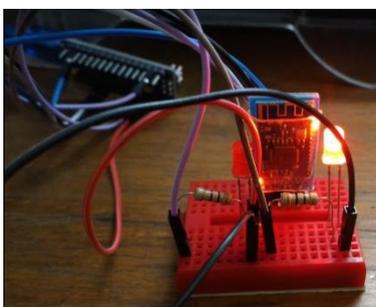
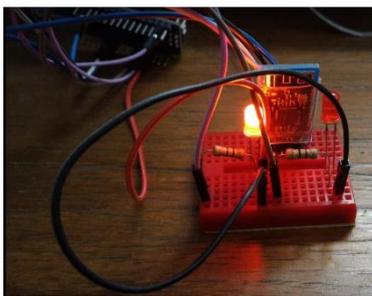
4.2. Pengujian di Sisi Appliances (Node)

Pengujian dilakukan dengan melihat kondisi dari *node BLE HM-10* dimulai sejak aktif hingga dioperasikan oleh aplikasi.

Ketika *node BLE HM-10* aktif, maka paket *advertising* akan segera dikirimkan ke *smartphone*. Dan ketika *smartphone* melakukan *scanning*, nama *node BLE HM-10* tersebut akan muncul pada aplikasi. Ketika sudah terkoneksi, lampu indikator pada modul *BLE HM-10* yang awalnya berkedip-kedip akan berhenti berkedip dan menyala konstan. Saat *node BLE HM-10* menerima perintah untuk mengirim layanan menuju aplikasi, maka *node BLE HM-10* akan mengirim kalimat layanan tersebut secara bertahap. Ini disebabkan karena maksimal *data/payload* yang dapat dimuat oleh paket data

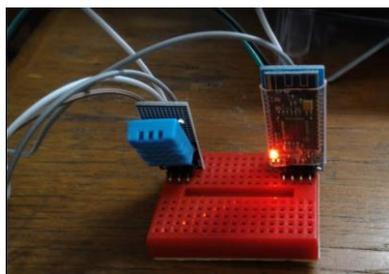
BLE HM-10 per sekali transmisi adalah 20 byte (20 karakter). Sehingga per tahap akan ditampilkan tidak lebih dari 20 karakter layanan pada aplikasi.

Ketika pengguna mengirim perintah berupa angka kepada node BLE HM-10, angka tersebut akan disesuaikan dengan program apa yang akan dikerjakan oleh node BLE HM-10. Selain itu, node BLE HM-10 akan mengirimkan feedback hasil operasi dan ditampilkan pada aplikasi tersebut. Feedback dikirim menggunakan fungsi write dan kalimat feedback yang dikirim tidak lebih dari 20 karakter, sesuai muatan maksimum yang dapat dikirim oleh paket data BLE HM-10. Pada sistem ini, untuk node lampu dapat menerima perintah yaitu 1 (LED A ON), 2 (LED B ON), 3 (LED A OFF), 4 (LED B OFF), 5 (LED A DAN B ON), dan 6 (LED A DAN B OFF).



Gambar 4.8 Contoh Hasil Operasi pada Node BLE HM-10 Lampu

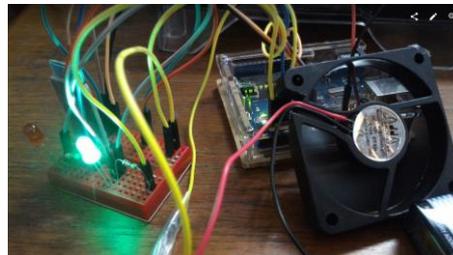
Node suhu dan kelembaban dapat menerima perintah 1 untuk menampilkan suhu, dan 2 untuk menampilkan kelembaban.



Gambar 4.9 Node Suhu dan Kelembaban

Sedangkan, node kipas dapat menerima perintah 1 untuk menampilkan suhu, dan 2 untuk

menampilkan kelembaban. Node kipas dapat menerima perintah 1 (KIPAS ON) dan 2 (KIPAS OFF). Berikut adalah contoh ketika kipas menyala saat menerima perintah 1.



Gambar 4.10 Contoh Hasil Operasi pada Node BLE HM-10 Kipas

Setelah selesai menggunakan, maka dapat dilakukan pemutusan koneksi terhadap node BLE HM-10.

4.3. Pengujian Kualitas Sinyal

Pengujian dilakukan untuk mengetahui kualitas sinyal BLE HM-10 berdasarkan jarak antara node dan smartphone.

Berikut adalah hasil pengujian kualitas sinyal BLE HM-10.

Tabel 4.1 Hasil Pengujian Kualitas Sinyal dan Jarak pada Node BLE HM-10

JARAK (METER)	KEKUATAN SINYAL (dBm)	NODE TERDETEKSI	KUALITAS SINYAL (%)	JARAK (METER)	KEKUATAN SINYAL (dBm)	NODE TERDETEKSI	KUALITAS SINYAL (%)
0	-51	YA	98%	16	-84	YA	32%
1	-51	YA	98%	17	-84	YA	32%
2	-73	YA	54%	18	-85	YA	30%
3	-72	YA	56%	19	-85	YA	30%
4	-74	YA	52%	20	-89	YA	22%
5	-83	YA	34%	21	-88	YA	24%
6	-79	YA	42%	22	-88	YA	24%
7	-76	YA	48%	23	-91	YA	18%
8	-78	YA	44%	24	-87	YA	26%
9	-82	YA	36%	25	-91	YA	18%
10	-78	YA	44%	26	-89	YA	22%
11	-82	YA	36%	27	-87	YA	26%
12	-82	YA	36%	28	-86	YA	28%
13	-83	YA	34%	29	-89	YA	22%
14	-80	YA	40%	30	-90	YA	20%
15	-83	YA	34%	31	-	TIDAK	-

Berdasarkan hasil pengujian, kekuatan sinyal BLE HM-10 agar dapat terdeteksi dan dioperasikan maksimal yaitu 98% pada jarak 0 meter hingga minimal yaitu 20% pada jarak 30 meter. Lebih dari 30 meter node BLE HM-10 sudah tidak terdeteksi lagi dan tidak dapat digunakan.

4.4. Pengujian Konsumsi Energi

Pengujian ini dilakukan untuk mengetahui

performa penggunaan energi dalam mengoperasikan *BLE HM-10* dan dibandingkan dengan performa *Bluetooth* Klasik *HC-05*. Pengujian dilakukan sebanyak 15 kali terhadap masing-masing *Bluetooth* dengan mengukur arus dan voltase menggunakan multimeter digital. Arus dan voltase yang digunakan oleh masing-masing *Bluetooth* lalu dihitung daya yang digunakan menggunakan Persamaan (1):

$$\text{Daya (W)} = V \text{ (Volt)} \times I \text{ (Ampere)} \quad (1)$$

Berikut adalah hasil pengujian terhadap performa konsumsi energi pada *BLE HM-10* dan *HC-05*:

Tabel 4.2 Hasil Pengujian Perbandingan Konsumsi Energi *BLE HM-10* vs *Bluetooth HC-05*

Pengujian Ke-	Arus HC-05 (mA)	Voltase HC-05 (V)	Daya HC-05 (mW)	Arus BLE HM-10 (mA)	Voltase BLE HM-10 (V)	Daya BLE HM-10 (mW)
1	2,6	5	13	1,1	4,96	5,456
2	2,7	5,01	13,527	1,1	4,94	5,434
3	2,7	5	13,5	1,2	5	6
4	2,7	5,02	13,554	1,1	5	5,5
5	2,6	5,06	13,156	1,1	4,97	5,467
6	2,5	5,02	12,55	1,1	5,02	5,522
7	2,7	5,04	13,608	1,1	4,98	5,478
8	2,7	5,08	13,716	1,2	4,98	5,976
9	2,6	5,03	13,708	1,1	5,01	5,511
10	2,7	5,23	14,121	1,1	5	5,5
11	2,7	5,22	14,094	1,1	5,15	5,665
12	2,7	5,08	13,716	1,2	5,08	6,096
13	2,7	5,09	13,743	1,1	5,05	5,555
14	2,7	5,07	13,689	1,2	5,04	6,048
15	2,7	5,06	13,662	1,2	5,06	6,072
Rata-rata	2,6667	5,0673	13,56	1,1333	5,016	5,69

Berdasarkan hasil pengujian sebanyak 15 kali terhadap masing-masing *node Bluetooth*, rata-rata *node BLE HM-10* mengonsumsi arus dan tegangan lebih sedikit (1,1333 mA dan 5,016 V) dibandingkan *node Bluetooth HC-05* (2,6667 mA dan 5,0673 V) sehingga daya yang digunakan oleh *node BLE HM-10* lebih hemat (5,69 mW) dibandingkan *node Bluetooth HC-05* (13,56 mW). Dapat disimpulkan, dalam urusan penghematan konsumsi energi, *node BLE HM-10* lebih unggul kira-kira 2,5 kali lipat daripada *node Bluetooth HC-05*.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

1. *Node BLE HM-10* lampu yang dirangkai menggunakan modul *BLE HM-10*, *Arduino Nano*, dan lampu *LED* dapat menyalakan

dan mematikan lampu *LED* sesuai dengan perintah yang diberikan oleh *smartphone*. *Node BLE HM-10* suhu dan kelembaban yang dirangkai menggunakan modul suhu dan kelembaban *DHT11* dan *Arduino Nano* dapat melakukan *sensing* suhu dan kelembaban sesuai dengan perintah yang diberikan oleh *smartphone*. *Node BLE HM-10* kipas yang dirangkai menggunakan kipas *DC*, lampu *LED*, dan *Arduino Uno* dapat menyalakan kipas dan lampu *LED* sesuai dengan perintah yang diberikan oleh *smartphone*.

2. Konsep *pervasive* pada sistem ini diterapkan dengan cara layanan dari *node BLE HM-10* akan ditampilkan pada *smartphone* apabila *node BLE HM-10* tersebut telah terkoneksi dengan *smartphone* melalui media *Bluetooth 4.0 (BLE)*. Nama dari *node BLE HM-10* diberikan melalui *AT Command* menggunakan *USB-to-TTL* agar *node* tersebut dapat dikenali dan dibedakan dengan *node BLE HM-10* lainnya.
3. Aplikasi untuk mengontrol *node-node BLE HM-10* dibuat menggunakan *MIT APP Inventor 2*. Aplikasi dibuat dengan tampilan sedemikian rupa dan dapat melakukan proses *scanning* untuk mengetahui *node-node BLE HM-10* yang aktif, melakukan *connect* dan *disconnect*, menampilkan layanan, dan melakukan proses *write* untuk mengirim perintah berupa *UUID Characteristics*, *UUID Services*, dan *value* yang berbentuk *string* dalam mengoperasikan *node BLE HM-10*. Aplikasi mampu menampilkan layanan yang diberikan oleh *node BLE HM-10* ketika sudah terkoneksi dengan *node BLE HM-10* tersebut. Aplikasi juga dapat menampilkan *feedback* sebagai hasil kerja yang diberikan oleh *node-node BLE HM-10*.
4. *BLE HM-10* dapat terdeteksi dan digunakan apabila memiliki kualitas sinyal paling tinggi yaitu sekitar 98% dan paling rendah yaitu sekitar 20% sesuai pengujian. *BLE HM-10* juga lebih hemat dalam konsumsi arus dan voltase jika dibandingkan dengan media komunikasi nirkabel lain seperti *Bluetooth HC-05* sehingga daya yang digunakan oleh *BLE HM-10* menjadi lebih sedikit (5,69 mW sesuai pengujian)

dibandingkan *Bluetooth HC-05* (13,56 mW). *BLE HM-10* dapat direkomendasikan sebagai media komunikasi nirkabel masa depan dalam upaya mendukung penghematan energi.

5.2. Saran

1. *Feedback* status yang dikirimkan oleh *node BLE HM-10* dan ditampilkan pada aplikasi di *smartphone* akan hilang saat *smartphone* sudah tidak terkoneksi lagi dengan *node BLE HM-10* sehingga perlu adanya penambahan fitur untuk mengetahui status terakhir dari *node BLE HM-10*.
2. Tampilan dan fitur aplikasi dapat dibuat lebih baik lagi dan dapat dilakukan penambahan jumlah *node BLE HM-10* selain 3 *node BLE HM-10* diatas.
3. Pengembangan sistem diharapkan tidak hanya untuk *Smart Home* saja, tetapi untuk aspek-aspek lainnya seperti industri, kesehatan, pemerintahan, dan lain-lain.

6. DAFTAR PUSTAKA

- Akbar, S. R. et al., 2016. *Pervasive Device and Service Discovery Protocol In XBee Sensor Network*.
- Arduino, C., t.thn. *Arduino Nano*. [Online] Available at: <https://www.Arduino.cc/en/Main/ArduinoBoardNano> [Diakses 12 September 2016].
- Arduino, C., t.thn. *Arduino Uno*. [Online] Available at: <https://www.Arduino.cc/en/main/ArduinoBoardUno> [Diakses 12 September 2016].
- D. P. Mendes, T. et al., 2015. *Smart Homes. Smart Home Communication Technologies and Applications: Wireless Protocol Assessment for Home Area Network Resources*.
- Gomez, C., Oller, J. & Paradells, J., 2012. *Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology*, p. 2.
- Hansmann, U., Merk, L., S. Nicklous, M. & Stober, T., 2003. *Pervasive Computing Second Edition*. 2nd penyunt. New York: Springer-Verlag Berlin Heidelberg.
- Iksan, N., Harso Supangkat, S. & Bagus Baskara Nugraha, I. G., 2013. *Membangun Smart Home untuk Efisiensi Energi*.
- Lee, W.-M., 2012. *Beginning Android 4 Application Development*. 1st penyunt. Indianapolis: John Wiley & Sons, Inc..
- Mandarani, P., 2014. *Perancangan dan Implementasi User Interface Berbasis Web untuk Monitoring Suhu, Kelembaban, dan Asap pada Ruangan Berbeda dengan Memanfaatkan Jaringan Local Area Network*, Volume II.
- Montaner, J., 2015. *HM-10 BLE Module*. [Online] Available at: <http://blog.BLEcentral.com/2015/05/05/hm-10-peripheral/> [Diakses 12 September 2016].
- Munoz-Organero, M., J. Munoz-Merino, P. & Delgado Kloos, C., 2012. *Using Bluetooth to Implement a Pervasive Indoor Positioning System with Minimal Requirements at the Application Level*.
- Rouse, M., 2014. *Bluetooth Low Energy (Bluetooth LE)*. [Online] Available at: <http://internetofthingsagenda.techtarget.com/definition/Bluetooth-Low-Energy-Bluetooth-LE> [Diakses 12 September 2016].
- Satyanarayanan, M., 2001. *Pervasive Computing: Vision and Challenges*, p. 1.
- Townsend, K., Cufi, C., Davidson, A. & Davidson, R., 2014. *Getting Started with Bluetooth Low Energy*. 1st penyunt. California: O'Reilly Media.