PENGEMBANGAN *GATEWAY* BERBASIS *EMBEDDED DEVICE* UNTUK INTEROPERABILITAS JARINGAN SENSOR NIRKABEL DAN PROTOKOL INTERNET

SKRIPSI

Disusun oleh:
GUNTUR DHARMA PUTRA
09/284593/TK/35393

JURUSAN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA

2014

HALAMAN PENGESAHAN

PENGEMBANGAN *GATEWAY* BERBASIS *EMBEDDED DEVICE* UNTUK INTEROPERABILITAS JARINGAN SENSOR NIRKABEL DAN PROTOKOL INTERNET

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik Program S-1
Pada Jurusan Teknik Elektro dan Teknologi Informasi Fakultas Teknik
Universitas Gadjah Mada

Disusun oleh:

GUNTUR DHARMA PUTRA
09/284593/TK/35393

Telah disetujui dan disahkan
pada tanggal 3 Februari 2014

Dosen Pembimbing I	Dosen Pembimbing II
Sigit Basuki Wibowo, S.T., M.Eng.	**Bimo Sunarfri Hantono, S.T., M.Eng.**
NIP. 1976 0501 2002 12 1 002	**NIP. 1977 0131 2002 12 1 003**

PERNYATAAN

Dengan ini saya menyatakan bahwa skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 24 Maret 2014

Guntur Dharma Putra

HALAMAN PERSEMBAHAN

Untuk Ibu, Bapak,
dan Adik-adikku tercinta.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur penulis panjatkan ke hadirat Allah SWT karena hanya dengan rahmat dan hidayah-Nya, Tugas Akhir ini dapat terselesaikan tanpa halangan berarti. Keberhasilan dalam menyusun laporan Tugas Akhir ini tidak lepas dari bantuan berbagai pihak yang mana dengan tulus dan ikhlas memberikan masukan guna sempurnanya Tugas Akhir ini. Oleh karena itu dalam kesempatan ini, dengan kerendahan hati penulis mengucapkan terima kasih kepada:

1. Bapak Sarjiya, S.T., M.T., Ph.D., selaku Ketua Jurusan Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada,

2. Bapak Sigit Basuki Wibowo, S.T., M.Eng. selaku dosen pembimbing pertama yang telah memberikan banyak bantuan, bimbingan, serta arahan dalam Tugas Akhir ini,

3. Bapak Bimo Sunarfri Hantono, S.T., M.Eng. selaku dosen pembimbing kedua yang juga telah memberikan banyak bantuan, bimbingan, serta arahan dalam Tugas Akhir dan kegiatan-kegiatan yang lain,

4. Bapak Warsun Najib, S.T., M.Sc. selaku dosen pembimbing akademis penulis dan juga dosen pembimbing lapangan penulis pada KKN-PPM UGM 2013 Unit SLM07,

5. Seluruh Dosen di Jurusan Teknik Elektro dan Teknologi Informasi FT UGM, yang tidak bisa disebutkan satu-satu, atas ilmu dan bimbingannya selama penulis berkuliah di JTETI,

6. Ibu dan Bapak yang selama ini telah sabar membimbing, mengarahkan, dan mendoakan penulis tanpa kenal lelah untuk selama-lamanya,

7. Segenap staf dan karyawan di JTETI FT UGM, yang telah banyak membantu penulis selama belajar di JTETI FT UGM. Khususnya staf bagian pengajaran S1 JTETI, Mas Mery, Mas Tri, dan Mas Nanang; Mbak Ria Ryong; Mas Jeff dan Mbak Lilik Ruang Server yang telah meminjamkan ruang server untuk interview Skype; Mas Rudy, Mas Yaenuri, dan staff TU JTETI; Mas Dedy S2 JTETI atas ilmu-ilmunya,

8. Senior-senior saya Teknik Elektro angkatan 2005, Mas Galang, Mas Giovanni Yoko, dkk; 2006, Mas Bagor, Mas Uday, Mbak Geo, dkk; 2007, Mas Canggih, Mas Nasikun, Mas Kuntul, Mas Odit, Mas Adit, dkk; dan 2008, Mas Qius, Mas Azka, Mas Kuntoro, Mas Andika, Mas Mas Ajay, dkk. atas segala ilmu dan pengalaman yang telah dibagi,
9. Teman-teman di Laboratorium Sistem Frekuensi Tinggi, Mas Yanto, Mas Nanang, dkk atas keramahan dan kehangatannya; Sistem Elektronis, Mas Heru atas share pengalamannya; Jaringan Komputer, atas keramahan, kehangatan, konsumsi, dan logistik walaupun saya member baru; dan Informatika, Pak Muji atas K3-nya,
10. Teman-teman seperjuangan Teknik Elektro konsentrasi SKI angkatan 2009, Damar, Putra, Danu, Karin, Momon, Gilang, Billy, Aldo, Rawamantan (semoga cepat lulus), Amy, Dandy, Nufus, Dedy, Dino, Reza, Age, Firdy, dkk,
11. Junior-junior saya Teknik Elektro dan Teknologi Informasi, mulai dari angkatan 2010 sampai seterusnya yang saya tidak bisa sebutkan satu-satu,
12. Seluruh civitas akademika JTETI, FT, dan UGM,
13. Teman-teman KKN SLM 07 di dusun Gamplong I, II, dan III, Ayu, Fitra, Mala, Syefira, dan Nitiya,
14. Bapak dan Ibu Pondokan Rumah Pintar Mentari, Pak Wid, Bu Tati, Mas Novi, Mba Endang, Mas Fajar, dan Miya, di Dusun Gamplong III, Desa Sumberrahayu, Moyudan, Sleman, D.I.Yogyakarta,
15. dan semua yang saya tidak bisa sebutkan satu-satu pada kata pengantar ini.

Penulis menyadari bahwa penyusunan Tugas Akhir ini jauh dari sempurna. Kritik dan saran dapat ditujukan langsung pada e-mail atau *mention* langsung pada akun *twitter* saya. Akhir kata penulis mohon maaf yang sebesar-besarnya apabila ada kekeliruan di dalam penulisan Tugas Akhir ini.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 15 Januari 2014

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN **i**

HALAMAN PERNYATAAN **i**

HALAMAN PERSEMBAHAN **iv**

KATA PENGANTAR **v**

DAFTAR ISI **ix**

DAFTAR TABEL **x**

DAFTAR GAMBAR **xii**

DAFTAR SINGKATAN **xiii**

Intisari **xv**

Abstract **xvi**

I LATAR BELAKANG **1**

1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 3
1.6 Keaslian Penelitian 3
1.7 Sistematika Penulisan 3

II TINJAUAN PUSTAKA DAN DASAR TEORI **5**

2.1 Tinjauan Pustaka 5
2.2 Landasan Teori 6
2.2.1 *Wireless Sensor Network* 6
2.2.2 IQRF 8
2.2.3 XBee 9
2.2.4 TCP/IP 9

2.2.5 *Access Point* 10
2.2.6 TP-LINK MR3020 10
2.2.7 Web Server 12
2.2.8 *Asynchronous JavaScript and XML* 12
2.2.9 OpenWRT 13
2.2.10 SSHFS 13
2.2.11 Bootstrap 14
2.2.12 Sublime Text 3 14

III METODOLOGI PENELITIAN 16
3.1 Alat dan Bahan 16
3.1.1 Perangkat Keras 16
3.1.2 Perangkat Lunak 16
3.2 Alur Penelitian 17
3.2.1 Pra Penelitian 17
3.2.2 Pengembangan Aplikasi 17
3.2.3 Evaluasi dan Perbaikan 19
3.2.4 Pasca Penelitian 20
3.2.5 Diagram Alir Penelitian 20
3.3 Tahapan Pelaksanaan 21
3.4 Jadwal Kegiatan 22

IV HASIL DAN PEMBAHASAN 23
4.1 Analisis Kebutuhan Sistem 23
4.1.1 Fitur-Fitur Aplikasi 23
4.1.2 *Use Case Diagram* 24
4.1.3 Diagram Arsitektur Sistem 24
4.1.4 *Software Development Life Cycle* (SDLC) 25
4.2 Pengembangan Aplikasi 26
4.2.1 Persiapan Pra Pengembangan Aplikasi 26
4.2.2 Pengembangan Aplikasi WSN 31
4.2.3 Pengembangan Aplikasi Python 32
4.2.4 Pengembangan Aplikasi Berbasis Web 36
4.2.5 Skalabilitas Sistem 40
4.2.6 Evaluasi dan Perbaikan 40
4.2.7 *Screenshot* Aplikasi 40

4.2.8 Kode Sumber 49
4.3 Analisis Unjuk Kerja Aplikasi 49
4.3.1 Instalasi Piranti 50
4.3.2 Hasil Uji Coba Aplikasi 54
4.3.3 Kelebihan dan Kekurangan Sistem 56
4.3.4 Masalah dan Penyelesaian 57

V KESIMPULAN DAN SARAN 58
5.1 Kesimpulan 58
5.2 Saran 58

DAFTAR PUSTAKA 61

DAFTAR TABEL

Tabel 2.1 Fitur perangkat keras dari TP-LINK MR3020. 11
Tabel 2.2 Fitur komunikasi nirkabel dari TP-LINK MR3020. 12

Tabel 3.1 Jadwal Penelitian. 22

DAFTAR GAMBAR

Gambar 2.1 Jaringan sensor nirkabel. 6
Gambar 2.2 Contoh sebuah simpul sensor IQRF. 7
Gambar 2.3 Jaringan bintang menggunakan WiFi. 7
Gambar 2.4 Sepasang piranti XBee. 9
Gambar 2.5 TP-LINK MR3020. 11
Gambar 2.6 Tampilan antarmuka *command-line* OpenWRT versi *BackFire*. 13
Gambar 2.7 Aplikasi Sublime Text 3 yang berjalan pada Mac OS X. . . . 14

Gambar 3.1 Diagram ilustrasi aplikasi yang dibangun. 17
Gambar 3.2 Diagram ilustrasi detil komponen-komponen dari sistem beserta alur komunikasinya. 18
Gambar 3.3 Diagram alir penelitian. 20
Gambar 3.4 Arsitektur WSN dan WiFi dengan sebuah AP. 21

Gambar 4.1 Diagram *use case* dari penelitian. 24
Gambar 4.2 Diagram Arsitektur Sistem. 25
Gambar 4.3 Diagram SDLC. 26
Gambar 4.4 Konfigurasi SSH pada Mac OS X. 30
Gambar 4.5 Manipulasi komunikasi data serial [1]. 33
Gambar 4.6 Diagram alir aplikasi Python untuk IQRF. 34
Gambar 4.7 Diagram alir aplikasi Python untuk menangani piranti XBee. 35
Gambar 4.8 Diagram alir aplikasi Python untuk menyala-matikan lampu pada kondisi tertentu. 37
Gambar 4.9 Peta situs aplikasi web. 38
Gambar 4.10 *Entity Relationship Diagram* dari basis data untuk aplikasi web. 38
Gambar 4.11 Diagram Alir Penambahan Piranti IQRF ke Aplikasi. 39
Gambar 4.12 Diagram Alir Penambahan Piranti XBee ke Aplikasi. 39
Gambar 4.13 Halaman *dashboard* saat belum ada piranti terpasang. 41
Gambar 4.14 Halaman untuk menambahkan piranti IQRF. 42
Gambar 4.15 Halaman untuk menambahkan piranti XBee. 43
Gambar 4.16 Halaman *dashboard* saat piranti sudah terpasang. 44
Gambar 4.17 Halaman daftar piranti IQRF yang terpasang. 45
Gambar 4.18 Halaman daftar piranti XBee yang terpasang. 45

Gambar 4.19 Halaman untuk menambahkan profil. 46
Gambar 4.20 Halaman yang menampilkan daftar pengguna. 47
Gambar 4.21 Halaman untuk menambahkan pengguna baru. 48
Gambar 4.22 Halaman login. 49
Gambar 4.23 Koordinator sensor IQRF yang akan disambungkan pada AP. 50
Gambar 4.24 Koordinator piranti XBee yang akan disambungkan pada AP. 51
Gambar 4.25 AP TP-LINK MR3020. 52
Gambar 4.26 Sensor IQRF. 53
Gambar 4.27 Piranti XBee. 54
Gambar 4.28 Uji coba aplikasi dengan ponsel cerdas. 55
Gambar 4.29 Membaca temperatur yang terbaca pada sensor. 56

DAFTAR SINGKATAN

A

AJAX	Asynchronous JavaScript and XML
AP	Access Point
API	Application Programming Interface

C

CLI	Command Line Interface

C

DFM	Discovered Full Mesh

E

ERD	Entity Relationship Diagram

F

FTDI	Future Technology Devices International
FUSE	Filesystem in Userspace

I

IP	Internet Protocol

J

JTETI	Jurusan Teknik Elektro dan Teknologi Informasi

L

LAN	Local Area Network

O

OSI	Open Systems Interconnection

R

RF	Radio Frequency

S

SDLC	Software Development Life Cycle
SFTP	Secure Shell File Transfer Protocol
SSHFS	Secure Shell Filesystem

U

UGM	Universitas Gadjah Mada
USB	Universal Serial Bus

V

VRS	Virtual Routing Structure

W

WAP	Wireless Access Point
WIT	Western Indonesian Time
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

Intisari

Penggunaan *Wireless Sensor Network* (WSN) untuk gedung dan perumahan kian populer karena dapat dimanfaatkan untuk berbagai kepentingan seperti *home automation* dan *home surveillance*. Oleh karena itu, untuk meningkatkan fleksibilitas penggunaan WSN, diperlukan sistem pengendalian yang dapat dikendalikan secara jarak jauh. Padahal pada umumnya, WSN dikendalikan oleh sebuah pengendali utama yang berada di sekitar tempat WSN itu berada.

Dengan menggunakan metode pengembangan iteratif, yang sesuai untuk pengembangan aplikasi utama dan aplikasi-aplikasi pendukung pada penelitian ini dengan beberapa kali iterasi, penelitian ini mengusulkan interoperabilitas WSN dan *Internet Protocol* (IP) yang memungkinkan WSN untuk dapat dikendalikan dari manapun dan dengan apapun selama masih terhubung dengan jaringan Internet. Pada akhirnya, penelitian ini menghasilkan sebuah aplikasi berbasis web yang diimplementasikan pada sebuah *Access Point* (AP) nirkabel sehingga dapat diakses dengan Wi-Fi secara mudah guna pengendalian WSN secara jarak jauh.

Kata kunci : *wireless sensor network*, *Internet Protocol*, WiFi, interoperabilitas.

Abstract

Wireless Sensor Network (WSN) usage for buildings and household has been increasingly popular because it offers various benefits, such as home automation and home surveilliance. Therefore, in order to increase WSN usage flexibility, remote controlling which enables remote administration is needed. In fact, generally WSN is controlled by a coordinator (sink node) that is located closely to the WSN area itself.

By utilizing iterative development, which is suitable for main and supporting application development with several iteration, this research proposes WSN and Internet Protocol interoperability that enables remote controlling through Internet. Eventually, this research creates a web-based application which will be implemented to a wireless AP for easier WSN remote controlling.

Keywords : *wireless sensor network, Internet Protokol, WiFi, interoperability.*

BAB I

LATAR BELAKANG

1.1 Latar Belakang Masalah

Jaringan sensor nirkabel, atau yang juga dikenal dengan istilah *Wireless Sensor Network* (WSN), adalah jaringan simpul sensor otonom terdistribusi yang dapat berkomunikasi satu sama lain secara nirkabel [1, 2]. WSN tidak hanya digunakan dalam lingkup penelitian tapi juga sudah digunakan serara luas dalam implementasinya di pengamatan lingkungan, fasilitas rumah cerdas, pelacakan alam dan ekologis, pengendalian kualitas dalam teknik industri dan pembangunan, prediksi tanah longsor, dan lain-lain.

Penggunaan WSN untuk sebuah gedung dan perumahan kian populer [3] karena dapat dimanfaatkan untuk berbagai kepentingan. Contoh penerapan WSN dalam rumah yang sangat populer adalah *home automation* yaitu proses automatisasi segala rutinitas yang ada di rumah. Sebagai contoh, sang pemilik rumah harus menyalakan lampu di kala waktu sudah senja dan atau menyalakan pendingin ruangan saat pemilik baru saja pulang dari bekerja. Segala sesuatu yang mungkin untuk diautomatisasi, dapat terealisasi dengan bantuan WSN. Contoh lain penerapan WSN dalam rumah adalah *home surveillance* yaitu pemanfaatan WSN untuk mengawasi tiap sudut rumah secara *realtime*. Dengan ini, sang pemilik rumah tidak perlu lagi khawatir jika rumahnya kurang pengawasan karena mengawasi rumah menjadi semakin mudah dengan bantuan teknologi WSN ini.

Pada umumnya, WSN dikendalikan oleh sebuah *sink node* yang berada dekat pada wilayah jaringan sensornya [4]. Padahal, pengendalian WSN sebagai fasilitas rumah cerdas acap kali memerlukan pendendalian secara jarak jauh karena sang pemilik rumah tidak selalu berada di dalam rumah.

Selain itu, sebuah *sink node*-pun juga memiliki keterbatasan, terlebih dalam hal menangani beberapa WSN yang berasal dari berbagai *vendor*. Sebagai contoh, sebuah *sink node* keluaran IQRF tidak dapat menangani WSN lain selain dari IQRF. Begitu pula *sink node* produk XBee, misalnya, tidak bisa menangani sama sekali sensor-sensor selain produk XBee.

Lebih jauh lagi, protokol yang digunakan dalam WSN adalah protokol yang sifatnya *proprietary* yang sifatnya tertutup dan tidak mendukung interoperabilitas.

Hal ini menyebabkan tiap WSN dari *vendor* yang berbeda tidak dapat saling berkomunikasi.

Protokol Internet menawarkan fleksibilitas dan interoperabilitas WSN dengan piranti lain [5], bahkan dengan piranti selain WSN seperti komputer dan telepon seluler. Namun demikian, protokol Internet boros dalam pengkonsumsian daya listrik. Ada beberapa cara untuk mengintegrasikan IP dengan WSN [6], salah satu caranya adalah dengan membangun *gateway* [7] yang terhubung dengan *sink node* dari WSN.

1.2 Rumusan Masalah

Bagaimana cara membangun sistem pengendalian WSN yang dapat dikendalikan secara jarak jauh dengan mudah dan dapat menginteroperabilitaskan WSN dan protokol Internet. Selain itu, sistem pengendalian yang dibangun juga harus dapat mengintegrasikan beberapa vendor WSN ke dalam sebuah *gateway* yang sama dengan biaya yang rendah.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Penelitian ini difokuskan pada interoperabilitas beberapa *vendor* WSN dan protokol Internet.
2. Tipe WSN yang digunakan dalam penelitian ini dibatasi dua buah.
3. Pengujian yang dilakukan hanya sebatas eksperimen dalam lingkup laboratorium.
4. Purwarupa yang dihasilkan akan diimplementasikan pada sebuah *Access Point* (AP).

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah mempelajari kemungkinan pengembangan perangkat lunak yang akan ditanamkan ke dalam sebuah AP untuk difungsikan sebagai gateway sehingga mampu digunakan untuk mengintegrasikan jaringan WiFi dan beberapa protokol WSN ke jaringan internet dengan biaya yang murah.

1.5 Manfaat Penelitian

Dengan terhubungnya WSN ke jaringan internet dimungkinkan pengembangan aplikasi WSN yang dapat diakses melalui jaringan internet. Terhubungnya WSN ke jaringan internet akan membuka kemungkinan pengembangan layanan-layanan yang lebih beragam terutama layanan yang berbasis IP. Hal ini sejalan dengan perkembangan teknologi komunikasi yang menuju konvergensi penggunaan IP.

Selain itu, pengintegrasian gateway untuk WiFi dan WSN dalam satu piranti juga membuka peluang besar untuk memecahkan persoalan interoperabilitas perangkat keras dari berbagai *vendor* dan kemudahan sistem.

1.6 Keaslian Penelitian

Penelitian ini tidak untuk menguji hipotesis baru melainkan merupakan pengembangan perangkat lunak yang akan ditanamkan ke dalam gateway sehingga mampu menghubungkan jaringan WiFi dan WSN ke jaringan internet. Penelitian ini akan meningkatkan fungsi AP menjadi gateway yang menghubungkan WiFi dan WSN dengan jaringan internet.

1.7 Sistematika Penulisan

BAB I : PENDAHULUAN

Pada bab ini dijelaskan latar belakang, rumusan masalah, batasan, tujuan, manfaat, keaslian penelitian, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA DAN LANDASAN TEORI

Pada bab ini dijelaskan teori-teori dan penelitian terdahulu yang digunakan sebagai acuan dan dasar dalam penelitian.

BAB III : METODOLOGI PENELITIAN

Pada bab ini dijelaskan metode yang digunakan dalam penelitian meliputi langkah kerja, pertanyaan penilitian, alat dan bahan, serta tahapan dan alur penelitian.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan hasil penelitian dan pembahasannya.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini ditulis kesimpulan akhir dari penelitian dan saran untuk pengembangan penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Secara umum, cara untuk menghubungkan *Wireless Sensor Network* (WSN) dengan jaringan internet dapat dikelompokkan menjadi dua [6]. Cara pertama adalah menggunakan *gateway* [7] dan cara yang kedua adalah dengan menggunakan simpul sensor yang sudah dilengkapi dengan protokol internet. Cara yang lebih mudah ditempuh adalah dengan cara yang pertama karena pengubahan yang dilakukan relatif tidak terlalu besar. Sedangkan cara yang kedua akan menemui banyak kendala terutama pada WSN yang sudah terpasang karena harus dilakukan penggantian tiap simpul sensor.

Salah satu usaha untuk mengintegrasikan jaringan WSN dengan jaringan Wi-Fi menggunakan *gateway* misalnya dilakukan pada penelitian yang dilakukan oleh [8]. Pada penelitian tersebut pengintegrasian dilakukan dengan sebuah komputer yang didedikasikan untuk keperluan tertentu. Penggunaan komputer khusus ini adalah *hardware-solution* yang membutuhkan biaya dan kerumitan sistem.

Riset [9] juga menawarkan pengintegrasian dengan jaringan berbasis *Internet Protocol* (IP). Namun demikian di dalam riset ini diperlukan pengubahan yang signifikan jika konfigurasi jaringan sensor nirkabel sudah terpasang. Simpul sensor yang digunakan harus diganti dengan simpul sensor yang mendukung IP. Hal ini jelas akan memakan biaya yang cukup besar dan tidak praktis untuk dilakukan. Terlebih lagi jika jumlah sensor yang telah terpasang jumlahnya cukup banyak.

Penelitian [1] sudah berhasil mengembangkan sebuah *Access Point* (AP) menjadi *gateway* yang dapat digunakan untuk menghubungkan sebuah protokol WSN dengan jaringan IP. Protokol WSN yang digunakan adalah protokol dari IQRF. Penelitian tersebut kemudian dilanjutkan dengan penelitian [10] yang sudah diterapkan dalam sistem *domotic*.

Penelitian ini merupakan kelanjutan dari penelitian yang telah dikembangkan oleh [1] dan [10]. Pada penelitian tersebut, belum ada interoperabilitas berbagai *vendor* ke dalam sebuah AP. Oleh karena itu, penelitian ini berusaha untuk, selain menginteroperabilitaskan WSN dan IP, juga mengintegrasikan dua jenis WSN ke dalam sebuah AP untuk meningkatkan sisi interoperabilitas. Selain itu, penelitian ini

juga mengembangkan antarmuka berbasis web yang responsif, yang akan beradaptasi dengan ukuran layar secara otomatis, sehingga pengendalian WSN dapat dilakukan dari komputer hingga ponsel cerdas.

2.2 Landasan Teori

2.2.1 *Wireless Sensor Network*

Jaringan sensor nirkabel, atau yang juga dikenal dengan *Wireless Sensor Network*, adalah jaringan simpul sensor otonom yang terdistribusi digunakan untuk memonitor kondisi fisik atau lingkungan [11] misalnya suhu, suara, getaran, kelembaban, dan lain-lain. Selain itu, tidak menutup kemungkinan untuk menambahkan fungsi tambahan pada setiap simpul misalnya port masukan/keluaran (I/O port) yang terdapat dalam setiap simpul dihubungkan dengan aktuator sehingga dapat digunakan untuk mengendalikan piranti elektrik atau elektronis.

Secara umum, WSN dapat diilustrasikan seperti Gambar 2.1. Pada gambar tersebut terlihat adanya beberapa simpul yang diwakili dengan titik berukuran kecil dan satu buah simpul yang diwakili dengan titik berukuran lebih besar. Titik yang berukuran kecil mewakili simpul sensor sedangkan titik yang berukuran besar mewakili *gateway* yang berfungsi menghubungkan jaringan sensor nirkabel dengan pengendali utama yang dalam gambar tersebut diwakili oleh sebuah komputer. Contoh sebuah simpul dari IQRF ditunjukkan pada Gambar 2.2.

Gambar 2.1: Jaringan sensor nirkabel.

Gambar 2.2: Contoh sebuah simpul sensor IQRF.

Pada umumnya, WSN adalah jaringan yang berdiri sendiri. Untuk menghubungkan WSN dengan jaringan yang lain misalnya jaringan internet, maka salah satu cara adalah dengan membangun *gateway* WSN yang mampu menjembatani perbedaan protokol yang ada pada WSN dan jaringan internet. Cara tersebut adalah cara yang ditempuh dalam penelitian ini karena lebih mudah dilakukan dibandingkan dengan cara yang lain seperti sudah dijelaskan pada Bab Tinjauan Pustaka.

Sementara itu, jaringan WiFi sebagai jaringan lokal nirkabel yang digunakan untuk komunikasi data dalam suatu area lokal dan sudah tersebar di berbagai tempat. Lokal yang dimaksud disini adalah area yang tidak terlalu luas yaitu dengan radius sekitar 20m atau dalam sebuah gedung. Untuk membangun jaringan lokal menggunakan WiFi, perangkat utama yang digunakan adalah AP. AP adalah piranti yang akan menjadi koordinator dalam jaringan lokal jika diinginkan topologi bintang (*star*) seperti diilustrasikan pada Gambar 2.3.

Gambar 2.3: Jaringan bintang menggunakan WiFi.

Gambar 2.3 memberi ilustrasi sebuah jaringan WiFi yang terdiri dari tiga buah komputer dan satu buah AP yang terhubung ke jaringan internet. Dengan konfigurasi tersebut, semua komputer yang ada di dalam jaringan WiFi dapat berkomunikasi

dengan internet dengan aturan yang ditentukan oleh AP.

Jika dilihat lebih dalam lagi, AP ini sebenarnya adalah piranti tertanam (*embedded device*) yang didalamnya sudah terdapat pusat pengolahan utama, memori, dan penyimpanan (*storage*). Dengan kenyataan inilah maka AP mempunyai potensi untuk menjagi *gateway* bagi jaringan WiFi (IP) dan WSN ke jaringan internet. Untuk mengembangkan aplikasi yang akan ditanamkan ke dalam AP, maka diperlukan sistem operasi yang sesuai untuk AP. Karena pada dasarnya sistem Operasi dari pabrikan yang sudah tertanam pada AP tidak dapat dimodifikasi dan dikonfigurasi secara mudah. Oleh karena itu, sistem operasi harus diganti dengan sistem operasi baru yang dapat dikustomisasi dengan mudah.

2.2.2 IQRF

IQRF adalah teknologi komunikasi nirkabel berbasis paket melalui frekuensi radio dalam pita frekuensi sub-GHz [12, 13]. Teknologi ini dimaksudkan untuk penggunaan umum saat konektivitas nirkabel dibutuhkan, baik *point to point* atau jaringan yang kompleks. Fungsionalitas lengkapnya bergantung semata-mata pada aplikasi berbahasa C yang ditulis oleh pengguna.

Piranti komunikasi dasar dari IQRF adalah sebuah modul pancar-rima (*transceiver*), termasuk unit mikrokontroller dengan sistem operasi tertanam yang mengimplementasikan lapisan *link* dan lapisan jaringan (*network layer*) yang mendukung jaringan jala (*mesh network*). Tidak ada tingkat komunikasi yang lebih tinggi seperti lapisan *transport* yang termasuk kedalam teknologi ini.

Protokol jaringan jala dari piranti IQRF adalah IQMESH ([14]). IQMESH menggunakan koordinator jaringan untuk mengendalikan keseluruhan komunikasi. Sebuah jaringan IQMESH dapat menampung hingga 65.000 sensor. Setiap sensor di dalam jaringan memiliki alamat yang unik (*logical address*) yang diberikan pada saat *bonding*. Algoritma perutean *Discovered Full Mesh* (DFM) memanfaatkan *Virtual Routing Structure* (VRS) yang dibentuk pada saat proses *discovery*.

Fitur-fitur yang dimiliki antara lain:

- Kecepatan, daya, dan ukuran data yang rendah,
- RF yang berbasis paket data, maksimal 128 Byte per paket,
- pita frekuensi sub-GHz (868 MHz, 916 MHz, dst.), *multichannel*, dan modulasi FSK,

- *bit rate* 1.2 kb/s - 86.2 kb/s,
- daya keluaran maksimal 20 mW,
- maksimal 65.000 piranti dalam satu jaringan,
- konsumsi daya yang rendah: 380 nA saat *standby*, 25 μA saat menerima.

2.2.3 XBee

XBee adalah sebuah merk dari Digi International untuk keluarga modul radio [15]. XBee pertama diperkenalkan dalam merk MaxStream pada tahun 2005 yang berdasarkan pada standar IEEE 802.15.4-2003 untuk *point to point* dan komunikasi bintang dalam *baud rate* 250 kbit/s.

Pada awalnya diperkenalkan dua model, yaitu 1mW XBee dan 100mW XBee-PRO. Sejak pertama kali diperkenalkan, beberapa buah XBee baru juga diperkenalkan dan semua XBee sekarang dipasarkan dengan merk Digi. Contoh piranti XBee dapat dilihat pada Gambar 2.4.

Gambar 2.4: Sepasang piranti XBee.

Pada Gambar 2.4 dapat dilihat dua buah XBee radio yang tidak tertanam pada piranti lain. XBee radio memiliki 20 kaki (*pin*) yang dapat ditancapkan pada piranti lain.

2.2.4 TCP/IP

Protokol internet adalah kumpulan protokol-protokol komunikasi yang digunakan dalam internet dan jaringan komputer sejenis, dan umumnya merupakan protokol yang paling populer untuk WAN. Pada umumnya hal ini dikenal dengan TCP/IP,

karena protokol utamanya merupakan protokol jaringan pertama yang terstandarisasi. Terkadang hal ini dikenal dengan model DoD karena pengaruh ARPANET pada dekade 1970an.

TCP/IP menyediakan konektivitas antar ujung yang menspesifikasikan bagaimana data harus diformat, dialamatkan, ditransmisikan, dirutekan, dan diterima di tujuan. TCP/IP memiliki empat layer abstraksi yang digunakan untuk mengurutkan semua protokol internet menurut jangkauan jaringan yang terlibat. Dari terendah sampai tertinggi, lapisan-lapisan tersebut adalah layer link, layer internet, layer transport, dan layer aplikasi.

2.2.5 *Access Point*

Access Point, disingkat AP, atau juga dikenal dengan istilah *Wireless Access Point* adalah sebuah piranti yang memungkinkan piranti-piranti nirkabel untuk terkoneksi dengan jaringan kabel menggunakan Wi-Fi atau standar lain. AP biasanya terkoneksi dengan sebuah *router* (melalui jaringan kabel) sebagai piranti yang berdiri sendiri, namun juga dapat menjadi bagian dalam komponen *router* tersebut.

Penggunaan secara korporat melibatkan beberapa AP ke dalam jaringan kabel dan menyediakan akses nirkabel ke LAN kantor. AP diatur dengan WLAN *Controller* yang menangani pengaturan daya RF, kanal-kanal, autentikasi, dan keamanan.

Sebuah *hotspot* adalah aplikasi dari satu atau beberapa AP, yang memungkinkan piranti dapat terhubung ke Internet dengan mudah. Konsep ini sudah menjadi hal yang umum di beberapa kota besar. Kombinasi dari warung kopi, perpustakaan, dan AP milik pribadi memungkinkan klien untuk terkoneksi dengan Internet. Koleksi dari *hotspot* yang terkoneksi dapat disebut sebagai sebuah jaringan *lily pad*.

2.2.6 TP-LINK MR3020

TP-LINK MR3020 adalah *Portable 3G/4G Wireless N Router* keluaran TP-LINK, perusahaan asal Shenzen, China, yang bergerak dalam bidang piranti jaringan komputer. TP-LINK MR3020 pada dasarnya adalah Wi-Fi router yang dapat meneruskan internet dari modem 3G/4G yang terpasang di port USB-nya. TP-LINK MR3020 termasuk AP yang populer karena bentuknya yang kecil, seperti dapat dilihat pada Gambar 2.5, sehingga mudah dibawa dan harganya yang tergolong murah. Banyak forum di internet yang membahas AP jenis ini, sehingga dukungan untuk memanipulasinya luas dari komunitas. Penelitian ini menggunakan sistem opera-

si OpenWRT guna memberikan fleksibilitas dalam pengembangan aplikasi *gateway* berbasis web.

Gambar 2.5: TP-LINK MR3020.

Seperti yang tampak pada Gambar 2.5, TP-LINK MR3020 memiliki satu buah *port* USB dan satu buah *port* RJ-45. Untuk sumber daya listrik, AP ini menggunakan koneksi *micro*-USB.

Spesifikasi teknis dari fitur perangkat keras dan komunikasi nirkabel dari MR3020 dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1: Fitur perangkat keras dari TP-LINK MR3020.

Fitur Perangkat Keras	
Antarmuka	1 10/100Mbps WAN/LAN Port, USB 2.0 Port for 3G/4G modem, a mini USB Port for power supply.
Tombol	Quick Setup Security Button, Reset Button, Mode Switch
Suplai Daya Eksternal	5VDC/1.0A
Dimensi (P x L xT)	2.9 x 2.6 x 0.9 in. (74 x 67 x22 mm)
Tipe Antena	Internal Antenna

Tabel 2.2: Fitur komunikasi nirkabel dari TP-LINK MR3020.

Fitur Komunikasi Nirkabel	
Standar Nirkabel	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Frekuensi	2.4-2.4835GHz
EIRP	<20dBm
Mode Nirkabel	3G Router, Travel Router (AP), WISP Client Router
Sekuritas Nirkabel	Support 64/128 bit WEP, WPA-PSK/WPA2-PSK, Wireless MAC Filtering

2.2.7 Web Server

Web server dapat mengacu pada perangkat keras atau perangkat lunak yang membantu dalam penyampaian konten web yang dapat diakses melalui internet.

Penggunaan web server yang paling umum adalah sebagai host untuk halaman web, walaupun ada beberapa penggunaan lain seperti game, media penyimpan data, atau penjalanan aplikasi perusahaan.

Pemilihan aplikasi yang bertindak sebagai Web Server pada piranti tertanam (*embedded device*) harus mempertimbangkan spesifikasi dari piranti tersebut. Penggunaan aplikasi Web Server yang biasa ditanamkan pada komputer *desktop* bahkan *server*, misal Apache, sangat tidak tepat bila ditanamkan pada piranti tertanam. Hal ini akan mengakibatkan pemborosan pada penggunaan sumber daya karena piranti tertanam memiliki sumber daya yang sangat terbatas. Piranti tertanam sebaiknya menggunakan aplikasi web server yang dirancang secara spesifik untuk piranti tersebut. Pilihan yang ada antara lain uhttpd dan light-httpd.

2.2.8 *Asynchronous JavaScript and XML*

Asynchronous JavaScript and XML, atau lebih populer dengan istilah AJAX, adalah kelompok dari teknik-teknik pengembangan web yang digunakan pada klien untuk membuat aplikasi asinkron. Dengan AJAX, aplikasi web dapat mengirim dan menerima data dari sebuah server secara asinkron tanpa mengganggu tampilan dari halaman yang ada. Data dapat diambil menggunakan obyek XMLHttpRequest.

AJAX bukanlah sebuah teknologi, tapi kelompok dari beberapa teknologi. HTML dan CSS dapat digunakan dalam kombinasi untuk menyajikan informasi yang akan ditampilkan. Kemudian, DOM diakses oleh JavaScript untuk menampilkan dan

mengijinkan pengguna untuk berinteraksi dengan informasi tertampil. JavaScript dan obyek XMLHttpRequest menyediakan sebuah metode pertukaran data secara asinkron antara *browser* dan *server* untuk menghindari proses muat ulang halaman secara keseluruhan.

2.2.9 OpenWRT

OpenWRT adalah sebuah sistem operasi untuk *embedded device* yang berbasis pada Linux kernel [16]. OpenWRT pada umumnya digunakan dalam routing *network traffic*. Komponen-komponen utamanya adalah Linux kernel, util-linux, uClibc dan BusyBox. Semua komponen sudah dioptimalkan dan dimampatkan untuk bisa dimuat dalam *router* rumahan yang memiliki keterbatasan media penyimpan dan memori. OpenWRT dapat dikonfigurasikan melalui antarmuka *command-line* (*ash shell*), seperti dapat dilihat pada Gambar 2.6, atau dengan antarmuka Web (LuCI). Terdapat kurang lebih 3.500 paket-paket perangkat lunak tambahan yang tersedia untuk diinstal melalui sistem manajemen paket *opkg*.

```
 _______                     ________          __
|       |.-----.-----.-----.|  |  |  |.----.|  |_
|   -   ||  _  |  -__|     ||  |  |  ||   _||   _|
|_______||   __|_____|__|__||________||__|  |____|
         |__| W I R E L E S S   F R E E D O M
 Backfire (10.03.1-RC5, r27608) --------------------
  * 1/3 shot Kahlua    In a shot glass, layer Kahlua
  * 1/3 shot Bailey's  on the bottom, then Bailey's,
  * 1/3 shot Vodka     then Vodka.
 ----------------------------------------------------
root@OpenWrt:~#
```

Gambar 2.6: Tampilan antarmuka *command-line* OpenWRT versi *BackFire*.

OpenWRT dapat berjalan pada router CPE (*Customer Premised Equipment*), *gateway* residensial, komputer saku (seperti Ben NanoNote), dan komputer jinjing. OpenWRT juga dapat berjalan pada komputer konvensional atau komputer dengan arsitektur x86. Banyak *patch* dari kode sesumber berbasis OpenWRT yang diubah ke dalam Linux kernel utama.

2.2.10 SSHFS

SSHFS (SSH Filesystem) adalah sebuah klien *filesystem* untuk *mount* dan berinteraksi dengan direktori dan arsip yang berlokasi pada server atau *workstation*. Klien berinteraksi dengan server dengan SSH *File Transfer Protocol* (SFTP), sebuah

protokol jaringan yang menyediakan akses ke arsip, transfer arsip, dan fungsionalitas manajemen arsip melalui aliran data yang didesain sebagai ekstensi dari protokol SSH versi 2.0.

2.2.11 Bootstrap

Bootstrap adalah koleksi gratis dari alat-alat untuk membuat situs web dan aplikasi berbasis web. Bootstrap terdiri dari HTML dan contoh desain berbasis CSS untuk tipografi, borang, tombol, navigasi, komponen antarmuka lain, dan juga ekstensi JavaScript yang bersifat opsional.

Bootstrap merupakan proyek paling populer pada GitHub, dan sudah digunakan oleh, diantaranya, NASA dan MSNBC.

2.2.12 Sublime Text 3

Sublime Text adalah aplikasi teks editor *cross-platform* yang memiliki *Application Programming Interface* (API) Python. Sublime Text juga dapat menambah fungsionalitasnya dengan beberapa tambahan *plugin*. Aplikasi ini bukan merupakan aplikasi *open source* atau aplikasi gratis. Versi 3 dari Sublime Text dirilis dalam versi beta pada 29 Januari 2013. Antar muka Sublime Text dapat dilihat pada Gambar 2.7.

Gambar 2.7: Aplikasi Sublime Text 3 yang berjalan pada Mac OS X.

Antar muka Sublime Text sangat sederhana. Antar muka terdiri atas *side bar* yang menampilkan direktori *project* yang sedang dikerjakan pada sebelah kiri dan teks editor pada sebelah kanan.

Sublime Text mendukung beberapa bahasa pemrograman dan mampu memberikan warna-warna pada kode tertentu (*syntax highlighting*) untuk mempermudah penulisan kode. Bahasa yang didukung antara lain ActionScript, AppleScript, ASP, batch files, C, C++, C#,Clojure, CSS, D, Diff, Erlang, Go, Graphviz (DOT), Groovy, Haskell, HTML, Java, JSP, JavaScript, JSON, LaTeX, Lisp, Lua, Makefiles, Markdown, MATLAB, Objective-C, OCaml, Perl, PHP, Python, R, Rails, Regular Expressions, reStructuredText, Ruby, Scala, shell scripts (Bash), SQL, Tcl, Textile, XML, XSL, dan YAML. Sublime Text juga dapat menambahkan dukungan *syntax highlighting* pada bahasa pemrograman lain dengan tambahan *plugin*.

BAB III
METODOLOGI PENELITIAN

3.1 Alat dan Bahan

Alat dan bahan yang digunakan pada penelitian ini terbagi atas perangkat keras dan perangkat lunak yang akan dijelaskan seperti berikut.

3.1.1 Perangkat Keras

Perangkat keras yang digunakan secara garis besar terdiri atas kit-kit *Wireless Sensor Network* (WSN) dari *vendor* IQRF dan XBee, dan sebuah *Access Point* (AP) TP-LINK MR3020 yang bertindak sebagai *gateway*. Untuk lebih jelasnya dapat dilihat pada daftar berikut.

a. Kit pancar-rima IQRF TR-53B (3 unit),
b. Kit pengunduh program CK-USB-04 (1 unit),
c. Kit pengembangan DK-EVAL-03 (2 unit),
d. Kit pengembangan CK-EVAL-04 (1 unit),
e. *XBee 802.15.4 Radios (Series 1)* (3 unit),
f. *XBee Explorer USB Board* (1 unit),
g. *2 channel Relay Shield For Arduino (With XBee/BTBee interface)* (2 unit),
h. Arduino Uno (2 unit),
i. TP-LINK MR3020 (1 unit),
j. Kabel USB ke Serial Prolific (1 unit).

3.1.2 Perangkat Lunak

Perangkat lunak yang digunakan terdiri atas perangkat lunak yang digunakan untuk mengembangkan aplikasi dan perangkat lunak yang digunakan untuk men-*deploy* aplikasi pada AP. Daftar berikut menjelaskan secara lebih detail tentang aplikasi yang digunakan.

a. Arduino for Mac OS X,
b. CoolTerm,
c. Driver FTDI for Mac OS X,

d. PHP, MySQL, dan uHTTPd,
e. Python dan pustaka PySerial,
f. IQRF IDE v 2.08 for TR-53B,
g. SSHFS,
h. Sublime Text 3.

3.2 Alur Penelitian

3.2.1 Pra Penelitian

Sebelum penelitian dimulai, dilakukan studi literatur terkait dengan sistem yang akan dibangun. Selain itu, analisis kebutuhan juga dirancang pada tahap ini. Setelah semua selesai, dilajutkan dengan penulisan proposal penelitian.

3.2.2 Pengembangan Aplikasi

Ada tiga aplikasi yang akan dibangun, yaitu aplikasi berbasis bahasa C untuk masing-masing sensor IQRF dan Arduino Uno, aplikasi berbasis web yang nantinya akan berinteraksi langsung dengan pengguna, dan aplikasi berbasis bahasa Python untuk mengomunikasikan sensor-sensor dengan aplikasi berbasis web seperti yang diilustrasikan pada Gambar 3.1.

Gambar 3.1: Diagram ilustrasi aplikasi yang dibangun.

Seperti yang dapat dilihat pada Gambar 3.1, aplikasi berbasis web dan aplikasi Python berjalan pada AP, sedangakan aplikasi C berjalan pada masing-masing sensor IQRF dan XBee (lebih tepatnya Arduino Uno).

Pendekatan Interoperabilitas

Secara lebih detil, ilustrasi komponen-komponen sistem yang dibangun beserta alur komunikasi dari masing-masing bagian dapat dilihat pada Gambar berikut.

Gambar 3.2: Diagram ilustrasi detil komponen-komponen dari sistem beserta alur komunikasinya.

Gambar 3.2 menggambarkan pendekatan interoperabilitas yang dilakukan pada penelitian ini. Seperti yang dapat dilihat pada Gambar 3.2, komunikasi antara *sink-node* dan aplikasi web (warna hijau) dijembatani oleh USB *to serial* dan aplikasi Python. Hal ini dimungkinkan karena masing-masing *sink-node* dari WSN sama-sama menggunakan komunikasi data serial untuk bertukar informasi dengan *gateway*. Pada pendekatan ini, seluruh aplikasi yang dirancang berjalan di atas sistem operasi OpenWrt. Pendekatan ini dapat menginteroperabilitaskan beberapa WSN dengan hanya satu buah *gateway*.

Pendekatan interoperabilitas WSN yang lain dapat dilakukan dengan menggunakan *gateway* internet dari masing-masing vendor. Pendekatan ini akan akan memanfaatkan *web-service* yang tersedia di masing-masing vendor untuk diinteroperabilitaskan. Namun demikian, untuk menempuh pendekatan ini, diperlukan biaya yang tinggi karena harga dari masing-masing *gateway* internet WSN berkisar di angka ratusan Dolar Amerika Serikat atau jutaan Rupiah. Oleh karena itu, pendekatan ini sangat bertolak belakang dengan tujuan penelitian ini yang mengedepankan sisi rendah biaya (*low-cost*) sehingga pendekatan ini tidak dilakukan. Selain itu, faktor ketertutupan dari *gateway* internet WSN juga menjadi kelemahan dari pilihan ini karena hal ini akan menyulitkan jika *web-service* yang telah tersedia kurang memenuhi kebutuhan dan diinginkan modifikasi *gateway*.

Teknis Pengembangan

Aplikasi untuk sensor-sensor IQRF terdiri dari dua bagian, yaitu aplikasi untuk sensor koordinator dan sensor simpul. Namun demikian, aplikasi sama-sama ditulis dan kembangkan menggunakan Sublime Text 3. Setelah kode sumber untuk aplikasi selesai dibuat, kode sumber dikompiliasi menggunakan IDE IQRF untuk kemudian diunggah ke sensor menggunakan bantuan aplikasi yang sama. Aplikasi yang dikembangkan adalah hasil fork dari iHome, aplikasi rumah hijau yang dikembangkan oleh [10].

Sedangkan aplikasi untuk Arduino Uno, yang bertugas menyala-matikan relay dengan komunikasi berbasis ZigBee, dikembangkan dengan aplikasi Arduino for Mac OS X dengan bahasa pemrograman C. Proses kompilasi dan pengunggahan dilakukan dengan bantuan aplikasi yang sama.

Aplikasi web dikembangkan dengan bahasa PHP pada server (terletak pada AP) dan JavaScript pada klien. Halaman yang tertampil pada web browser disusun menggunakan HTML5 dan CSS3 dengan bantuan pustaka Bootstrap agar halaman dapat bersifat responsif, yaitu dapat menyesuaikan tampilan sesuai ukuran layar web browser. AJAX diterapkan dalam pengembangan agar halaman web yang ditampilkan bersifat dinamis.

Aplikasi berbasis bahasa Python dikembangkan dengan bantuan pustaka (*library*) PySerial. Pustaka ini diperlukan agar Python dapat berkomunikasi dengan *port* serial, yaitu antar muka untuk berkomunikasi dengan sensor. Kode sumber ditulis menggunakan Sublime Text 3.

3.2.3 Evaluasi dan Perbaikan

Evaluasi dilakukan dengan melakukan simulasi dalam skala laboratorium. Simulasi yang diujikan mencakup semua fitur yang dimiliki oleh aplikasi untuk memastikan bahwa aplikasi dapat berjalan dengan semestinya. Kemudian Perbaikan dilakukan dengan bantuan SSHFS agar AP dapat mengakses direktori yang tersimpan pada komputer karena *coding* tidak dilakukan pada AP itu sendiri, melainkan komputer. Perbaikan dilakukan pada komputer karena OpenWRT tidak memiliki antar muka berbasis grafis *Graphical User Interface*.

3.2.4 Pasca Penelitian

Setelah penelitian selesai dilakukan dan aplikasi siap untuk diimplementasikan, naskah skripsi dan makalah skripsi ditulis sebagai manifesto penelitian.

3.2.5 Diagram Alir Penelitian

Diagram alir penelitian yang menjelaskan tentang alur pekerjaan penelitian dapat dilihat pada Gambar 3.3.

Gambar 3.3: Diagram alir penelitian.

Pada Gambar 3.3, dapat dilihat bahwa penelitian dilakukan dengan urutan studi literatur, analisis kebutuhan, pengembangan, iterasi perbaikan dan evaluasi, dan diakhiri dengan penulisan skripsi.

3.3 Tahapan Pelaksanaan

Rancangan arsitektur yang akan digunakan pada penelitian ini diilustrasikan seperti pada Gambar 3.4. Pada gambar tersebut diilustrasikan sebuah sistem yang terdiri atas dua buah WSN dengan protokol yang berbeda dan satu buah jaringan nir-kabel lokal (WiFi). Protokol WSN yang akan digunakan dalam penelitian ini adalah dari IQRF dan ZigBee. Pelaksanaan penelitian ini akan dibagi menjadi tiga paket pekerjaan atau *Work Package* (WP).

WP 1: Perancangan Perangkat Lunak

Pada tahap ini akan dilakukan studi literatur yang dititikberatkan pada sistem operasi untuk piranti tertanam (*embedded device*). Langkah selanjutnya adalah re-rancangan perangkat lunak yang akan ditanamkan pada *Access Point* (AP). Perangkat lunak yang akan ditanamkan harus bekerja secara efisien karena kemampuan komputasi yang terbatas pada AP.

Gambar 3.4: Arsitektur WSN dan WiFi dengan sebuah AP.

WP 2: Implementasi Perangkat Lunak

Implementasi perangkat lunak dilakukan pada tahap ini. Langkah pertama yang dilakukan adalah memastikan bahwa WSN dapat terhubungan dengan internet sesuai dengan yang direncanakan. Langkah selanjutnya adalah memastikan bahwa jaringan WiFi tidak mengalami gangguan setelah perangkat lunak yang baru tertanam

pada AP. Penambahan layanan-layanan yang diperlukan dapat pula dilakukan pada tahap ini.

WP 3: Integrasi dan Pengujian Seluruh Sistem

Jika jaringan WiFi dan dua protokol WSN masing-masing dapat berhubungan dengan internet, maka pada tahap ini akan dilakukan pengujian sistem secara keseluruhan. Pengujian dinaikkan dari skala lab menjadi skala *test-bed*. Pengujian dalam *test-bed* dilakukan untuk menjamin bahwa sistem yang dikembangkan bekerja sesuai dengan yang direncanakan.

3.4 Jadwal Kegiatan

Penelitian direncanakan akan dilaksanakan selama enam bulan. Rincian rencana jadwal penelitian dicantumkan dalam Tabel 3.1.

Tabel 3.1: Jadwal Penelitian.

No	Keterangan	Bulan					
		1	2	3	4	5	6
1	Studi literatur	■	■				
2	Desain		■	■			
3	Pembelian bahan			■			
4	Pembuatan prototipe			■	■	■	
5	Uji coba dan perbaikan				■	■	
6	Penulisan skripsi						■

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan Sistem

Bagian ini menjelaskan hal-hal yang terkait tentang pengembangan aplikasi sebelum penulisan kode sumber.

4.1.1 Fitur-Fitur Aplikasi

Kemampuan utama dari aplikasi yang dibangun adalah dapat mengendalikan sensor-sensor yang terhubung dengan *gateway*. Pengendalian yang dimaksud adalah menyala-matikan *relay* dan membaca temperatur yang terbaca. Selain itu, aplikasi juga dapat menambahkan sensor baru atau menghapus sensor lama. Untuk mendukung automatisasi, aplikasi juga dapat menjalankan *profile* tertentu dari kombinasi suhu dan relay atau waktu tertentu. Kemampuan aplikasi dapat dilihat pada daftar berikut.

- Mampu membaca dan menampilkan suhu yang terbaca pada sensor IQRF,
- mampu menyala-matikan relay pada piranti yang diinginkan,
- mampu menambah dan mengurangi piranti baru baik IQRF atau XBee,
- mampu menjalankan *profile* tertentu dari kombinasi suhu dan relay atau waktu tertentu.

Pengguna diharuskan untuk *login* terlebih dahulu sebelum dapat menggunakan aplikasi. Jika pengguna terdaftar sebagai *administrator* maka pengguna tersebut dapat menambahkan pengguna baru, menghapus akun pengguna, mengangkat dan menurunkan pengguna menjadi *administrator*. Kemampuan aplikasi dalam menangani pengguna dapat dilihat pada daftar berikut.

- Mengharuskan pengguna untuk memasukkan nama dan kata sandi sebelum masuk ke aplikasi,
- dapat menambah atau mengurangi pengguna yang dapat memasuki sistem.

4.1.2 *Use Case Diagram*

Aktor yang terlibat dalam aplikasi ini hanya satu seperti yang tertera pada Gambar 4.1. Untuk memanfaatkan semua fitur yang ada pada aplikasi, pengguna diharuskan untuk *login* terlebih dahulu. *Case-case* yang ada pada aplikasi ini adalah, membaca suhu, menambah-kurangi sensor IQRF, menambah-kurangi sensor XBee, menyala-matikan relay, dan menambah-kurangi pengguna baru, seperti yang tergambar pada Gambar 4.1.

Gambar 4.1: Diagram *use case* dari penelitian.

Pada Gambar 4.1, menambah-kurangi sensor XBee tidak melibatkan sensor XBee (tidak ada garis yang menghubungkan keduanya), pada *case* tersebut hanya melibatkan aplikasi. Saat *case* ini dijalankan, data sensor pada basis data dihapus. Namun, menambah-kurangi sensor IQRF melibatkan sensor IQRF (ada garis yang menghubungkan keduanya). Karena pada *case* ini, selain menghapus data pada basis data, proses *unbonding* pada IQRF juga melibatkan *node* IQRF.

4.1.3 Diagram Arsitektur Sistem

Penelitian ini menggunakan AP, *sensor coordinator*, dan *sensor node* sebagai piranti keras. Sedangkan piranti lunak yang dikembangkan meliputi aplikasi berbasis

web, aplikasi Python, dan aplikasi C yang terunggah pada sensor-sensor. Gambar 4.2 menggambarkan diagram arsitektur sistem.

Gambar 4.2: Diagram Arsitektur Sistem.

Seperti terlihat pada Gambar 4.2, pengguna yang ingin mengakses aplikasi berbasis web guna mengendalikan sensor harus menggunakan web browser. Web browser yang dimaksud tidak terbatas hanya yang terinstal pada komputer, namun juga web browser yang terdapat pada ponsel cerdas atau komputer tablet.

4.1.4 *Software Development Life Cycle* (SDLC)

Pengembangan aplikasi ini menggunakan konsep pengembangan iteratif dan inkremental. Pada konsep pengembangan ini, kode-kode aplikasi dirancang, dikembangkan, dan diuji pada siklus yang terulang. Pada tiap iterasi, fitur tambahan dapat dirancang, dikembangkan, dan diuji sampai terbangun aplikasi yang berfungsi seperti yang dikehendaki. Ilustrasi SDLC yang digunakan dapat dilihat pada Gambar 4.3.

Gambar 4.3: Diagram SDLC.

Aplikasi akan dikembangkan fitur demi fitur dan diuji apakah masing-masing fitur dapat berjalan sebagaimana mestinya. Masing-masing aplikasi yang dikembangkan (aplikasi web, Python, dan C) juga akan dikembangkan dan diuji apakah masing-masing aplikasi dapat saling berinteraksi. Iterasi terus dilakukan hingga fitur-fitur dan aplikasi-aplikasi tersebut dapat berfungsi untuk interoperabilitas *Wireless Sensor Network* (WSN) dan protokol Internet.

4.2 Pengembangan Aplikasi

Tahap-tahap dalam pengembangan aplikasi adalah persiapan, pengembangan aplikasi WSN yang ditulis dengan bahasa C, pengembangan aplikasi Python untuk menghubungkan aplikasi web dan sensor, pengembangan aplikasi berbasis web, serta evaluasi dan perbaikan.

4.2.1 Persiapan Pra Pengembangan Aplikasi

Persiapan yang harus dilakukan sebelum mengembangkan aplikasi adalah mengkonfigurasi router/AP, mengkonfigurasi komputer yang akan digunakan untuk pengembangan aplikasi, dan mengkonfigurasi pengalamatan pada XBee.

Konfigurasi Router/*Access Point* (AP)

Penelitian ini menggunakan AP keluaran TP-LINK seri MR3020. AP jenis ini dipilih karena bentuknya yang kecil sehingga mudah dibawa atau dipindahkan

dan kemudahannya untuk dimodifikasi. TP-LINK MR3020 juga terbilang populer di ranah komunitas sistem benam (*embedded device*) sehingga memiliki dukungan yang baik dari pabrikan dan komunitas.

Sebelum digunakan, *firmware* bawaan TP-LINK MR3020 harus diganti dengan sistem operasi OpenWRT. Proses penggantian cukup mudah karena hanya memanfaatkan menu *firmware upgrade* dari web admin yang sudah tersedia. Sistem Operasi OpenWRT yang digunakan adalah Attitude Adjustment versi 12.09 dengan Linux kernel 3.3.8. *Image file* sistem operasi tersebut bisa diunduh secara gratis pada situs `www.openwrt.org`.

Setelah OpenWRT berhasil terinstall, langkah selanjutnya adalah mengimplementasikan *extroot*. *Extroot* adalah fitur dari OpenWRT yang dapat memetakan direktori *root* pada memori eksternal, sehingga *extroot* dapat memperbesar memori penyimpanan dengan bantuan USB *flash drive*. Langkah yang harus dilakukan pertamakali adalah menginstall perangkat lunak dengan perintah sebagai berikut:

```
# opkg update
# opkg install block-extroot block-hotplug block-mount
```

Kemudian dilanjutkan dengan menyalin isi dari memori internal TP-LINK MR3020 ke USB *flash drive* yang di-*mount* pada direktori /mnt/sda1 dengan perintah:

```
# tar -C /overlay -cvf - . | tar -C /mnt/sda1 -xf -
```

Langkah terakhir adalah mengkonfigurasi file /etc/config/fstab dan menyalakan ulang AP. Konfigurasi diganti dengan detil sebagai berikut:

```
config mount
        option target        /mnt
        option device        /dev/sda1
        option fstype        ext3
        option options       rw,sync
        option enabled       1
        option enabled_fsck  0
        option is_rootfs     1
```

Konfigurasi di atas akan memetakan direktori *root* (/) ke dalam piranti (*device*) /dev/sda1 yang di-*mount* pada direktori /mnt dengan format *filesystem* ext3 dan dan mode baca tulis.

Setelah proses implementasi extroot selesai, berarti AP sudah memiliki memori penyimpanan yang cukup (atau mungkin lebih) untuk menginstal aplikasi-aplikasi pendukung lainnya. Penelitian ini menggunakan memori USB 4 Gigabyte.

Agar TP-LINK MR3020 dapat membaca dan mengirimkan data dari dan ke WSN melalui kanal serial, dibutuhkan Python dan pustaka PySerial yang diinstal dengan perintah:

```
# opkg update
# opkg install python pyserial
```

Sedangkan paket aplikasi 'at' digunakan agar AP dapat menjalankan perintah untuk menyala-matikan relay pada waktu tertentu. Namun, sebelum at dapat berjalan, pada router sudah harus tersedia file /var/spool/cron/atjobs/.SEQ dan pemilik dari file tersebut harus diganti menjadi daemon.daemon. Aplikasi 'at' diinstall dengan perintah:

```
# opkg update && opkg install at
```

Langkah selanjutnya adalah instalasi aplikasi pendukung aplikasi berbasis web yang nantinya akan dikembangkan yaitu Web Server, PHP, dan MySQL. Web server yang digunakan adalah yang sudah terinstal pada OpenWRT versi Attitude Adjustment yaitu uHTTPd. Sedangkan PHP dan MySQL harus diinstal dengan perintah:

```
# opkg update
# opkg install php5 php5-cgi mysql
```

Setelah PHP berhasil terinstal, buka file /etc/config/uhttpd dan pastikan baris yang memuat baris di bawah ini tidak dalam keadaan terkomentar.

```
list interpreter ".php=/usr/bin/php-cgi"
```

Aplikasi yang terakhir yang harus diinstal adalah SSHFS yang berguna dalam uji coba dan perbaikan aplikasi. SSHFS dapat diinstal dengan perintah:

```
# opkg update
# opkg install sshfs
```

Zona waktu standar pada OpenWRT adalah UTC yang berlokasi pada kota Greenwich di Inggris Raya. Agar zona waktu dapat dikonfigurasi sesuai dengan zona waktu kota Yogyakarta, maka isi dari file /etc/config/system harus disesuaikan. Pada file tersebut, zona waktu UTC diganti menjadi WIT-7 atau *Western Indonesian Time*-7.

Konfigurasi Komputer untuk Pengembangan

Komputer yang digunakan dalam penelitian ini adalah MacBook Pro dengan sistem operasi Mac OS X Mountain Lion. Sedangkan aplikasi yang harus tersedia adalah Sublime Text 3, Arduino for Mac OS X, CoolTerm, Driver FTDI for Mac OS X, IQRF IDE v 2.08 for TR-53B.

Sublime Text 3 berperan banyak dalam mengedit kode-kode sumber PHP, C, dan Python.

Sedangkan proses pengembangan dan pengunduhan aplikasi untuk Arduino sepenuhnya dilakukan dengan Arduino for Mac OS X karena aplikasi tersebut sudah mencakup editor teks dan alat kompilasi.

Agar komputer dapat membaca kanal serial sehingga dapat melakukan tugas-tugas seperti konfigurasi XBee Radio dan pengujian aplikasi Python, komputer membutuhkan aplikasi CoolTerm. Sebelum aplikasi ini terinstal, pastikan *driver* FTDI untuk Mac OS X sudah terinstal, agar komputer dapat mendeteksi XBee yang terhubung melalui USB.

Walaupun kode sumber untuk IQRF ditulis dengan bantuan Sublime Text 3, proses kompilasinya dilakukan dengan aplikasi IQRF IDE v 2.08 for TR-53B. Proses pengunduhan file ke IQRF juga dilakukan dengan aplikasi yang sama.

Semua proses instalasi aplikasi yang dibutuhkan dilakukan dengan prosedur standar dari Mac OS X. Yaitu memperoleh file binary-nya dan kemudian menyalin file tersebut ke direktori Application atau mengikuti prosedur standar dari masing-masing aplikasi.

Langkah terakhir yang dilakukan adalah memastikan AP dapat mengakses komputer melalui SSH. Pastikan *remote sharing* dalam keadaan tercentang di *System Preference*, *Sharing*. Atau untuk lebih jelasnya dapat melihat pada Gambar 4.4.

Gambar 4.4: Konfigurasi SSH pada Mac OS X.

Piranti XBee

Setiap piranti XBee yang akan digunakan harus terkonfigurasi terlebih dahulu. Parameter-parameter yang harus terkonfigurasi adalah ATID (alamat dari jaringan/*network ID*), ATMY (alamat piranti itu sendiri), ATDH (*destination address high*), dan ATDL (*destination address low*).

Proses konfigurasi dilakukan dengan menyambungkan XBee Radio dengan *XBee Explorer USB Board* ke komputer dan menjalankan aplikasi untuk membaca kanal serial, misalnya CoolTerm pada Mac OS X. Setelah CoolTerm dibuka, buka koneksi kanal serial ke XBee dan jalankan perintah dengan format:

```
+++ #masuk ke AT Mode
ATID <id jaringan>
ATMY <alamat dari zigbee>
ATDH <destination high>
ATDL <destination low>
ATWR  #tulis ke non volatile memory
```

Contoh perintahnya adalah:

```
+++ #masuk ke AT Mode
```

```
ATID 1234
ATMY 5
ATDH 0
ATDL 1
ATWR  #tulis ke non volatile memory
```

4.2.2 Pengembangan Aplikasi WSN

Langkah selanjutnya yang dilakukan dalam pengembangan adalah pengembangan aplikasi untuk WSN yang terdiri atas aplikasi C yang masing-masing ditujukan untuk IQRF dan XBee. Setelah aplikasi selesai ditulis, kode sumber kemudian di-*compile* dan diunduh ke dalam WSN.

Aplikasi IQRF

Piranti IQRF dalam penelitian ini ada dua jenis, begitu pula aplikasi yang tertanam di dalamnya. Aplikasi tersebut adalah *Sink* dan *Node*. Aplikasi *Sink* akan ditanamkan pada koordinator dari IQRF yang terhubung langsung dengan AP melalui kabel USB dan jumlahnya hanya satu. Sedangkan aplikasi *Node* akan ditanamkan pada sensor-sensor IQRF yang akan saling membentuk jaringan jala (*mesh network*) dan jumlahnya lebih dari satu.

Aplikasi *Sink* lebih ditekankan pada pembentukan ikatan (*bond*) antara koordinator dan sensor-sensor, pengendalian sensor-sensor, serta pengakuisisian data dari masing-masing sensor.

Sedangkan aplikasi *Node* lebih bersifat pasif karena hanya akan merespon perintah-perintah yang akan dikirimkan oleh koordinator, seperti pembentukan ikatan dan pembacaan temperatur.

Data yang akan diakuisisi dari masing-masing sensor adalah temperatur yang terbaca di masing-masing sensor.

Aplikasi yang digunakan adalah aplikasi iHome yang dikembangkan oleh [10] yang fiturnya sebenarnya tidak terbatas hanya pada pembacaan temperatur sekitar.

Aplikasi XBee

Sama seperti IQRF, piranti XBee terdiri dari dua jenis piranti, yaitu koordinator dan sensor-sensor. Koordinator bertugas menyalakan atau mematikan relay-relay yang terdapat di masing-masing sensor sesuai permintaan pengguna.

Koordinator tersusun atas *XBee 802.15.4 Radios (Series 1)* yang tertanam pada *XBee Explorer USB Board* dan terkoneksi langsung dengan AP dengan kabel USB.

Sedangkan sensornya terdiri dari tiga bagian, yaitu *XBee 802.15.4 Radios (Series 1)*, *2 channel Relay Shield For Arduino (With XBee/BTBee interface)*, dan Arduino Uno. Ketiga bagian tersebut saling terhubung dengan pin-pin yang tersedia.

XBee pada hakikatnya hanya sebatas piranti komunikasi antara AP dengan relay-relay yang tersedia, sehingga pemrograman teletak pada masing-masing Arduino pada relay. Bahasa yang digunakan adalah bahasa C (Arduino). Aplikasi yang dikembangkan akan menunggu perintah dari AP untuk menyalakan atau mematikan relay.

4.2.3 Pengembangan Aplikasi Python

Aplikasi Python adalah jantung dari penelitian ini karena menghubungkan aplikasi berbasis web dengan piranti-piranti IQRF dan XBee. Aplikasi terdiri dari tiga bagian yang ditujukan untuk penggunaan berbeda, yaitu komunikasi dengan sensor IQRF, XBee, dan profil untuk automatisasi.

Manipulasi Komunikasi Data Serial

Aplikasi yang nantinya tertanam pada AP, yaitu aplikasi berbasis web dan aplikasi Python, akan berjalan di atas sistem operasi OpenWrt. Hal ini berarti aplikasi yang dibangun tidak akan bersentuhan secara langsung dengan aliran data aras rendah seperti aliran bit-bit data. Segala bentuk komunikasi data aras rendah yang melibatkan bit-bit data dikelolah sepenuhnya oleh OpenWrt sehingga aras terendah dari komunikasi data yang terbaca oleh aplikasi adalah pada aras karakter yang saling dipertukarkan oleh masing-masing aplikasi yang dibangun. Hal ini dijelaskan dengan ilustrasi sebagai berikut.

Gambar 4.5: Manipulasi komunikasi data serial [1].

Dari Gambar 4.5, tampak bahwa alur manipulasi data serial aras rendah dari *sink node* dimulai dari USB to serial, sistem operasi OpenWrt, aplikasi Python, kemudian pada akhirnya aplikasi berbasis web. Pengubahan bit-bit data serial terjadi pada bagian OpenWrt OS.

Aplikasi IQRF

Aplikasi Python IQRF memiliki kemampuan untuk melakukan *bonding* antara koordinator dan sensor, dan pembacaan temperatur pada sensor dengan ID tertentu.

Diagram alir dari aplikasi berikut dapat dilihat pada Gambar 4.6.

Gambar 4.6: Diagram alir aplikasi Python untuk IQRF.

Aplikasi dijalankan dengan menjalankan format perintah berikut pada *terminal console*:

```
$ python iqrf.py <perintah><ID node>
```

Aplikasi ini berjalan dalam bentuk CLI dan membutuhkan satu parameter, yaitu perintah yang langsung disambung dengan ID node tanpa spasi. Perintah yang tersedia yaitu membaca gemperatur pada node ID tertentu (g), *bonding* node ID tertentu (b), *unbonding* node ID tertentu (u).

Sehingga contoh penggunaan aplikasi pada *terminal console*:

```
$ python iqrf.py g3
```

Perintah di atas adalah perintah untuk membaca temperatur pada node ID 3.

Aplikasi XBee

Aplikasi Python XBee memiliki kemampuan untuk menyalakan atau mematikan relay pada piranti tertentu dan membaca status relay pada piranti tertentu, apakah relay tersebut sedang menyala atau mati.

Diagram alir dari aplikasi berikut dapat dilihat pada Gambar 4.7.

Gambar 4.7: Diagram alir aplikasi Python untuk menangani piranti XBee.

Aplikasi dijalankan dengan menjalankan format perintah berikut pada *terminal console*:

```
$ python xbee.py <perintah> <ATMY piranti> <ID relay>
```

Aplikasi ini berjalan dalam bentuk CLI dan membutuhkan tiga parameter, yaitu perintah, alamat piranti (ATMY), dan ID relay (terdapat dua relay di tiap piranti). Perintah yang tersedia yaitu menyalakan relay, 'on', mematikan relay, 'off', dan membaca status relay, 'status'.

Sehingga contoh penggunaan aplikasi pada *terminal console*:

```
$ python iqrf.py status 2 1
```

Perintah di atas akan membaca status (on/off) dari relay 1 pada piranti dengan alamat (ATMY) 2 dan menampilkannya pada layar dalam karakter 'H' (menyala) atau 'L' (mati).

Profiling

Aplikasi ini memanfaatkan aplikasi *at* pada Linux yang dapat menjalankan perintah tertentu di *terminal console* pada waktu tertentu. Dengan aplikasi ini, pengguna dapat menyala-matikan relay pada waktu tertentu. Jika hal ini dikombinasikan dengan bacaan temperatur dari sensor IQRF, maka pengguna dapat menyala-matikan relay tertentu pada saat temperatur pada sensor IQRF node tertentu mencapai suhu tertentu, dan juga bisa hanya terjadi saat waktu tertentu.

Diagram alir dari aplikasi berikut dapat dilihat pada Gambar 4.8.

4.2.4 Pengembangan Aplikasi Berbasis Web

Aplikasi web untuk mengendalikan sensor-sensor IQRF dan XBee dibangun dengan PHP dan basis data MySQL. Aplikasi ini dibangun tanpa menggunakan *framework* karena aplikasi ini tergolong tidak terlalu rumit, aplikasi ini hanya memliki lima halaman utama seperti terlihat pada sitemap Gambar 4.9. Penggunaan *framework* juga akan memakan sumber daya komputasi sedikit lebih banyak, padahal aplikasi ini akan diimplementasikan pada sebuah AP yang memiliki tingkat komputasi yang rendah.

Antarmuka yang dikembangkan adalah hasil *fork* dari *template* halaman administrator buatan Vince G [17]. Antarmuka ditulis menggunakan HTML5 dan CSS3, dengan tambahan pemrograman JavaScript pada sisi klien agar menambah interaktivitas. Pustaka Bootstrap digunakan untuk membuat halaman web menjadi responsif, sehingga dapat menyesuaikan diri secara otomatis sesuai dengan ukuran layar. Pustaka jQuery digunakan untuk membantu pemrograman di sisi klien, sedangkan kode sumber untuk aplikasi berbasis web ditulis dengan bantuan Sublime Text 3.

Gambar 4.8: Diagram alir aplikasi Python untuk menyala-matikan lampu pada kondisi tertentu.

Peta situs dari aplikasi berbasis web yang dikembangkan dapat dilihat pada Gambar 4.9.

Gambar 4.9: Peta situs aplikasi web.

Total halaman utama dari situs ada enam halaman, seperti dapat dilihat pada Gambar 4.9 dengan tambahan halaman login, halaman yang harus diakses sebelum pengguna dapat masuk ke dalam sistem. Empat halaman utama mempunyai halaman anak yang secara garis besar ditujukan untuk melihat, menghapus dan menambah piranti, profil, atau pengguna.

Basis data yang dikembangkan tergolong sederhana karena hanya dimaksudkan untuk menyimpan data-data yang sifatnya kecil, seperti tampak pada Gambar 4.10.

Gambar 4.10: *Entity Relationship Diagram* dari basis data untuk aplikasi web.

Seperti yang terpampang pada Gambar 4.10, basis data terdiri atas empat tabel. Relasi yang ada pada basis data hanya relasi tabel *iqrf_device* dengan *profile* dan *xbee_device* dengan *profile*.

Sedangkan diagram alir untuk menambah piranti IQRF dan XBee dijelaskan pada Gambar 4.11 dan Gambar 4.12.

Gambar 4.11: Diagram Alir Penambahan Piranti IQRF ke Aplikasi.

Gambar 4.12: Diagram Alir Penambahan Piranti XBee ke Aplikasi.

Penambahan piranti IQRF mengharuskan pengguna untuk menekan tombol yang ada pada sensor IQRF, sedangkan penambahan piranti XBee tidak mengharuskan pengguna untuk menekan tombol atau mengintervensi perangkat keras XBee.

4.2.5 Skalabilitas Sistem

Seperti yang sudah dijelaskan sebelumnya tentang rancangan teknis sistem, secara *default* aplikasi ini hanya mendukung interoperabilitas antara dua jenis WSN. Karena pendekatan interoperabilitas dilakukan pada ranah data dan aplikasi disusun hanya untuk mendukun dua jenis WSN, sistem tidak memiliki skalabilitas dalam hal penambahan jenis WSN. Jika diinginkan jenis WSN baru yang ingin diinteroperabilitaskan, maka harus ditambahkan beberapa baris kode baru untuk menangani WSN baru yang mungkin struktur komunikasinya sedikit berbeda.

Walaupun demikian, sistem yang dibangun memiliki skalabilitas dalam hal penambahan jumlah piranti yang terpasang. Piranti atau *node* yang ingin dipantau dengan aplikasi dapat ditambahkan dan dikurangi melalui antarmuka yang telah dirancang dengam mudah.

4.2.6 Evaluasi dan Perbaikan

Evaluasi dilakukan dengan mengujikan semua fitur yang dimiliki oleh aplikasi. Jika terdapat kesalahan, maka dilacak dimanakah letak kesalahan berada, apakah pada aplikasi berbasis web, aplikasi Python, atau aplikasi C pada sensor.

Masalah yang timbul adalah saat akan merevisi aplikasi yang sudah terlanjur tertanam pada AP karena AP, dengan sistem operasi OpenWRT, tidak memiliki aplikasi teks editor yang mumpuni untuk melakukan pengeditan kode sumber dengan nyaman. Oleh karena itu, seluruh kode sumber tetap berada pada komputer agar tetap bisa dibuka dengan Sublime Text 3, namun AP me-*mount*-nya dengan bantuan SSHFS sehingga seolah-olah kode sumber tersebut berada dalam dua tempat dan saling tersinkronisasi.

Mounting pada AP dilakukan dengan menggunakan perintah:

```
# sshfs /direktori/pada/ap <user>@<alamat IP>:/direktori/pada/komputer
```

Contoh penggunaan perintah tersebut:

```
# sshfs /www/web/ gtrdp@192.168.1.1:/var/www/web/
```

4.2.7 *Screenshot* Aplikasi

Bagian ini menampilkan beberapa *screenshot* dari aplikasi web yang dikembangkan.

Halaman pertama yang akan terbuka setelah pengguna melakukan *log in* adalah halaman *dashboard* seperti yang dapat dilihat pada Gambar 4.13. Tampilan halaman web akan menyesuaikan dengan layar pengguna karena halaman tersebut didesain responsif.

(a) Dibuka pada layar komputer.

(b) Dibuka di layar ponsel yang kecil.

Gambar 4.13: Halaman *dashboard* saat belum ada piranti terpasang.

Seperti dapat dilihat pada Gambar 4.13b, ukuran layar akan menjadi lebih kecil jika dibanding dengan Gambar 4.13a karena halaman dibuka pada ponsel cerdas yang memiliki ukuran layar lebih kecil dari komputer.

Aplikasi mempunyai halaman untuk menambahkan piranti IQRF, sehingga pengguna dapat menambahkan piranti IQRF baru ke dalam sistem menggunakan ha-

laman ini. Gambar 4.14 menunjukkan halaman untuk menambahkan piranti IQRF.

Gambar 4.14: Halaman untuk menambahkan piranti IQRF.

Seperti dapat dilihat pada Gambar 4.14, pengguna diharuskan untuk memasukkan alamat *node* dari IQRF yang akan ditambahkan. Alamat tersebut adalah bilangan bulat dari dua dan tidak boleh menggunakan alamat yang sudah terpasang pada sistem. Pada bagian atas halaman, terdapat langkah-langkah yang harus diikuti pengguna untuk menambahkan piranti IQRF baru.

Seperti piranti IQRF, aplikasi juga menyediakan halaman khusus untuk menambahkan piranti XBee, seperti yang dapat dilihat pada Gambar 4.15.

Gambar 4.15: Halaman untuk menambahkan piranti XBee.

Pengguna juga diharuskan untuk memasukkan alamat ATMY dari XBee yang akan ditambahkan, seperti cara menambahkan piranti IQRF. Pada bagian atas halaman, terdapat langkah-langkah yang harus diikuti pengguna untuk menambahkan piranti XBee baru.

Saat piranti sudah ditambahkan melalui halaman-halaman yang sudah dijelaskan sebelumnya, halaman *dashboard* akan berubah, seperti yang tergambar pada Gambar 4.16. Halaman ini masih bersifat responsif, sehingga akan menyesuaikan dengan layar di mana halaman tertampil.

(a) Dibuka pada layar komputer.

(b) Dibuka pada layar ponsel yang kecil.

Gambar 4.16: Halaman *dashboard* saat piranti sudah terpasang.

Pada saat halaman dibuka pada layar yang kecil, maka perbedaan yang signifikan adalah hilangnya *sidebar* dan mengecilnya *navigation bar* seperti yang dapat dilihat pada Gambar 4.16a dan 4.16b.

Berapapun banyaknya sensor IQRF atau XBee yang terpasang, halaman *dashboard* tetap akan menampilkan satu piranti saja, yaitu piranti yang pertama kali terpasang. Jika penggune ingin melihat semua sensor yang terpasang, pengguna dapat membuka halaman khusus untuk melihat semua sensor IQRF yang terpasang, seperti yang ditunjukkan pada Gambar 4.17.

Gambar 4.17: Halaman daftar piranti IQRF yang terpasang.

Pada Gambar 4.17 tertampil tiga buah termometer digital yang masing-masing menampilkan hasil bacaan temperatur dari masing-masing sensor IQRF. Perubahan suhu dapat dilihat secara langsung.

Pengguna juga dapat melihat semua piranti XBee yang terpasang dengan halaman khusus yang tertera pada Gambar 4.18. Halaman ini akan menampilkan semua piranti XBee yang terdaftar pada aplikasi.

Gambar 4.18: Halaman daftar piranti XBee yang terpasang.

Seperti yang dapat dilihat pada Gambar 4.18, terdapat tiga piranti XBee yang terdaftar pada aplikasi. Melalui halaman ini, pengguna dapat memati-nyalakan sensor dengan meng-klik saklar yang diinginkan. Dibutuhkan sekitar dua sampai tiga detik untuk menyalakan relay.

Seperti yang sudah dijelaskan, aplikasi mendukung profil guna otomatisasi tugas-tugas tertentu. Jika pengguna mengiginkan untuk menambahkan profil tertentu, tersedia halaman khusus seperti yang terlihat pada Gambar 4.19. Jika suhu yang terbaca di sensor IQRF tertentu mencapai nilai yang diinginkan, maka kondisi relay pada piranti XBee tertentu akan terkonfigurasi seperti yang diinginkan pengguna. Untuk lebih jelasnya bisa dilihat pada Gambar 4.19.

Gambar 4.19: Halaman untuk menambahkan profil.

Informasi yang harus dimasukkan ke dalam aplikasi agar pengguna dapat menambahkan profil baru, seperti yang terlihat pada Gambar 4.19, adalah:

- temperatur yang diinginkan,

- konfigurasi relay saat temperatur tercapai,
- nama profil,
- alamat sensor IQRF yang diinginkan (pembacaan temperatur),
- alamat ATMY piranti XBee yang diinginkan (relay),
- dan tanggal dan waktu yang diinginkan.

Namun jika pengguna hanya ingin menyala-matikan relay XBee pada waktu tertentu, maka pengguna tinggal memilih alamat nol (0) untuk sensor IQRF. Dengan demikian, pembacaan temperatur pada IQRF akan diabaikan.

Jika pengguna ingin relay XBee terkonfigurasi saat IQRF membaca temperatur tertentu, tanpa memperhatikan tanggal dan waktu, pengguna hanya mencentang *checkbox* yang sudah tersedia. Dengan demikian, profil akan berjalan tanpa memperhatikan waktu tertentu.

Halaman selanjutnya adalah halaman yang menampilkan semua pengguna yang terdaftar, seperti yang dapat dilihat pada Gambar 4.20. Namun yang dapat mengakses halaman ini adalah pengguna yang berstatus *administrator*.

Gambar 4.20: Halaman yang menampilkan daftar pengguna.

Pada halaman ini, terdapat tautan untuk menambahkan pengguna baru, meng-edit data pengguna yang sudah terdaftar, menghapus data pengguna, dan menaikkan status pengguna menjadi *administrator* atau menurunkannya.

Jika pengguna meng-klik tautan untuk menambahkan pengguna baru, maka pengguna akan diarahkan menuju halaman baru seperti yang dapat dilihat pada Gambar 4.21. Pada halaman ini, pengguna diminta untuk memasukkan informasi-informasi tentang pengguna baru.

Gambar 4.21: Halaman untuk menambahkan pengguna baru.

Informasi yang harus dimasukkan, seperti yang dapat dilihat pada Gambar 4.21, antara lain:

- *username* pengguna baru,
- nama lengkap pengguna baru,
- dan kata sandi pengguna (dua kali).

Untuk dapat mengakses semua fitur/halaman di atas, pengguna harus *log in* terlebih dahulu. Halaman login dari aplikasi web terlihat pada Gambar 4.22.

Gambar 4.22: Halaman login.

Informasi yang dibutuhkan untuk *log in* hanya *username* atau nama pengguna, dan *password* atau kata sandi.

Pendekatan yang dilakukan pada aplikasi ini tidak seperti aplikasi web pada umumnya, di mana pengguna dapat mengklik tautan registrasi yang biasanya terdapat pada halaman *log in*. Pendekatan yang dilakukan seperti pada proses *log in* pada komputer. Jadi jika pengguna baru ingin masuk, ia bisa mendaftarkan dirinya melalui pengguna lama yang sudah dapat *log in*, menggunakan halaman yang sudah dijelaskan sebelumnya pada Gambar 4.21.

4.2.8 Kode Sumber

Kode sumber dapat diperoleh pada situs GitHub dengan alamat URL `https://github.com/gtrdp/wsn-ip-interoperability`. Pada laman tersebut, kode sumber dapat diunduh dalam format .zip. Dengan bantuan aplikasi git, *project* dari aplikasi ini, lengkap dari aplikasi C, Python, dan Web, dapat di-kloning.

4.3 Analisis Unjuk Kerja Aplikasi

Bagian terakhir dari skripsi ini akan ini menjelaskan hal-hal terkait tentang cara perangkaian perangkat keras sehingga dapat difungsikan, instalasi aplikasi ke piranti sehingga sesuai kondisi sesungguhnya, hasil uji coba aplikasi, serta masalah yang dihadapi dan penyelesaiannya.

4.3.1 Instalasi Piranti

Perangkat keras pertama yang akan dirakit adalah koordinator IQRF yang tersambung langsung pada AP. Koordinator IQRF ini terdiri atas kabel USB to Serial Prolific, kit pengembangan CK-EVAL-04, dan IQRF TR-52B seperti dapat dilihat pada Gambar 4.23. Kondisi koordinator sensor IQRF sebelum dirakit dapat dilihat pada Gambar 4.23a.

(a) Sebelum dirakit.

(b) Setelah dirakit.

Gambar 4.23: Koordinator sensor IQRF yang akan disambungkan pada AP.

Kemudian semua bagian disambungkan pada tempatnya. Untuk kabel USB to Serial Prolific, posisi kabel *ground* terpasang pada bagian bawah seperti terlihat pada Gambar 4.23b.

Koordinator piranti XBee pada AP terdiri atas *XBee 802.15.4 Radios (Series 1)* dan *XBee Explorer USB Board* yang tersambung pada AP dengan kabel mini-USB seperti dapat dilihat pada Gambar 4.24.

(a) Koordinator XBee saat sebelum dirakit.

(b) Koordinator XBee saat telah dirakit.

Gambar 4.24: Koordinator piranti XBee yang akan disambungkan pada AP.

Merakit koordinator piranti XBee termasuk mudah, karena hanya perlu memasang *XBee 802.15.4 Radios (Series 1)* pada *XBee Explorer USB Board*, kemudian menghubungkan keduanya ke AP, seperti yang dapat dilihat pada Gambar 4.24b.

AP yang digunakan adalah TP-LINK MR3020 dengan USB Hub, seperti yang dapat dilihat pada Gambar 4.25a. USB Hub digunakan untuk menyambungkan USB *flash drive*, koordinator IQRF, dan koordinator XBee.

(a) AP saat sebelum dirakit.

(b) AP saat telah dirakit.

Gambar 4.25: AP TP-LINK MR3020.

Urutan piranti yang terpasang ke AP, dari atas ke bawah, adalah USB *flash drive*, koordinator XBee, dan koordinator IQRF. Posisi tersebut tidak boleh terbalik karena tidak sesuai dengan konfigurasi aplikasi yang sudah dirancang dan kembangkan. Sehingga kondisi saat semua piranti sudah dipasangkan dapat dilihat pada Gambar 4.25b.

Setelah AP sudah siap, langkah selanjutnya adalah perakitan sensor IQRF dan piranti XBee. Sensor-sensor IQRF terdiri atas TR-52B dan kit pengembangan DK-EVAL-03 seperti dapat dilihat pada Gambar 4.26.

(a) Sebelum dirakit.

(b) Setelah dirakit.

Gambar 4.26: Sensor IQRF.

Untuk merakit sensor IQRF, masukkan TR-52B ke dalam kit pengembangan DK-EVAL-03 pada *slot* yang sudah disediakan. Kondisi setelah dipasangkan dapat dilihat pada Gambar 4.26b.

Sedangkan piranti XBee relay terdiri atas tiga bagian, yaitu *XBee 802.15.4 Radios (Series 1)*, *2 channel Relay Shield For Arduino (With XBee/BTBee interface)*, dan Arduino Uno yang tersambung ke sumber listrik dengan kabel USB seperti dapat dilihat pada Gambar 4.27.

(a) Sebelum dirakit.

(b) Setelah dirakit.

Gambar 4.27: Piranti XBee.

Piranti XBee akan tersusun dalam tiga lapis rangkaian, mulai dari bawah, Arduino Uno, *2 channel Relay Shield For Arduino (With XBee/BTBee interface)*, dan *XBee 802.15.4 Radios (Series 1)*. Hasil rakitan piranti XBee dapat dilihat pada Gambar 4.27b.

4.3.2 Hasil Uji Coba Aplikasi

Uji coba yang dilakukan menggunakan satu set AP lengkap yang sudah dilengkapi dengan koordinator IQRF dan XBee. Sensor yang digunakan adalah dua buah IQRF dan satu buah XBee.

Uji coba pertama dilakukan dengan penambahan XBee Relay baru via aplikasi web dan menyalakan relay 1 dengan ponsel cerdas seperti dapat dilihat pada Gambar

4.28. Pada Gambar 4.28 dapat dilihat bahwa relay 1 dalam keadaan menyala pada layar ponsel cerdas dan XBee Relay.

Gambar 4.28: Uji coba aplikasi dengan ponsel cerdas.

Pada saat pengguna meng-klik salah satu saklar, akan tampak *progress* dari proses menyalakan relay berupa garis tepi lingkaran berwarna hijau. Saat relay sudah dalam kondisi menyala, maka lingkaran akan mempunyai garis tepi berwarna hijau yang sempurna.

Pengujian IQRF dilakukan dengan penambahan dua sensor via aplikasi web dan kemudian melihat temperatur yang terbaca. Salah satu sensor digenggam dengan tangan untuk memanipulasi temperatur. Hasilnya dapat dilihat pada Gambar 4.29.

Gambar 4.29: Membaca temperatur yang terbaca pada sensor.

Jarum penunjuk pada masing-masing penampil akan bergerak sesuai dengan bacaan temperatur dari masing-masing sensor IQRF. Gerakan akan terjadi secara langsung, tanpa pengguna harus memuat ulang halaman web.

Pengujian yang dilakukan pada penelitian ini tidak ditujukan pada ketelitian pembacaan temperatur pada sensor IQRF karena penelitian ini hanya berfokus pada fungsionalitas sistem.

Dari hasil pengujian di atas, didapati bahwa aplikasi yang dibangun sudah dapat berfungsi sebagaimana mestinya.

4.3.3 Kelebihan dan Kekurangan Sistem

Secara garis besar, sistem yang dibangun mampu menginteroperabilitaskan dua jenis jaringan sensor nirkabel dan protokol internet dengan biaya yang rendah. Untuk lebih jelasnya, kelebihan sistem yang dibangun dijelaskan pada daftar berikut,

- mampu mengintegerasikan dua jenis WSN,
- jumlah piranti yang terpasang dapat ditambah dan dikurangi secara mudah,
- aplikasi web yang dibangun dapat diakses dari komputer maupun ponsel cerdas karena bersifat responsif,
- mendukung fitur profil guna mendukung implementasinya pada *green building* dan *smart house*,
- biaya yang dibutuhkan tergolong murah, dan
- AP berukuran kecil sehingga mudah ditempatkan di sudut ruangan.

Sedangkan kekurangan sistem yang dapat dikembangkan atau diperbaiki pada penelitian selanjutnya dijelaskan pada daftar berikut ini,

- skalabilitas yang kurang karena hanya mampu mendukung dua jenis WSN secara *default*,
- baterai IQRF *node* harus diisi ulang dalam waktu berkala,
- memori eksternal dari AP kurang stabil, baru terbaca setelah AP dinyalakan ulang, dan
- sistem tidak memiliki baterai cadangan apabila listrik mati.

4.3.4 Masalah dan Penyelesaian

Masalah yang pertamakali muncul adalah tentang extroot. TP-LINK MR3020 memiliki *bug* yang membuat memori eksternal tidak berfungsi saat *booting* pertamakali. Setelah TP-LINK MR3020 dijalankan ulang, memori eksternal baru terbaca. Penyebab dari masalah ini belum ditemukan, sehingga solusi untuk masalah ini adalah melakukan *rebooting* jika TP-LINK MR3020 tidak membaca memori eksternal. Masalah ini tidak ditemukan pada AP dengan jenis lain seperti pada penelitian yang dilakukan oleh [1].

IQRF dinilai kurang stabil, terlebih saat menancapkan kabel USB to Serial Prolific ke USB Hub pada AP saat AP sudah menyala. Jika kurang berhati-hati, AP justru akan menjadi tidak stabil. Gejala yang muncul seperti pustaka PHP yang tidak terbaca, sampai jaringan Wi-Fi yang terputus. Untuk mengantisipasi masalah ini, sebaiknya semua kabel terpasang sebelum AP dinyalakan.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil analisis dan pengujian fungsional aplikasi ini, didapat kesimpulan sebagai berikut:

1. Interoperabilitas *Wireless Sensor Network* (WSN) *multiple vendor* dengan *Internet Protocol* (IP) dapat tercapai dengan *gateway* yang ditanamkan aplikasi berbasis web yang didukung dengan aplikasi Python untuk berkomunikasi dengan WSN yang terisi dengan aplikasi C.

2. Sesuai pengujian yang dilakukan dalam skala laboratorium, aplikasi dapat berjalan baik sesuai dengan fitur-fitur yang dirancang.

3. Salah satu kendala yang dihadapi adalah tidak stabilnya AP seri TP-LINK MR3020 dalam pembacaan memori eksternal. Saat pertama kali dinyalakan, AP tidak membaca memori eksternal yang sudah dipasang. Setelah AP dinyalakan ulang, baru AP dapat membaca memori eksternal.

4. Masalah lain yang dihadapi juga masalah ketidakstabilan koordinator IQRF yang tertancap pada AP. Agar tidak terjadi kesalahan perangkat keras, sebaiknya koordinator IQRF sudah terpasang pada saat AP masih dalam kondisi mati.

5.2 Saran

1. Penelitian selanjutnya dapat menggunakan AP selain TP-LINK MR3020 sebagai gateway dan membandingkannya dengan performa TP-LINK MR3020 yang memiliki *bug* pada extroot.

2. Penelitian ini menggunakan Python untuk berkomunikasi dengan kanal serial, penelitian selanjutnya dapat menggunakan pendekatan yang lain, seperti membangun aplikasi dengan bahasa C.

3. Penelitian selanjutnya dapat menggunakan jenis-jenis WSN yang lebih bervariatif guna interoperabilitas yang lebih luas.

4. Skalabilitas sistem dapat menjadi pertimbangan lebih lanjut dalam penelitian selanjutnya terlebih guna mendukung interoperabilitas berbagai jenis WSN yang lebih luas.

DAFTAR PUSTAKA

[1] S. B. Wibowo and Widyawan, "Wireless Sensor Network and Internet Protocol Integration with COTS," in *2013 AUN/SEED-Net Regional Conference in Electrical and Electronics Engineering*. Department of EE, Chulalongkorn University, 2013, pp. 120–123.

[2] M.-E. Raluca, M.-E. Razvan, and A. Terzis, "Gateway design for data gathering sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*. IEEE, 2008, pp. 296–304.

[3] T. Matsuura, K. Hisazumi, T. Kitasuka, T. Nakanishi, and A. Fukuda, "UDSS: Sensor Device for Context Awareness in Home Network," *2007 Fourth International Conference on Networked Sensing Systems*, pp. 196–200, Jun. 2007. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4297419

[4] H.-C. Lin, Y.-C. Kan, and Y.-M. Hong, "The comprehensive gateway model for diverse environmental monitoring upon wireless sensor network," *Sensors Journal, IEEE*, vol. 11, no. 5, pp. 1293–1303, 2011.

[5] K.-i. Hwang, J. In, N. Park, and D.-s. Eom, "A design and implementation of wireless sensor gateway for efficient querying and managing through world wide web," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1090–1097, Nov. 2003. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1261201

[6] J. J. P. C. Rodrigues and P. A. C. S. Neves, "A survey on IP-based wireless sensor network solutions," no. January, pp. 963–981, 2010.

[7] B. da Silva Campos, J. J. P. C. Rodrigues, L. D. P. Mendes, E. F. Nakamura, and C. M. S. Figueiredo, "Design and construction of wireless sensor network gateway with IPv4/IPv6 support," in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[8] R. Spinar, P. Muthukumaran, R. de Paz, D. Pesch, W. Song, S. A. Chaudhry, C. J. Sreenan, E. Jafer, B. O'Flynn, and J. O'Donnell, "Demo Abstract: Effi-

cient Building Management with IP-based Wireless Sensor Network," in *6th European Conference on Wireless Sensor Networks, Cork*, 2009, pp. 1–2.

[9] A. Dunkels, T. Voigt, N. Bergman, and M. Jönsson, "The design and implementation of an IP-based sensor network for intrusion monitoring," in *Swedish National Computer Networking Workshop*. Citeseer, 2004.

[10] Widyawan, S. B. Wibowo, M. I. Zul, and B. Nugroho, "iHome: Low-Cost Domotic for Residential Houses," in *5th AUN/SEED-Net Regional Conference on Information and Communications Technology (RCICT)*. AUN/SEED-Net, 2012.

[11] N. Erratt and Y. Liang, "The design and implementation of a general WSN gateway for data collection," pp. 4392–4397, 2013.

[12] MICRORISC, "IQRF," 2013, [Online]. Available: https://iqrf.org/. [Accessed: March 13, 2013].

[13] V. Sulc, "Home Automation with IQRF Wireless Communication Platform : A Case Study," no. c, pp. 212–217, 2011.

[14] P. Seflova, V. Sulc, J. Pos, and R. Spinar, "IQRF wireless technology utilizing IQMESH protocol," in *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on*. IEEE, 2012, pp. 101–104.

[15] Digi International Inc., "ZigBee® Wireless Standard," 2013, [Online]. Available: https://www.digi.com/technology/rf-articles/wireless-zigbee. [Accessed: March 13, 2013].

[16] OpenWrt Project, "OpenWRT," 2013, [Online]. Available: https://openwrt.org/. [Accessed: March 13, 2013].

[17] V. Gabriel, "Bootstrap Admin Theme," 2013, [Online]. Available: http://vinceg.github.io/Bootstrap-Admin-Theme/. [Accessed: December 29, 2013].