

DESAIN OVER THE AIR PROGRAMMING PADA JARINGAN SENSOR NIRKABEL BERBASIS PLATFORM ARDUINO

Nanan Setiadi¹, Mohammad Fajar^{2*}, Musfirah Putri Lukman³

^{1,2} Teknik Informatika, STMIK Kharisma Makassar

³ Teknik Elektro, Politeknik Negeri Ujung Pandang

e-mail: ¹nanan_14@kharisma.ac.id, ²fajar@kharisma.ac.id, ³musfirahputrilukman@poliupg.ac.id

Abstrak

Penelitian ini bertujuan untuk merancang mekanisme pemrograman ulang node sensor melalui *over the air programming* (OTAP) atau dari jarak jauh pada jaringan sensor nirkabel. Pengumpulan data dilakukan melalui uji coba sistem usulan di lingkungan terbatas. Pada rancangan, perangkat Arduino yang menjadi platform target pemrograman ulang dimodifikasi bootloadernya agar mampu mengeksekusi program tertanam yang disimpan dalam SD Card. Kode sumber Arduino berekstensi *.ino* dikompilasi menjadi file *.hex* menggunakan Arduino IDE yang kemudian diproses kembali menggunakan program *avrdude* untuk menghasilkan file *.bin* (*firmware*) yang siap dijalankan. Untuk melengkapi mekanisme rancangan dikembangkan aplikasi desktop yang dapat mengendalikan perangkat komunikasi XBee pengirim dalam pengiriman data berupa paket per paket. Jika pengiriman kode berhasil dilakukan, node target melakukan operasi *restart*, menjalankan bootloader untuk memuat kode baru dari SD Card dan menjalankan kode baru tersebut. Evaluasi sistem dilakukan melalui beberapa skenario pengujian pada jarak yang berbeda, baik itu di area tertutup (*indoor*) maupun terbuka (*outdoor*). Hasil evaluasi menunjukkan mekanisme OTAP yang dirancang pada lingkungan *outdoor* untuk jarak pengujian hingga 15 meter diperoleh kecepatan transmisi mencapai 35 byte per detik untuk ukuran paket pengiriman sebesar 60 byte. Selain itu, ukuran paket tanpa *packet loss* pada saat pengiriman yaitu tidak melebihi 30 byte untuk jarak pengiriman hingga 15 meter. Kecepatan pengiriman dari seluruh pengujian antara 14 byte per detik hingga 35 byte per detik. Untuk pengujian *indoor* hasil evaluasi memperlihatkan belum efektifnya mekanisme OTAP yang dirancang dalam transmisi kode program dari node sumber ke node target, hal ini diasumsikan dipengaruhi oleh kemampuan transmisi modul komunikasi XBee yang digunakan.

Kata kunci: *Over the Air Programming, Jaringan Sensor Nirkabel, Arduino, Sistem Monitoring, Pemrograman Jarak Jauh*

Abstract

The purpose of this research is to design *over the air programming* mechanism for a node in wireless sensor network. Data are gathered from experiments of the proposed system on a constrained environment. On this design, the Arduino that becomes the target of reprogramming has its bootloader modified to enable the capability to run embedded program from the SD Card. Arduino's source code *.ino* file is compiled into a *.hex* file using the Arduino IDE. Then, it is processed using *avrdude* application into a ready to run *firmware* binary file on Arduino. A desktop application is made for this mechanism in order to control sender XBee device in transmitting data packet by packet. If the transmission was successful, the node will restart itself, and trigger the bootloader to load new program from the SD Card. The system is evaluated through preset scenarios which including testing at various distances and environments (*indoor* and *outdoor*). It was concluded that the reprogramming mechanism was ineffective in transmitting source code when at least one of the XBee devices is located inside the building which affected the

* Corresponding author : Mohammad Fajar (fajar@kharisma.ac.id)

transmission process. For the second testing scenario, both nodes were located outdoors with testing distance up to 15 metres, resulting with average transmission speed of 35 bytes per second for sent packet size of 60 bytes. The safest sent packet size to avoid packet loss while transmitting should not exceed 30 bytes for transmission distance up to 15 metres. Transmission speed across all test scenarios ranges between 14 bytes per second to 35 bytes per second. For indoor testing, the evaluation results showed that the designed OTAP mechanism in this study is not yet effective. This is assumed to be influenced by the transmission capability of the XBee communication module.

Keywords: *Over The Air Programming, Wireless Sensor Network, Arduino, Monitoring System, Remote Programming*

1. Pendahuluan

Jaringan sensor nirkabel merupakan sebuah jaringan yang terdiri dari sejumlah node sensor yang disebar di lingkungan tertentu untuk melakukan pengindraan kondisi lingkungan atau objek dan mengirim hasil pengindraannya ke pusat penyimpanan. Data pengindraan kemudian diolah dan dianalisis menjadi informasi yang disajikan ke pengguna. Jaringan sensor nirkabel merupakan teknologi penting dan pendukung utama pada sistem Internet of Things (IoT) [1]. Node-node pada jaringan sensor nirkabel dapat disebar di lokasi yang cukup jauh dan sulit dijangkau, lokasi yang belum memiliki infrastruktur energi yang memadai ataupun lokasi yang cukup berbahaya bagi manusia. Seperti jaringan sensor nirkabel untuk memantau aktivitas gunung berapi [2] [3], jaringan sensor nirkabel untuk memantau kebakaran hutan [4], pemakaian node sensor dan kamera untuk pemantauan alam liar di hutan tropis [5], dan sistem monitoring di fasilitas nuklir [6].

Salah satu tantangan pada sistem jaringan sensor nirkabel dan Internet of Things (IoT) yaitu perawatan kode aplikasi yang menggerakkan aktivitas node-node sensor tersebut. Jika kode program pada node perlu diprogram ulang untuk menjalankan instruksi yang baru sesuai kebutuhan atau karena adanya kegagalan (bug) pada kode termasuk masalah keamanan, maka tentunya datang ke lokasi dimana node ditempatkan untuk diprogram ulang merupakan hal yang sulit dilakukan. Terlebih jika lokasi node sensor tidak memungkinkan untuk dijangkau. Oleh karena itu, salah satu solusi yang dapat ditempuh yaitu memprogram ulang node sensor dari jarak jauh (remote) tanpa menyentuhnya secara fisik, yang dikenal dengan istilah over the air programming (OTAP) [7].

Terdapat sejumlah studi dalam literatur yang membahas prinsip OTAP tersebut, seperti yang diuraikan pada penelitian [8] terkait berbagai teknik pembaharuan kode firmware yang dapat digunakan. Penelitian [9] mengusulkan mekanisme OTAP bagi perangkat atau sistem berbasis IoT yang tidak memiliki layanan update kode secara bawaan. Demikian pula studi yang dilakukan oleh [10], merancang sebuah sistem yang memungkinkan pembaharuan program pada mikrokontroler AVR ATmega32 menggunakan modem GPRS. Sistem yang dirancang melibatkan sebuah server OTA yang berfungsi mendorong pembaharuan menuju perangkat melalui komunikasi protokol TCP/IP dari modem GPRS, dan pembaharuan tersebut kemudian disimpan

ke dalam EEPROM. Akan tetapi, model protokol yang digunakan masih mengandalkan ketersediaan infrastruktur jaringan internet. Demikian pula penelitian oleh [11] yang merancang sistem pemrograman OTA pada perangkat Imote2 menggunakan protokol Deluge dengan mekanisme pengiriman paket per paket dan metode OTAP pada studi [12] untuk platform MicaZ. Mekanisme pembaharuan kode secara remote menggunakan protokol MOAP (*Multihop Over-the-Air Programming*) juga dikaji dalam penelitian [13], sementara penelitian [14] mengkaji model protokol yang aman dalam mekanisme OTAP di jaringan sensor nirkabel. Selain itu, studi [15] merancang sistem pemrograman *Over The Air* pada jaringan sensor nirkabel menggunakan mote berbasis PsoC (*Programmable System on Chip*) yang memungkinkan banyaknya pilihan pengaturan antarmuka pada berbagai macam sensor seperti pengaturan kuat sinyal dan pengaturan resistansi arus yang digunakan masing-masing sensor berbasis perangkat CPLD. Meskipun untuk merancang antarmuka PsoC diperlukan sejumlah perangkat keras tambahan dengan tingkat kompleksitas pengembangan atau konfigurasi yang cukup tinggi.

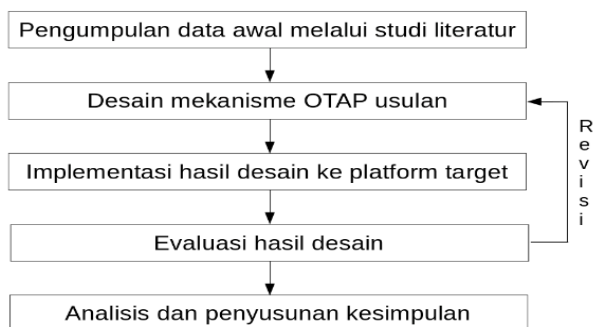
Oleh karena itu, penelitian ini bertujuan untuk merancang mekanisme over the air programming node sensor menggunakan teknik yang lebih sederhana tanpa melibatkan protokol komunikasi yang kompleks. Selain itu rancangan pada penelitian ini menggunakan platform Arduino yang berbiaya murah dan mudah digunakan dikalangan pengguna jaringan sensor nirkabel atau sistem Internet of Things. Penelitian ini diharapkan memberikan wawasan dan informasi penting bagi pengguna jaringan sensor dan IoT terkait pengembangan mekanisme OTAP.

2. Metodologi

Pada bagian ini dijelaskan terkait tahapan penelitian, over the air programming, evaluasi protocol jaringan, dan acuan pengukuran kualitas yang digunakan.

A. Tahapan Penelitian

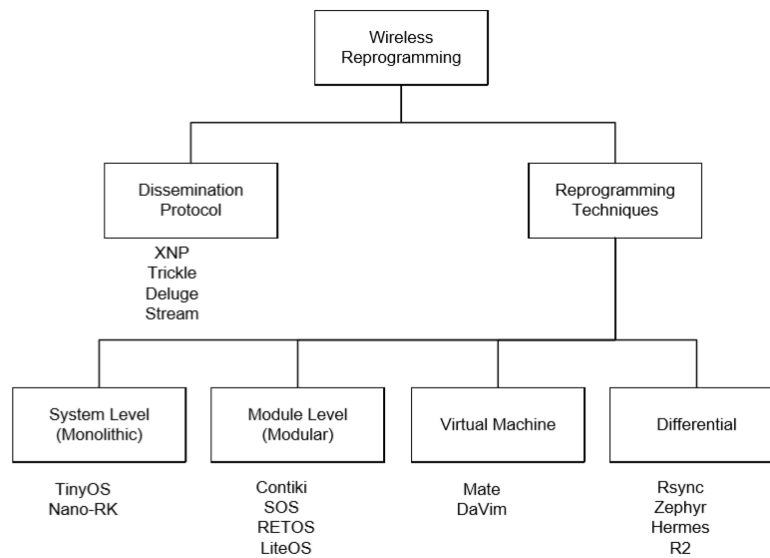
Penelitian diawali dengan melakukan kajian terhadap sejumlah literatur terkait, untuk mendefinisikan mekanisme OTAP dan berbagai teknik yang telah digunakan. Selanjutnya dilakukan analisis dan desain mekanisme OTAP yang diusulkan dalam penelitian ini. Analisis kebutuhan spesifikasi mekanisme OTAP divisualisasikan menggunakan diagram use-case UML. Untuk desain, digambarkan arsitektur sistem, desain perangkat dan desain aplikasi yang digunakan. Tahapan berikutnya yaitu mengimplementasikan hasil desain ke platform target, dalam penelitian ini yaitu Arduino dan modul komunikasi berbasis Xbee. Kemudian dilakukan evaluasi melalui berbagai skenario dan kondisi lingkungan yang terbatas, baik itu di luar ruangan atau gedung (outdoor) maupun di dalam gedung (indoor). Hasil evaluasi dianalisis dan dijadikan dasar penyusunan kesimpulan penelitian. Gambar 1 memperlihatkan secara umum tahapan yang dilakukan dalam penelitian ini.



Gambar 1. Tahapan Penelitian

B. Over The Air Programming

OTA Programming atau Over-the-Air Programming merupakan metode update (pembaharuan) program melalui media nir kabel (wireless) jarak jauh, misalnya dengan menggunakan GPRS, Wi-Fi dan sejenisnya. Kelengkapan yang harus dimiliki untuk bisa melakukan update program secara OTA antara lain: kemampuan koneksi ke jaringan atau internet, kemampuan download data atau program dari server OTA, memori untuk menampung data atau program selama proses download, dan kemampuan self flashing (pemrograman sendiri). Pemrograman OTA memerlukan bootloader yang secara penuh dapat menjalankan fungsi OTA [9]. *Over-the-air programming* dipandang sebagai metode pemrograman kode komputer pada sebuah platform remote yang memiliki komputer lokal lengkap dengan memori flash yang melibatkan proses menerima beragam paket kode komputer melalui transmisi nirkabel dimana selanjutnya akan disimpan dalam suatu area pada memori flash komputer lokal tersebut. Komputer lokal selanjutnya mengecek integritas kode yang diterima sebelum proses reboot. Sebagai respon pengecekan, alamat boot dari komputer lokal tersebut diganti sehingga komputer tersebut dapat reboot setelah perubahan terjadi. Pasca reboot, komputer akan direstart dengan kode baru dan tidak memerlukan pengecekan integritas lagi [15]. Gambar 2 menggambarkan secara garis besar sejumlah klasifikasi dari protokol standar dalam *Over-the-air Programming*.



Gambar 2. Bagan Klasifikasi Protokol Pemrograman Ulang Jaringan Sensor Nirkabel [16]

Sifat protokol OTA programming yang ideal, khususnya untuk jaringan sensor nirkabel antara lain: (a) kompleksitas waktu dan ruang algoritma pemrograman ulang harus disesuaikan dengan kemampuan dari sebuah node sensor; (b) pemrograman ulang harus efisien dalam penggunaan energi dimana diketahui bahwa penulisan ulang ke EEPROM atau sejenisnya memakan banyak energi dan terus-menerus menunggu update datang juga mengonsumsi energi dan sebaiknya dihindari; (c) kode yang dikirim harus terkirim secara keseluruhan dikarenakan banyak kemungkinan gangguan dari komunikasi yang menyebabkan kerusakan data; (d) skalabilitas dari struktur jaringan sensor berdasarkan jumlah node sensor atau jarak antar node [18].

C. Evaluasi Protokol Pemrograman Jaringan

Terdapat sejumlah parameter untuk mengevaluasi kualitas sebuah protokol pemrograman jaringan yang didasarkan pada prinsip protokol Deluge. Pertama, Keandalan dalam Pengiriman Data, dimana semua data yang dikirim dalam jaringan sensor nirkabel harus diterima sepenuhnya dengan tepat oleh semua node. Selanjutnya, terdapat waktu pengiriman, lalu disusul oleh penggunaan RAM, dan konsumsi energi [11].

Dari keempat faktor di atas, dua parameter pertama yang telah disebut itu diangkat sebagai fokus penelitian, yakni keandalan pengiriman dan waktu pengiriman. Dikarenakan tujuan semula dari perancangan mekanisme ini adalah untuk membuktikan bahwa mekanisme ini dapat memenuhi tujuannya dan adanya perhitungan waktu pengiriman menjadi faktor utama pertimbangan keefektifan mekanisme. Selain itu, dalam mekanisme ini pengujian kedua faktor yang telah disebutkan di atas, akan diamati performanya relatif terhadap jarak antara node dengan pengirim.

D. Acuan Pengukuran Kualitas Pengiriman

Untuk melakukan pengiriman file menggunakan XBee dari satu Arduino ke Arduino lain, ukuran *baud rate* yang dapat tergolong aman adalah 50% - 60% dari ukuran *RF rate* dimana jika melebihi nilai tersebut akan menimbulkan risiko *data overflow* pada saat pengiriman dalam komunikasi serial dimana data dapat hilang [19]. Pengujian pengiriman tersebut menggunakan Arduino Mega dengan ukuran *serial buffer* sebesar 256 byte, dengan rerata ukuran paket pengiriman sebesar 240 byte. Hasil pengujian menghasilkan sejumlah hasil dimana banyak terjadi *data loss*. Rerata waktu yang diperoleh dalam proses pengiriman adalah 1 Kilobyte per detik.

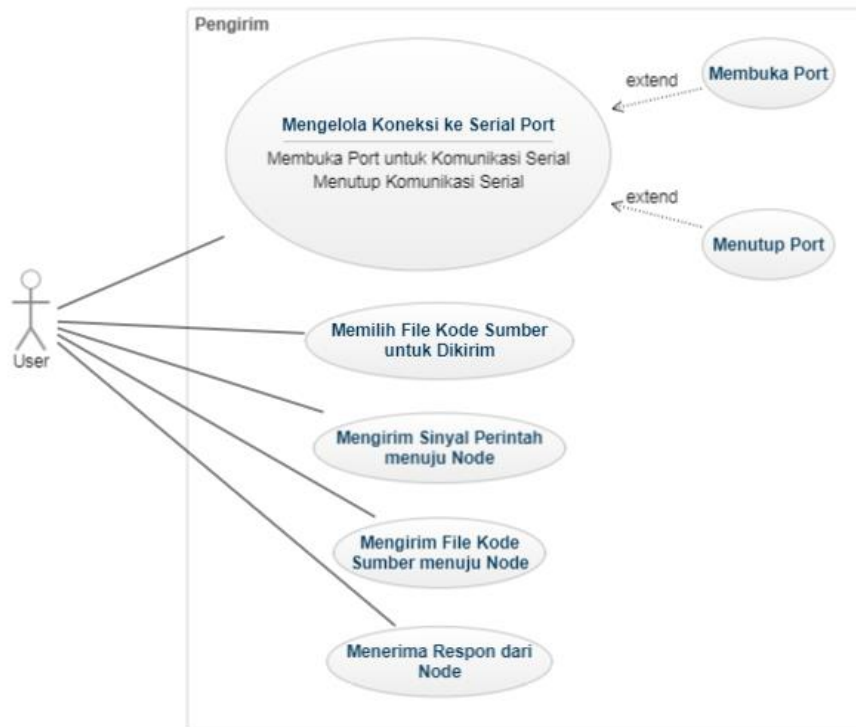
Pengujian pada [8] berupa pengiriman hingga 12.500 paket data (10.000 ribu berupa paket sinyal ACK dan CMD, 2.500 paket data asli) menggunakan XBee menuju Arduino Uno R3 dalam lingkungan *indoor* dengan jarak pengujian mencapai 31,5 meter. dimana kecepatan pengiriman yang diperoleh berkisar antara 6,64 Kbps hingga 7,36 Kbps (830 byte per detik hingga 920 byte per detik). Berdasarkan acuan-acuan tersebut, referensi nilai tertinggi yaitu 1 Kilobyte per detik digunakan sebagai nilai acuan untuk mengukur kualitas kecepatan harapan pengiriman data dalam mekanisme ini.

3. Hasil dan Pembahasan

Bagian ini membahas rancangan sistem pemrograman ulang yang meliputi analisis kebutuhan yang disajikan secara visual menggunakan diagram use-case, arsitektur sistem dan rancangan prototip perangkat serta antarmuka aplikasi.

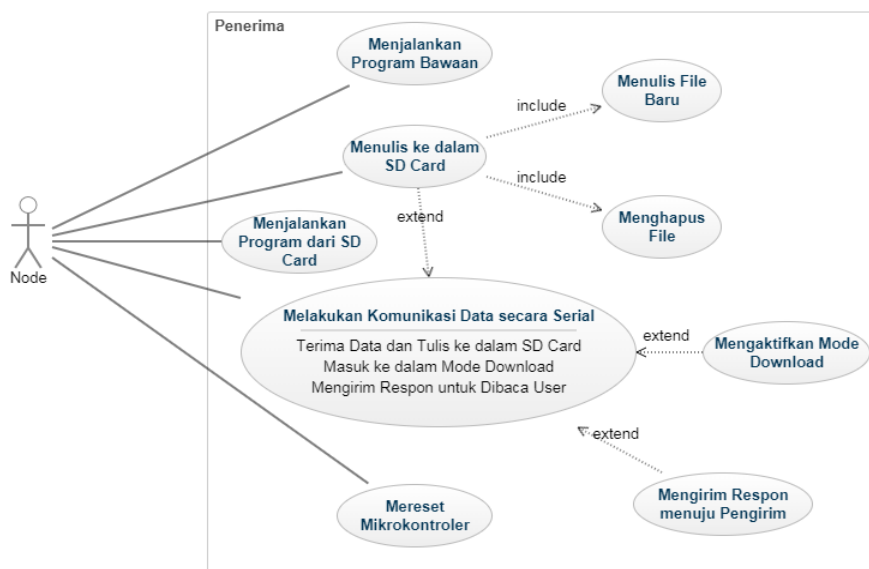
A. Analisis Sistem

Sistem pemrograman ulang dalam penelitian ini melibatkan dua aktor yaitu user atau pemrogram yang melakukan perubahan kode program dan node sensor target yang akan diubah kodenya. Fungsional utama yang dibutuhkan oleh user yaitu mengelola koneksi ke node-node sensor, memilih dan mengirim kode sumber yang akan dikirim, dan membaca respons dari node sensor. Gambar 3 memperlihatkan analisis kebutuhan sistem pada sisi pemrograman.



Gambar 3. Use Case Diagram Dengan Aktor User (Pemrogram)

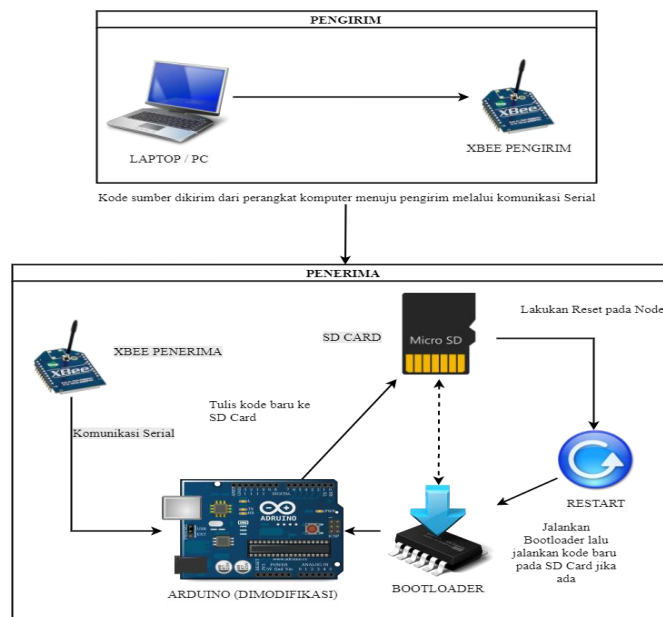
Untuk node sensor, fungsional utama yang dibutuhkan yaitu menerima dan menulis data (kode program baru) ke penyimpanan (SD card), mengaktifkan mode download, menjalankan program, melakukan reset node dan mengirim respon ke user. Gambar 4 menyajikan hasil analisis kebutuhan sistem pada sisi node sensor atau penerima.



Gambar 4. Use Case Diagram dengan Aktor Node Sensor (Penerima)

B. Desain Arsitektur Sistem

Pada rancangan sistem, pengirim terdiri dari perangkat komputer yang dipasangkan dengan sebuah modul komunikasi XBee yang digunakan dalam penelitian ini yang diprogram dengan *firmware ZNet 2.5 Coordinator AT*. Penerima terdiri dari sistem Arduino Uno R3 yang juga dilengkapi *XBee Shield* sebagai media bantu pemasangan XBee dan sebuah *MicroSD Card Adapter* untuk menampung *SD Card* yang berfungsi sebagai media penyimpanan kode sumber yang dikirim. XBee penerima diprogram dengan *firmware ZNet 2.5 Router/End Device AT*. Gambar 5 menyajikan rancangan arsitektur sistem yang dibuat.

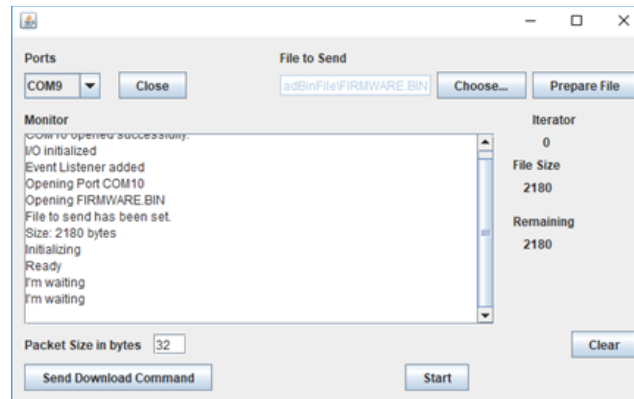


Gambar 5. Arsitektur Rancangan Sistem

Kemudian *bootloader* dari Arduino diganti dengan *bootloader AVR Boot* dari yang memungkinkan Arduino dapat menjalankan program baru dari *SD Card* pada saat proses *booting*.

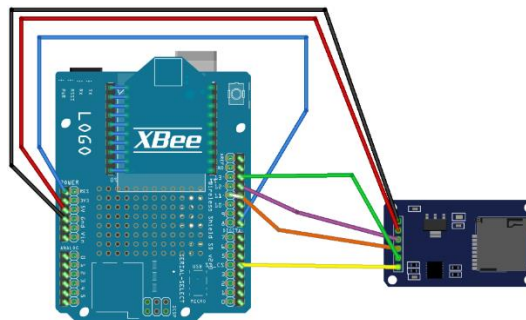
C. Desain Aplikasi dan Prototip

Sebuah aplikasi pengirim dibuat menggunakan bahasa pemrograman Java seperti yang diperlihatkan pada Gambar 6. Kode sumber Node sensor yaitu Arduino terlebih dahulu diolah menjadi file *firmware* berekstensi *.bin*. Selanjutnya, kode sumber dikirim dalam bentuk paket per paket. Node sensor menerima satu paket, menulis hingga selesai lalu mengirim respon kembali ke pengirim agar mengirim paket selanjutnya. Hal ini dilakukan hingga keseluruhan file terkirim dan berhasil diterima. Selama proses pengiriman, Node sensor hanya mengeksekusi perintah untuk menerima data dan menulis file.



Gambar 6. Aplikasi Pengirim

Perangkat Arduino dirangkai sebagaimana yang diperlihatkan pada Gambar 7 dimana *MicroSD Card Adapter* dihubungkan dengan Arduino menuju pin MISO, MOSI, dan SCK.



Gambar 7. Skema Dasar Arduino dan MicroSD

D. Evaluasi

Evaluasi dilakukan melalui pengujian *black box* yaitu menguji apakah mekanisme pemrograman ulang berhasil atau tidak dengan melibatkan sejumlah skenario. Pengujian awal menggunakan ukuran paket 60 byte dan jarak pengujian kurang dari 1 meter. Dikarenakan ukuran *serial buffer* dari Arduino Uno adalah 64 byte [4], untuk menghindari *overflow* dan *data loss* maka ukuran paket harus kurang dari 64 byte. Dalam pengujian digunakan dua file *firmware* sebagai file yang akan dikirim. Perhitungan kecepatan transmisi dihitung dari ukuran file dibagi waktu pengiriman dalam bentuk byte per detik.

File pertama merupakan file kode sumber *Blink* bawaan dari perangkat lunak Arduino IDE, dengan ukuran file 2.180 byte. File kedua merupakan file yang berisi kode sumber program untuk membaca data suhu dari sensor DS18B20 yang dilengkapi dengan *library* tambahan, berukuran 5.696 byte. Hasil pengujian disajikan pada Tabel 1.

Tabel 1. Hasil Pengujian pertama untuk (a) file “Blink” dan (b) pembacaan sensor DS18B20

(a) Blink		(b) DS18B20	
Percobaan	Waktu Pengiriman	Percobaan	Waktu Pengiriman
1	60,88 detik	1	161,16 detik
2	60,94 detik	2	159,65 detik
3	60,78 detik	3	157,92 detik
4	60,80 detik	4	160,20 detik
5	60,98 detik	5	159,70 detik
Rerata Waktu Pengiriman	60,87 detik	Rerata Waktu Pengiriman	159,72 detik

Pengujian menunjukkan bahwa kecepatan rata-rata pengiriman yaitu sekitar 35 byte per detik untuk kedua file tersebut. Untuk node dengan pemrograman ulang kode sensor DS18B20 pada Arduino berhasil dioperasikan dan mampu mengirim hasil pembacaan sensor menuju node penerima. Selanjutnya, untuk mengetahui performansi pengiriman dan penerimaan data, dilakukan pengujian kedua dengan tiga skenario pengujian lanjutan yakni: (1) pengiriman *indoor-indoor*; (2) pengiriman *indoor* (pengirim) – *outdoor* (penerima); (3) pengiriman *outdoor-outdoor*. Untuk skenario pertama, dibagi menjadi 2 subskenario. Pertama, penerima diletakkan di dalam ruangan kelas, dengan jarak kurang lebih 5 meter dari pintu masuk kelas. Pada percobaan pertama pengirim diletakkan di ruang lobi kampus dengan jarak kurang lebih 5 meter dari ruang sebelumnya. Kedua Node pada pengujian subskenario ini gagal membuat komunikasi yang diasumsikan terhalang oleh dinding ruangan. Kedua, penerima tetap di lokasi yang sama, sedangkan pengirim diletakkan 1 meter di depan pintu masuk ruang pertama, di luar ruangan. Hasil pengujian ditunjukkan oleh Tabel 2.

Tabel 2. Hasil Pengujian kedua untuk (a) file “Blink” dan (b) pembacaan sensor DS18B20

(a) Blink			(b) Sensor DS18B20		
Percobaan	Ukuran Paket (byte)	Hasil	Percobaan	Ukuran Paket (byte)	Hasil
1	60	Sukses dalam 60,21 detik	1	60	Gagal (<i>Data loss</i> setelah 182,14 detik)
2	60	Sukses dalam 62,67 detik	2	48	Gagal (<i>Data loss</i> setelah 186,62 detik)
3	60	Sukses dalam 61,74 detik	3	36	Gagal (<i>Data loss</i> setelah 31,29 detik)

4	60	Sukses dalam 61,69 detik
Rerata Waktu Pengiriman		61,5775 detik

4	24	Sukses dalam 394,88 detik
---	----	------------------------------

Dari evaluasi diketahui bahwa pengiriman pada skenario *indoor-indoor* mengalami kendala berupa *data loss* saat mengirim file dengan ukuran yang relatif besar (dalam konteks percobaan, dengan ukuran lebih dari 2 KB) sehingga pada iterasi percobaan selanjutnya ukuran paket pengiriman diturunkan. Kecepatan transmisi masih sama dengan hasil pengujian untuk kasus kode "Blink" dan untuk pengujian kode "DS18B20" diperoleh 14 byte per detik.

Untuk skenario kedua, dibagi pula menjadi dua subskenario. Subskenario pertama melibatkan pengirim diletakkan di sebuah ruangan di dekat jendela, dengan ketinggian kurang lebih 10 meter dari permukaan tanah dan penerima yang diletakkan di lokasi parkir kampus berjarak horizontal kurang lebih 2 meter dari pengirim. Subskenario kedua melibatkan pengirim berada di lobi kampus, sejauh 3 meter dari pintu keluar kampus dan penerima berada 10 meter di depan pintu masuk kampus. Hasil pengujian dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian ketiga untuk (a) file "Blink" dan (b) pembacaan sensor DS18B20

(a) Blink

Percobaan	Ukuran Paket (byte)	Hasil
1	60	Gagal (Data loss setelah 68,42 detik)
2	48	Gagal (Data loss setelah 91,08 detik)
3	42	Gagal (Data loss setelah 102,22 detik)
4	30	Sukses dalam 124,59 detik

(b) Sensor suhu DS18B20

Percobaan	Kode	Ukuran Paket (byte)	Hasil
1	Blink	60	Gagal (Data loss setelah 70,34 detik)
2	Blink	48	Gagal (Data loss setelah 68,72 detik)
3	Blink	36	Gagal (Data loss setelah 87,29 detik)
4	Blink	24	Sukses dalam 154,63 detik
5	DS18B20	24	Sukses dalam 399,72 detik

Dari hasil pengujian skenario kedua, terlihat bahwa pengaruh keberadaan penghalang sinyal seperti dinding gedung mengurangi keandalan pengiriman dan meningkatkan waktu transmisi dimana rata-rata waktu transmisi sebesar 17 byte per detik untuk subskenario 1 dan 14 byte per detik untuk subskenario 2.

Untuk skenario ketiga di ruang terbuka (*outdoor*), pengirim diletakkan 2 meter dari pintu masuk kampus dan penerima diletakkan 15 meter dari pengirim. Hasil evaluasi disajikan pada Tabel 4.

Tabel 4. Hasil Pengujian Skenario Ketiga

Percobaan	Kode	Ukuran Paket (byte)	Hasil
1	Blink	60	Sukses dalam 62,78 detik
2	DS18B20	60	Sukses dalam 176,06 detik

Dari hasil pengujian skenario ketiga, terlihat bahwa pengiriman di lingkungan *outdoor* untuk jarak mencapai 15 meter untuk masing-masing file tidak mengalami kendala signifikan dengan rata-rata transmisi pengiriman antara 32 – 34 byte per detik untuk paket berukuran 60 byte.

4. KESIMPULAN

Berdasarkan evaluasi yang dilakukan, disimpulkan bahwa desain sistem pemrograman ulang node sensor secara remote berhasil melakukan pengiriman dan pemrograman ulang kode sumber pada Arduino menggunakan protokol XBee. Rancangan arsitektur untuk pengiriman kode sumber menuju node target secara nirkabel (*Over-the-air programming*) menggunakan mekanisme pengiriman yang lebih sederhana tanpa kompleksitas protokol standar OTA, namun tetap menerapkan prinsip dasar pengiriman paket per paket. Modifikasi pada *bootloader* memungkinkan Arduino menjalankan *firmware* langsung dari *SD Card*, yang juga mengurangi tingkat kompleksitas dari mekanisme rancangan. Sebagai evaluasi awal, pengujian dilakukan di lingkungan terbatas baik indoor dan outdoor dengan jarak yang bervariasi hingga 15 meter. Pengujian pada skenario *indoor-indoor* dan *indoor-outdoor* tetap berhasil dengan angka transmisi rata-rata diperoleh sebesar 35 byte ke bawah. Pengujian pada skenario *outdoor-outdoor* berhasil tanpa kendala dengan angka kecepatan transmisi antara 32 – 35 byte per detik. Berdasarkan acuan dari studi-studi terkait, kinerja mekanisme OTAP pada studi ini dari aspek kecepatan transmisinya masih sangat jauh di bawah rata-rata nilai ekspektasi yakni 1 Kilobyte per detik. Salah satu faktor yang dianggap mempengaruhi yaitu platform perangkat yang digunakan, secara khusus modul komunikasi XBee. Sehingga pekerjaan selanjutnya yang perlu dipertimbangkan yaitu melakukan modifikasi pada algoritma pengiriman, melakukan percobaan pada tipe Arduino yang lebih handal seperti Arduino Mega serta pemakaian modul komunikasi wireless yang mendukung komunikasi jarak jauh. Selain itu juga pengembangan selanjutnya perlu mempertimbangkan mekanisme mode pengiriman *one-to-one* menjadi *one-to-many*.

DAFTAR PUSTAKA

- [1] K. Gulati, R. S. Kumar Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, "A review paper on wireless sensor network techniques in Internet of Things (IoT)," in *Materials Today: Proceedings*, Elsevier Ltd, 2021, pp. 161–165. doi: 10.1016/j.matpr.2021.05.067.
- [2] J. E. Lorenzo, ; Fransiska, and S. Mukti, "Performance Evaluation of Wireless PERFORMANCE EVALUATION OF WIRELESS SENSOR NETWORK ROUTING PROTOCOL FOR VOLCANO ACTIVITY MONITORING," vol. 18, no. 1, 2022, doi: 10.33480/pilar.v18i1.3270.

- [3] V. F. Amaliya *et al.*, "Development of IoT-Based Volcano Early Warning System," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Feb. 2021. doi: 10.1088/1742-6596/1772/1/012009.
- [4] U. Dampage, L. Bandaranayake, R. Wanasinghe, K. Kottahachchi, and B. Jayasanka, "Forest fire detection system using wireless sensor networks and machine learning," *Sci Rep*, vol. 12, no. 1, p. 46, 2022, doi: 10.1038/s41598-021-03882-9.
- [5] J. A. Zwerts *et al.*, "Methods for wildlife monitoring in tropical forests: Comparing human observations, camera traps, and passive acoustic sensors," *Conserv Sci Pract*, vol. 3, no. 12, p. e568, Dec. 2021, doi: <https://doi.org/10.1111/csp2.568>.
- [6] M. K. Baek, Y. S. Chung, S. Lee, I. Kang, J. J. Ahn, and Y. H. Chung, "Design of a Nuclear Monitoring System Based on a Multi-Sensor Network and Artificial Intelligence Algorithm," *Sustainability*, vol. 15, no. 7, 2023, doi: 10.3390/su15075915.
- [7] A. S. A. Quadri and B. O. Sidek, "An Introduction to Over-the-Air Programming in Wireless Sensor Networks," vol. 2, p. 2, 2014, [Online]. Available: <http://www.ijcsns.com>
- [8] K. Arakadakis, P. Charalampidis, A. Makrogiannakis, and A. Fragkiadakis, "Firmware over-the-air programming techniques for IoT networks -- A survey," Sep. 2020, [Online]. Available: <http://arxiv.org/abs/2009.02260>
- [9] J. M. Lozano Domínguez, T. J. Mateo Sanguino, M. J. Redondo González, and I. J. Fernández De Viana González, "Over-the-Air Programming System for IoT Devices with a Non-native Wireless Update Service," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Apr. 2022. doi: 10.1088/1742-6596/2224/1/012078.
- [10] P. Eru, "Teknik Update Program Secara On-line (Over The Air, OTA) Pada Sistem Berbasis Mikrokontroler Menggunakan Modem GPRS," 2012. [Online]. Available: <https://api.semanticscholar.org/CorpusID:60632941>
- [11] R. Parthasarathy, N. Peterson, W. Song, A. Hurson, and B. A. Shirazi, "Over the Air Programming on Imote2-Based Sensor Networks," in *2010 43rd Hawaii International Conference on System Sciences*, 2010, pp. 1–9. doi: 10.1109/HICSS.2010.300.
- [12] P. and R. P. and D. P. and A. R. D. and S. M. Sangeeth K. and Pradeep, "Over the Air Programming Method for Learning Wireless Sensor Networks," in *Information Science and Applications (ICISA) 2016*, N. Kim Kuinam J. and Joukov, Ed., Singapore: Springer Singapore, 2016, pp. 555–566.
- [13] T. Stathopoulos, J. Heidemann, and D. Estrin, "A Remote Code Update Mechanism for Wireless Sensor Networks," Nov. 2003. [Online]. Available: <https://ant.isi.edu/%7ejohnh/PAPERS/Stathopoulos03b.html>
- [14] F. Doroodgar, M. A. Razzaque, and I. F. Isnin, "Seluge++: A Secure Over-the-Air Programming Scheme in Wireless Sensor Networks," *Sensors*, vol. 14, no. 3, pp. 5004–5040, 2014, doi: 10.3390/s140305004.
- [15] I. Adly, H. F. Ragai, A. El-Hennawy, and K. A. Shehata, "Over-The-Air Programming of PSoC sensor interface in wireless sensor networks," in *Melecon 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, 2010, pp. 997–1002. doi: 10.1109/MELCON.2010.5475910.