

USULAN TUGAS AKHIR

1. IDENTITAS PENGUSUL

NAMA : Muhamad Hendri Febriansyah
NRP : 05111440000036
DOSEN WALI : Victor Hariadi, S.Si, M.Kom.
DOSEN PEMBIMBING : 1. Waskitho Wibisono, S.Kom., M.Eng., Ph.D
2. -

2. JUDUL TUGAS AKHIR

“Implementasi Kompresi Adaptive Menggunakan Metode Heatshrink Untuk Pengiriman Data Pada Wireless Sensor Network Berbasis Zigbee”

3. LATAR BELAKANG

Wireless Sensor Network merupakan salah satu teknologi yang paling menjanjikan untuk kebutuhan di masa depan. Hal ini dikarenakan harganya yang murah, mudah dibangun, terdapat sensor cerdas, ukurannya kecil dan multi fungsionalitas sesuai kebutuhan. Berdasarkan research dari IDTechEx, diprediksi bahwa pasar WSN akan mengalami pertumbuhan menjadi \$ 1,8 miliar pada tahun 2024. Data ini mengacu pada WSN yang didefinisikan sebagai jaringan mesh nirkabel, yaitu self-healing dan self-organising [1].

Terdapat banyak permasalahan dan tantangan yang harus dihadapi dalam WSN untuk meningkatkan efisiensi, kelayakan dan manfaat. Tantang tersebut dapat dikategorikan kedalam empat kategori, yaitu efisiensi daya, pengumpulan data, jaringan dan strategi penyebaran [2]. WSN pada dasarnya adalah system yang berbasis event, node sensor akan mendeteksi keadaan di lingkungan sekitarnya untuk

dikirim ke sink. Namun, karena node sensor sering mendeteksi fenomena umum, maka kemungkinan ada beberapa redundansi dalam data yang sumbernya beragam berkomunikasi dengan sink tertentu. Untuk itu pemfilteran dan pemrosesan dalam jaringan diperlukan agar dapat menghemat penggunaan energi yang terbatas.

Agregasi data merupakan proses pengumpulan data dari berbagai node untuk menghilangkan redundansi, meminimalkan jumlah transmisi dan memberikan informasi yang ringkas ke simpul utama [3]. Tujuan agregasi data adalah untuk memperpanjang umur jaringan dengan mengurangi transmisi waktu atau ukuran data yang dipancarkan node menggunakan algoritma cerdas. Secara umum, agregasi data dapat dikategorikan menjadi dua subsistem yang berbeda, yaitu protocol jaringan dan penggabungan data. Gagasan protocol agregasi data pada jaringan adalah bekerjasama antar node spasial dan temporal berkorelasi [4] dalam menyebarkan data yang dikumpulkan. Pendekatan ini telah banyak diajukan dalam penelitian seperti LEECH, TEEN, HEED dan PEGASIS. Disisi lain penggabungan data bertujuan untuk mengurangi ukuran data yang ditransmisikan oleh sink atau node itu sendiri. Ukuran data dapat dikurangi menggunakan teknik operasi matematika [5] (median, average, moving average), kompresi, estimasi data dan pemodelan.

Kompresi data merupakan salah satu metode efektif untuk mengurangi penggunaan daya yang terbatas pada WSN. Diasumsikan bahwa beberapa kehilangan presisi atau kedetailan data pada saat kompresi dapat ditolerir jika hal tersebut dapat mengurangi komunikasi. Namun, perlu diperhatikan juga bahwa kualitas kedetailan data harus dipenuhi pada saat kondisi tertentu agar informasi yang didapatkan semakin banyak dan detail.

Disisi lain, menggunakan pendekatan agregasi data akan mempengaruhi perilaku komunikasi pada jaringan. Transmisi data yang harusnya kontinu diubah ke transmisi buffer berdasarkan kemampuan pemrosesan data lokal. Dengan menggabungkan agregasi data dengan komunikasi buffer maka ukuran paket data akan meningkat. Dengan demikian, slot waktu yang dibutuhkan untuk mentransmisikan paket data juga ikut meningkat. Jika terjadi suatu kondisi dimana beberapa node ingin mentransmisikan datanya secara bersamaan, maka node forwarding masing-masing lingkungan multihop akan menjadi hambatan, terutama pada jaringan yang berdaya rendah dan bandwidth yang terbatas. Maka buffer overflows dan rasio paket loss yang meningkat akan menjadi masalah besar [6].

Pada kondisi nyata, jaringan komunikasi pada wireless sensor network memiliki buffer yang sangat kecil. Di nrf24l01+ payload data yang disediakan hanya 32 byte dan di IEEE 802.15.4 payload data sebesar 133 bytes. Belum lagi jika kita menggunakan modul tambahan. Pada Xbee beban buffer yang dialokasikan untuk 802.15.4 dan ZigBee masing-masing adalah 100 dan 72 byte [7].

Oleh karena itu, dibutuhkan suatu manajemen penggunaan paket yang baik untuk menghindari terjadinya buffer overflows. Dalam Tugas Akhir ini metode yang

diusulkan adalah implementasi kompresi adaptive menggunakan metode heatshrink untuk pengiriman data pada wireless sensor network berbasis Zigbee. Algoritma kompresi data yang digunakan adalah heatshrink. Algoritma ini berbasis pada algoritma Lempel-Ziv-Storer-Szymanski (LZSS) yang merupakan lossless kompresi data yang cocok untuk kompresi data pada embedded system.

4. RUMUSAN MASALAH

Rumusan masalah yang terdapat dalam tugas akhir ini adalah sebagai berikut :

1. Bagaimana cara menentukan level kompresi data ?
2. Bagaimana metode pengiriman kompresi data pada protocol Zigbee ?
3. Bagaimana tingkat efisiensi dari sistem yang dibangun dapat diukur ?

5. BATASAN MASALAH

Permasalahan yang akan diselesaikan dalam tugas akhir ini memiliki beberapa batasan sebagai berikut :

1. Menggunakan mikrokontroler Arduino
2. Komunikasi nirkabel menggunakan protocol Zigbee
3. Menggunakan algoritma heatshrink untuk kompresi dan dekompresi data

6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan dari pembuatan Tugas Akhir ini adalah untuk mengurangi konsumsi energi pada saat komunikasi dan mengemat memori penyimpanan pada wireless sensor network menggunakan metode kompresi adaptive dengan algoritma heatshrink.

7. MANFAAT TUGAS AKHIR

Dengan dibuatnya Tugas Akhir ini diharapkan akan memberikan manfaat dalam penggunaan kompresi adaptive pada platform wireless sensor network untuk mengemat konsumsi energi dan memori.

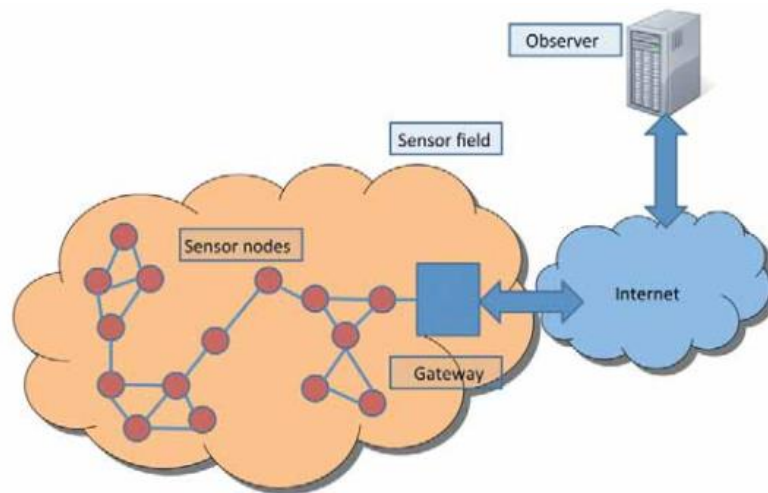
8. TINJAUAN PUSTAKA

8.1 Wireless Network Sensor

Wireless Network Sensor (WSN) adalah jaringan yang dibentuk oleh sekumpulan banyak node sensor dimana masing-masing node dilengkapi dengan sensor untuk mendeteksi fenomena fisik seperti cahaya, panas, tekanan, getaran dan lain-lain. WSN dianggap sebagai metode pengumpulan informasi revolusioner untuk membangun sistem informasi dan komunikasi dalam meningkatkan

kehandalan dan efisiensi sistem infrastruktur. Jika dibandingkan dengan solusi kabel, WSN lebih mudah dipasang dan memiliki fleksibilitas perangkat yang lebih baik [8].

WSN pada umumnya dapat digambarkan sebagai jaringan simpul yang secara kooperatif merasakan dan mengendalikan lingkungan, memungkinkan interaksi antar orang atau computer dan lingkungan sekitar [9]. WSN saat ini biasanya mencakup node sensor, node akuator, gateway dan klien. Sejumlah besar node sensor diletakkan secara acak didalam atau di dekat area pemantauan (sensor field), kemudian membentuk jaringan melalui self-organization. Node sensor memonitor data yang terkumpul untuk dikirim bersama ke node sensor lainnya dengan melompat. Selama proses transmisi, data yang dipantau dapat ditangani oleh beberapa node untuk sampai ke node gateway setelah multihop routing dan akhirnya mencapai node management melalui internet atau satelit. Gambar 1 merupakan contoh ilustrasi atau gambaran mengenai WSN.



Gambar 1. Ilustrasi Wireless Sensor Network [8]

8.2 Arduino

Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwrenya memiliki bahasa pemrograman sendiri. Arduino juga merupakan platform hardware terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan hardware dan software yang fleksibel dan mudah digunakan. Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan syntax dengan bahasa pemrograman C. Karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema hardware arduino dan membangunnya. [10]



Gambar 2. Arduino UNO

Terdapat tiga jenis memori di mikrokontroler yang digunakan pada papan Arduino berbasis avr [19], yaitu :

1. Flash memory, berfungsi sebagai tempat untuk menyimpan sketsa Arduino
2. SRAM adalah tempat sketsa menciptakan dan memanipulasi variable yang sedang dijalankan atau digunakan.
3. EEPROM adalah memori yang dapat digunakan untuk programmer menyimpan informasi jangka Panjang

Pada Flash memory dan EEPROM informasi yang disimpan tetap ada walaupun power telah dimatikan. Sedangkan untuk SRAM, informasi yang disimpan akan hilang ketika power dimatikan. Berikut ini adalah kapasitas memori yang ada pada Arduino UNO Chip ATmega328 (ditunjukkan pada table 1) dan jumlah memori pada Arduino Mega2560 Chip ATmega2560 (ditunjukkan pada table 2).

Jenis Memori	Kapasitas
Flash memory	32k bytes (5k digunakan untuk bootloader)
SRAM	2k bytes
EPPROM	1k byte

Tabel 1. Kapasitas memori pada Arduino UNO

Jenis Memori	Kapasitas
Flash memory	256k bytes (5k digunakan untuk bootloader)
SRAM	8k bytes
EPPROM	4k byte

Tabel 2. Kapasitas memori pada Arduino Mega2560

8.3 Arduino Integrated Development Environment

Arduino Integrated Development Environment (IDE) merupakan editor teks untuk menulis kode, area pesan, konsol teks dan memiliki toolbar dengan tombol untuk fungsi umum dan serangkaian menu. Arduino IDE terhubung ke perangkat keras Arduino dan Genuino untuk mengunggah program dan dapat berkomunikasi dengan mereka. Program yang ditulis menggunakan Arduino Software (IDE) disebut sketsa. Sketsa ini ditulis dalam editor teks dan disimpan dengan ekstensi file .ino. Editor memiliki fitur untuk memotong / menempel dan mencari / mengganti teks. Area pesan memberi umpan balik saat menyimpan dan mengeksport dan menampilkan kesalahan. Konsol menampilkan output teks oleh Arduino Software (IDE), termasuk pesan kesalahan dan informasi lainnya yang lengkap [11].

8.4 Xbee Shield

Shield adalah papan PCB atau lebih dikenal board yang dapat dihubungkan dengan papan Arduino untuk menambah fungsi dari arduino. Xbee Shield ini dirancang dengan agar arduino dapat berkomunikasi secara nirkabel dengan modul Xbee dari Maxstream. Dengan adanya modul ini, sebuah arduino akan mampu berkomunikasi secara nirkabel melebihi 30 meter di dalam ruangan dan 90 meter jika di luar ruangan. Dapat digunakan sebagai serial ataupun USB [12].



Gambar 3. Xbee Shield

8.5 Xbee Modul S2

Xbee modul seri 2 merupakan modul yang berfungsi untuk komunikasi antar jaringan nirkabel. Xbee modul seri 2 menggunakan protocol Zigbee untuk saling berkomunikasi. Modul ini menyediakan transfer data yang sangat handal dengan kecepatan transfer mencapai 250kbps. Setiap perangkat output serial dapat menggunakan modul untuk transfer data, transmisi point to point dan transmisi

jaringan multi-point. Modul ini dirancang untuk aplikasi high-throughput (35kbps) yang membutuhkan latency rendah dan waktu komunikasi yang dapat diprediksi.



Gambar 4. Xbee Modul S2 Pro

8.6 Protokol ZigBee

ZigBee merupakan standar komunikasi untuk perangkat nirkabel jarak pendek berdaya rendah yang berbasis pada standar IEEE 802.15.4 untuk jaringan area pribadi (PAN). Perangkat Zigbee mampu berkomunikasi peer-to-peer, point-to-multipoint dan mesh. Teknologi ini cocok untuk transfer data rate yang rendah, konsumsi daya yang rendah, biaya rendah, protocol jaringan nirkabel yang ditujukan untuk aplikasi otomasi dan remote control. Perangkat nirkabel yang sesuai dengan ZigBee diperkirakan dapat melakukan transmisi 10 sampai 75 meter tergantung pada lingkungan RF dan konsumsi daya yang dikeluarkan untuk aplikasi tertentu. ZigBee memiliki tiga jenis tipe, yaitu 2.4GHz global (data rate 250kbps), 915MHz Americas (data rate 40kbps) dan 868 MHz Europe (data rate 20kbps). Jaringan ZigBee terdiri dari tiga jenis perangkat, yaitu: koordinator, router, dan end devices. Setiap jaringan memiliki ID PAN 16bit. Semua perangkat dalam jaringan ZigBee diberi satu ID PAN.

Berikut ini adalah penjelasan lebih lanjut mengenai ketiga jenis perangkat Zigbee [13] :

- a. Zigbee Coordinator (ZC) : ZC bertindak sebagai coordinator yang mengatur lalu lintas jaringan komunikasi. Harus ada satu ZC dalam setiap jaringan karena perangkat ini memulai jaringan dari awal. Koordinator memulai Personal Area Network (PAN) dengan memilih saluran RF dan PAN ID. ZC memungkinkan router dan end-devices untuk bergabung dengan PAN. Selain itu ZC, mampu menyimpan informasi tentang jaringan, termasuk bertindak sebagai Trust Center dan repository untuk kunci keamanan.
- b. Zigbee Router (ZR) : ZR menjalankan fungsi aplikasi, selain itu router bertindak sebagai perantara, meneruskan data dari satu perangkat ke perangkat lain.

- c. Zigbee End Device (ZED) : ZED dapat melakukan komunikasi dengan koordinator dan router, akan tetapi tidak dapat menyampaikan data dari perangkat lain. Hubungan ini memungkinkan simpul untuk tidur dalam waktu yang cukup lama, sehingga dapat menghemat penggunaan baterai. ZED harus bergabung dengan PAN seperti router sebelum mengirimkan data sensor.

Kelebihan menggunakan ZigBee terutama terletak pada mode AT defaultnya, dimana lapisan PHY dan MAC frame transparan bagi pengguna. Artinya, pengguna biasa tidak akan melihat frame acknowledgment (ACK) atau transmisi ulang modul frekuensi radio (RF) termasuk semua byte aktual yang dikirim. Pengguna hanya akan menyaksikan apakah data berhasil dikirim atau tidak, dengan semua teknis seperti Carrier Sense Multiple Access - Collision Avoidance (CSMA-CA) tersembunyi dari pandangan biasa dan karenanya menawarkan antarmuka yang lebih sederhana [7].

Beban maksimum yang dapat dialokasikan buffer Xbee untuk 802.15.4 dan ZigBee masing-masing adalah 100 dan 72 byte. Manfaat buffer adalah pembacaan sensor ganda dapat dimasukkan ke dalam frame yang sama untuk satu sesi transmisi selama buffer overflow dihindari. [14] menunjukkan bahwa lonjakan arus untuk daya ZigBee menghabiskan 5 sampai 10 kali lebih besar daripada pada operasi normal. Oleh karena itu, buffer harus digunakan untuk mentransmisikan data sebanyak mungkin dalam interval yang dapat ditoleransi dengan aktivasi modul RF minimum.

8.7 Heatshrink

Heatshrink merupakan algoritma kompresi lossless yang berbasis pada Lempel-Ziv-Storer-Szymanski (LZSS). Algoritma kompresi lossless memungkinkan untuk membentuk data asli yang tepat sama dari data yang sudah dikompresi. Algoritma ini cocok digunakan pada system embedded karena dapat berjalan dalam jumlah memori yang sangat kecil (dibawah 50 byte untuk dekompresi praktis). Selain itu, heatshrink dapat bekerja sedikit demi sedikit sambil menangani kebutuhan lain dari system yang berjalan secara real time [15].

Heatshrink menggunakan algoritma Lempel-Ziv-Storer-Szymanski untuk melakukan kompresi dengan beberapa detail implementasi penting yang harus diperhatikan, yaitu [15] :

1. Kompresi dan dekompresi telah dirancang untuk berjalan secara bertahap, pemrosesan dapat bekerja beberapa byte setiap saat. Selain itu dapat juga menangguhkan dan melanjutkan proses sebagai data tambahan atau pada buffer yang tersedia
2. Teknik optional indexing yang digunakan dapat mempercepat proses kompresi
3. Secara umum trade-off implementasi banyak disukai pada penggunaan memori yang rendah.

Lempel-Ziv-Storer-Szymanski (LZSS) adalah salah satu jenis algoritma kompresi yang berbasis dictionary yang bersifat lossless (data dapat di rekonstruksi ulang menjadi data asli). LZSS merupakan salah satu varian dari LZ77 (Lempel Ziv 1977) yang dikembangkan oleh Storer dan Szymansky pada tahun 1982. Perbedaan yang mendasar antara kedua algoritma ini adalah jumlah token (tanda) yang terbentuk yakni dua token pada LZSS dan tiga token pada LZ77. Dua token yang dihasilkan oleh LZSS menunjukkan indeks dan panjang karakter yang sama pada dictionary. Sedangkan pada algoritma LZ77, dua token awal mempunyai fungsi sama dengan LZSS namun ada tambahan satu token yang berisi satu karakter yang mengikuti frasa yang sama tersebut [16].

Untuk proses kompresi dan dekompresi akan dijelaskan secara detail sebagai berikut [17].

1. Kompresi String

Buffer dibagi menjadi dua, yakni buffer untuk pencarian dan buffer look-ahead. Setelah menginisialisasi buffer, karakter dibaca dari input data ke buffer data yang belum di kodekan. Untuk setiap karakter pada buffer yg belum di kodekan, dilakukan proses pencarian substring yang terpanjang di buffer pencarian sesuai dengan buffer look-ahead dimulai dengan karakter inputan pertama. Jika kecocokan substring sudah cukup, maka program akan mengkodekan indeks dan panjang substring ke dalam output. Jika tidak ada substring yang cocok dimulai dengan input pertama karakter masukan yang diberikan, maka karakter tersebut akan langsung ditulis ke output dengan flag yang menandakan tidak ada pengkodean yang dilakukan. Algoritma ini melakukan langkah-langkah ini sampai tidak ada karakter yang tertinggal. Pengkodean dua karakter yang sesuai membutuhkan jumlah byte yang sama jika kita langsung menampilkan dua karakter.

Berikut ini adalah contoh ilustrasi dari proses kompresi data [18]. Pada Gambar 5 menampilkan text asli yang masih belum di kompresi. Pada Gambar 6 akan menampilkan hasil kompresi yang telah dilakukan oleh algoritma LZSS.

```

0: I am Sam
9:
10: Sam I am
19:
20: That Sam-I-am!
35: That Sam-I-am!
50: I do not like
64: that Sam-I-am!
79:
80: Do you like green eggs and ham?
112:
113: I do not like them, Sam-I-am.
143: I do not like green eggs and ham.

```

Gambar 5. Contoh text asli [18]

```

0: I am Sam
9:
10: (5,3) (0,4)
16:
17: That(4,4)-I-am!(19,16)I do not like
45: t(21,14)
49: Do you(58,5) green eggs and ham?
78: (49,14) them,(24,9).(112,15)(93,18).

```

Gambar 6. Hasil text yang telah di kompresi [18]

Pada text asli yang belum mengalami pengompresan, jumlah byte yang dihasilkan adalah 177 byte dari 177 karakter (termasuk spasi dan enter). Setelah dilakukan kompresi, jumlah byte berkurang menjadi 94 byte. Ini tidak termasuk 12 byte pada flag yang menunjukkan apakah potongan teks berikutnya adalah pointer atau literal. Jika ditambahkan dengan jumlah flag maka total ukurannya menjadi 106 byte, tentunya ini masih lebih pendek jika dibandingkan dengan ukuran aslinya 177 byte.

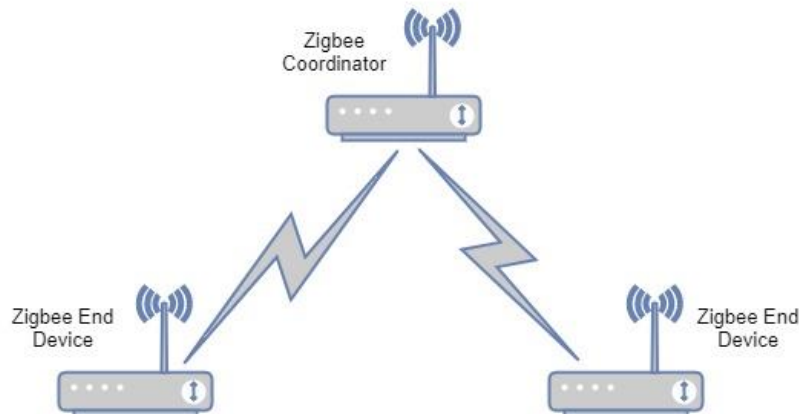
2. Dekompresi String

Proses dekompresi dilakukan dengan menguraikan kode secara langsung dengan melibatkan membaca dan menulis ulang hasil tanpa melakukan pencarian apapun. Flag pengkodean dibaca untuk mengetahui karakter mana yang dikodekan. Jika flag menunjukkan bahwa karakter tersebut dikodekan, jumlah karakter dan posisi awal dikumpulkan dari bagian yang dikodekan. Kemudian jumlah karakter dengan indeks yang diberikan ditulis dari jendela geser ke file output atau memori. Jika tidak dikodekan, karakternya adalah

output secara langsung. Dekompresi mengkonsumsi lebih sedikit sumber daya memori dan waktu komputasi jika dibandingkan dengan proses kompresi.

9. RINGKASAN ISI TUGAS AKHIR

Dalam Tugas Akhir ini akan dibuat sebuah implementasi kompresi adaptive menggunakan metode heatshrink pada wireless sensor network berbasis Zigbee. Gambaran dari arsitektur jaringan aplikasi ini dapat dilihat pada Gambar 5.

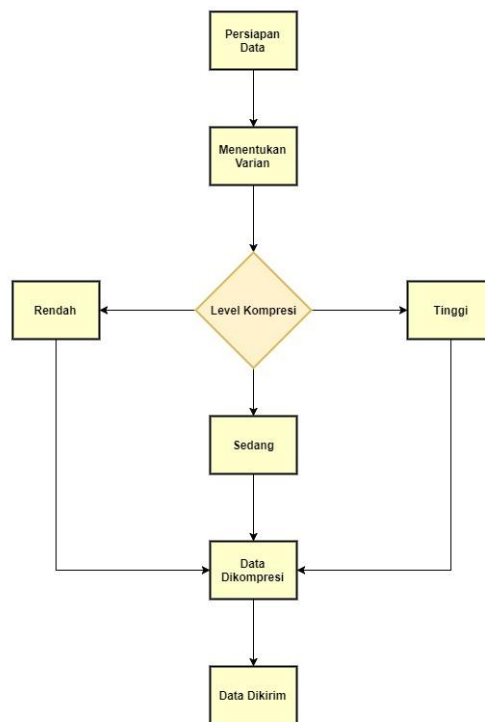


Gambar 7. Arsitektur jaringan yang akan dibangun

Untuk level kompresi data akan ditentukan oleh seberapa besar keberagaman data yang ada menggunakan nilai varian. Untuk menghitung nilai varian kita dapat menggunakan formula :

$$s^2 = \frac{n \sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Semakin besar nilai varian maka data mengalami banyak variasi atau perubahan. Jika nilai varian besar maka level kompresi yang akan diterapkan adalah kompresi dengan level rendah, hal tersebut bertujuan untuk menjaga kedetailan data. Begitu pula sebaliknya, jika nilai varian kecil maka data sedikit mengalami perubahan. Sehingga kompresi dengan level tinggi cocok diterapkan pada kondisi tersebut. Gambaran dari alur kompresi ini dapat dilihat pada Gambar 6.



Gambar 8. Diagram alur kompresi data

Untuk menentukan seberapa banyak data yang akan dikompresi, kita harus mengetahui seberapa besar ukuran data tersebut karena memori yang ada pada wireless sensor network sangat terbatas. Berikut ini adalah Table 3 yang menunjukkan ukuran data berdasarkan tipe datanya.

Nama Tipe	Ukuran Dalam 32-bit	Ukuran Dalam 64-bit
Char	1 byte	1 byte
Short	2 byte	2 byte
Int	4 byte	4 byte
Long	4 byte	8 byte
Float	4 byte	4 byte
Double	8 byte	8 byte
Long long	8 byte	8 byte

Tabel 3. Ukuran tipe data [20]

Pada algoritma heatshrink ukuran buffer telah ditetapkan untuk memungkinkan terjadinya trade-off antara efektivitas kompresi dengan memori kerja. Persyaratan yang dibutuhkan untuk penggunaan buffer IO adalah sebagai berikut [15] :

a. Encoding

$16 + 2 * 2^N$ byte untuk encoding, ditambah lagi untuk indeks pencarian optional yakni $2 * 2^N$ byte untuk mempercepat pengkodean

b. Decoding

$2 * 2^N$ byte untuk decoding, dimana N dapat diatur pada saat pengkodean (encoding)

Besarnya penggunaan ukuran memori pada kompresi ditentukan oleh seberapa besar ukuran window pencarian dan window look-ahead. Berikut ini adalah hasil pengukuran window yang digunakan dengan jumlah memori yang diperlukan, dapat dilihat di Table 4 [15].

	HS (13,4)	HS (11,4)	HS (8,4)	HS (6,3)	HS (5,3)
Memori yang digunakan	8.208 B	2.064 B	272 B	80 B	48 B

Tabel 4. Penggunaan jumlah memori berdasarkan ukuran window [15]

Pada Table 4 menunjukkan semakin besar ukuran window maka semakin besar pula memori yang dibutuhkan untuk melakukan proses kompresi. “HS (11,4)” artinya heatshrink berjalan pada window dengan ukuran 2^{11} byte dan window look-ahead 2^4 byte. Sebuah window hanya dapat menampung 1 byte data. Dikarenakan keterbatasan jaringan buffer Xbee untuk mengirimkan data, yaitu sebesar 72 byte, maka data hasil dari kompresi akan ditampung terlebih dahulu kedalam buffer sepanjang 65 byte. Setelah buffer penuh, maka akan dikirimkan melalui jaringan Zigbee dengan asumsi besar window yang digunakan untuk kompresi adalah HS (8,4).

10.METODOLOGI

a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dikerjakan. Secara detail, proposal tugas akhir ini berisi tentang beberapa bagian yaitu latar belakang diajukannya tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu, dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir dan ringkasan isi yang membahas metode yang akan digunakan dalam tugas akhir. Sub bab metodologi merupakan penjelasan mengenai tahapan penyusunan tugas akhir. Terdapat pula sub bab jadwal pengerjaan yang menjelaskan jadwal pengerjaan tugas akhir dan di akhir bagian terdapat daftar pustaka untuk mencantumkan referensi yang digunakan dalam tugas akhir.

b. Studi literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur sejumlah referensi tentang kompresi data pada WSN. Informasi dan studi literatur tersebut didapat dari buku, *internet*, dan materi-materi kuliah yang berhubungan dengan metode yang digunakan.

c. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dilakukan desain sistem dan desain proses-proses yang ada. Urutan dari analisis dan desain perangkat lunak yang akan dilakukan dengan urutan sebagai berikut.

1. Pemilihan sensor yang akan digunakan
2. Uji coba rangkaian setiap node yang dibuat
3. Uji coba komunikasi protokol Zigbee pada setiap node
4. Implementasi kompresi adaptive pada platform wireless sensor network

d. Implementasi perangkat lunak

Pembangunan aplikasi akan dilakukan menggunakan Bahasa pemrograman C/C++ pada Arduino IDE.

e. Pengujian dan evaluasi

Pengujian akan dilakukan dengan menguji coba fungsionalitas dan performa kompresi data pada sistem WSN sesuai skenario yang telah ditentukan. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian yang telah dilakukan. Pengujian fungsionalitas meliputi uji coba setiap bagian perangkat keras yang dirangkai pada Arduino dan juga uji coba keseluruhan sistem. Pengujian performa meliputi tingkat akurasi hasil kompresi data dan efisiensi energi yang dapat di hemat.

f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi

5. Kesimpulan dan Saran
6. Daftar Pustaka

11. JADWAL KEGIATAN

Tahapan	2017							2018																				
	Desember			Januari				Februari				Maret					April				Mei							
Penyusunan Proposal																												
Studi Literatur																												
Perancangan Sistem																												
Implementasi																												
Pengujian dan Evaluasi																												
Penyusunan Buku																												

12. DAFTAR PUSTAKA

- [1] Peter Harrop dan Raghu Das , “ Wireless Sensor Networks (WSN) 2014-2024: Forecasts, Technologies, Players” IDTechEx, 2014. [Daring]. Tersedia: <https://www.idtechex.com/research/reports/wireless-sensor-networks-wsn-2014-2024-forecasts-technologies-players-000382.asp> [Diakses 25-Des-2017].
- [2] S. Rhee, D. Seetharam dan S. Liu, “Techniques for Minimizing Power Consumption in Low Data-Rate Wireless Sensor Networks”, IEEE Wireless Communications and Networking Conference, vol. 3, pp. 1727-1731, 2004.
- [3] R. Rajagopalan dan P. K. Varshney, “Data Aggregation Techniques in Sensor Networks: A Survey” in , IEEE Communication Surveys & Tutorials, vol. 8, no. 4, pp. 48-63, 2006.
- [4] C. Wang, H. Ma, Y. He dan S. Xiong, "Adaptive Approximate Data Collection for Wireless Sensor Network", IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 6, pp. 1004-1016, 2012.
- [5] P. Ghaffariyan, “An Effective Data Aggregation Mechanism for Wireless Sensor Networks”, 6th International Conference on Wireless Communications Networking and Mobile Computing, 2010.
- [6] M. Vodel dan W. Hardt, “Data Aggregation and Data Fusion Techniques In WSN/SANET Topologies - A Critical Discussion,” TENCON 2012 - 2012 IEEE Region 10 Conference, pp. 1-6, 2012
- [7] Solahuddin Y. F dan W. Ismail, “Data fusion for reducing power consumption in Arduino-Xbee wireless sensor network platform”, Computer and Information Sciences (ICCOINS), 2014.
- [8] Shu Yinbiao dan Kang Lee, “Internet of Things: Wireless Sensor Networks”, International Electrotechnical Commission, White Paper, 2014.
- [9] Bröring A, “New Generation Sensor Web Enablement”, PMC, 2011. [Daring]. Tersedia: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231615/> [Diakses 2-Jan-2018]
- [10] “Arduino.” Wikipedia bahasa Indonesia, ensiklopedia bebas. [Daring]. Tersedia: <https://id.wikipedia.org/wiki/Arduino> [Diakses 2-Jan-2018]
- [11] Arduino, “Arduino Software (IDE).” [Daring]. Tersedia: <https://www.arduino.cc/en/Guide/Environment> [Diakses 3-Jan-2018]
- [12] “Shield Arduino.” [Daring]. Tersedia: <http://www.belajararduino.net/shield-arduino/> [Diakses 3-Jan-2018]
- [13] “Zigbee Networking with XBee Series 2 and Seeed's Products.” [Daring]. Tersedia: http://wiki.seeed.cc/Zigbee_Networking_with_XBee_Series_2_and_Seed's_Products/ [Diakses 3-Jan-2018]
- [14] J. Song dan Y. K. Tan, “Energy Consumption Analysis of ZigBee-based Energy Harvesting Wireless Sensor Network”, IEEE International Conference on Communication Systems (ICCS), 2012
- [15] Scott Vokes, “heatshrink: An Embedded Data Compression Library.” [Daring].

- Tersedia: <https://spin.atomicobject.com/2013/03/14/heatshrink-embedded-data-compression/> [Diakses 3-Jan-2018]
- [16] “Algoritma Kompresi LZSS” [Daring]. Tersedia: <https://mahfudharun.wordpress.com/2012/06/14/algoritma-kompresi-lzss/> [Diakses 6-Jan-2018]
- [17] Adnan Ozsoy dan Martin Swany, “CULZSS: LZSS Lossless Data Compression on CUDA”, IEEE International Conference on Cluster Computing (CLUSTER), 2011
- [18] “Lempel–Ziv–Storer–Szymanski.” Wikipedia, the free encyclopedia . [Daring]. Tersedia: <https://en.wikipedia.org/wiki/Lempel–Ziv–Storer–Szymanski> [Diakses 6-Jan-2018]
- [19] Arduino, “Memory.” [Daring]. Tersedia: <https://www.arduino.cc/en/Tutorial/Memory> [Diakses 6-Jan-2018]
- [20] Oracle, “Data Types and Sizes.” [Daring]. Tersedia: <https://docs.oracle.com/cd/E19253-01/817-6223/chp-typeopexpr-2/index.html> [Diakses 6-Jan-2018]