**Ado.Ok**

**TUGAS AKHIR – KI141502**

**IMPLEMENTASI KOMPRESI ADAPTIVE MENGGUNAKAN METODE HEATSHRINK UNTUK PENGIRIMAN DATA PADA WIRELESS SENSOR NETWORK BERBASIS ZIGBEE**

MUHAMAD HENDRI FEBRIANSYAH

NRP 051114410000036

Dosen Pembimbing

Waskito Wibisono, S.Kom., M.Eng., Ph.D.

Ir. Muchammad Husni, M.Kom.

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018



**TUGAS AKHIR – KI141502**

**IMPLEMENTASI KOMPRESI ADAPTIVE MENGGUNAKAN METODE HEATSHRINK UNTUK PENGIRIMAN DATA PADA WIRELESS SENSOR NETWORK BERBASIS ZIGBEE**

MUHAMAD HENDRI FEBRIANSYAH

NRP 051114410000036

Dosen Pembimbing

Waskito Wibisono, S.Kom., M.Eng., Ph.D.

Ir. Muchammad Husni, M.Kom.

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

[Halaman ini sengaja dikosongkan]



**FINAL PROJECT – KI141502**

**IMPLEMENTATION OF HUNT-AND-KILL ALGORITHM FOR PUZZLE ARRANGEMENTS ON ‘PLANT THE FUTURE’ GAME**

MUHAMAD HENDRI FEBRIANSYAH

NRP 051114410000036

Advisor

Waskito Wibisono, S.Kom., M.Eng., Ph.D.

Ir. Muchammad Husni, M.Kom.

INFORMATICS DEPARTEMENT

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2018

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN**

**IMPLEMENTASI KOMPRESI ADAPTIVE MENGGUNAKAN METODE HEATSHRINK UNTUK PENGIRIMAN DATA PADA WIRELESS SENSOR NETWORK BERBASIS ZIGBEE**

**TUGAS AKHIR**Diajukan Guna Memenuhi Salah Satu Syarat   
Memperoleh Gelar Sarjana Komputer  
pada   
Bidang Studi Interaksi Grafika dan Seni  
Program Studi S-1 Departemen Informatika  
Fakultas Teknologi Informasi dan Komunikasi  
Institut Teknologi Sepuluh Nopember

Oleh :  
**Muhamad Hendri Febriansyah**NRP : 05111440000036

Disetujui oleh Dosen Pembimbing Tugas Akhir :

|  |  |
| --- | --- |
| Waskito Wibisono, S.Kom., M.Eng., Ph.D.  NIP: 197612152003121001 | ........................... (pembimbing 1) |
| Ir. Muchammad Husni, M.Kom. NIP: 198106222005012002 | ........................... (pembimbing 2) |

**Surabaya  
MEI 2018**

[Halaman ini sengaja dikosongkan]

**IMPLEMENTASI KOMPRESI ADAPTIVE MENGGUNAKAN METODE HEATSHRINK UNTUK PENGIRIMAN DATA PADA WIRELESS SENSOR NETWORK BERBASIS ZIGBEE**

Nama Mahasiswa : MUHAMAD HENDRI FEBRIANSYAH

NRP : 05111440000036

Departemen : Informatika FTIK-ITS

Dosen Pembimbing I : Waskito Wibisono, S.Kom., M.Eng., Ph.D.

Dosen Pembimbing II : Ir. Muchammad Husni, M.Kom.

**Abstrak**

Kemajuan teknologi membuat banyak perubahan terutama dalam perkembangan game atau aplikasi permainan. Pengalaman bermain mulai berubah dari sesuatu yang statis menjadi dinamis, contohnya dalam penyusunan *maze* atau *puzzle* dalam aplikasi permainan. Cukup banyak algoritma yang dapat digunakan untuk menyusun *puzzle* dan membuatnya dinamis*,* namun *puzzle* sendiri memiliki aturan yang unik untuk masing-masingnya. Sehingga penerapan penyusunan *puzzle*  perlu disesuaikan dengan *gameplay* atau aturan permainan yang ada pada game.

Pada penelitian ini penulis menawarkan sebuah cara penggunaan algoritma penyusunan *maze* yang dapat digunakan dan disesuaikan agar dapat menghasilkan *puzzle* yang berubah-ubah dan dapat diselesaikan. ‘Plant the Future’ merupakan game *puzzle* yang dirancang untuk *smartphone* berbasis Android. Game ini memiliki stage dengan susunan *puzzle* yang mirip dengan *maze* sederhana dengan aturan permainan yang cukup unik, yaitu dengan adanya dua *puzzle* pada setiap stage. *Puzzle* yang ada bervariasi dimensinya mulai dari 4x4 hingga 11x11. Dalam permainan ini pergerakan pemain sangat bergantung pada susunan *puzzle* untuk mendapatkan poin agar dapat memenangkan permainan. Oleh karena itu diperlukan sebuah cara agar susunan *puzzle* dapat dimenangkan. Untuk penyusunan *puzzle* maka digunakan algoritma Hunt-and-Kill agar dapat menghasilkan susunan *puzzle* yang bukan hanya dinamis, namun juga dapat diselesaikan.

Uji coba dilakukan dengan memainkan 30 stage dengan susunan puzzle yang dirancang menggunakan algoritma Hunt-and-Kill. Setiap stage diuji coba masih-masing sebanyak lima kali dimainkan. Setelah diuji coba maka diamati apakah susunan puzzle stage yang dihasilkan oleh algoritma Hunt-and-Kill dapat diselesaikan atau tidak. Pada hasilnya, algoritma Hunt-and-Kill dapat menyusun puzzle pada 30 stage game ‘Plant the Future’ dengan baik. Puzzle yang dirancang dengan dimensi bervariasi dari 4x4 sampai 11x11 dapat diselesaikan dan dimenangkan, sehingga algoritma ini dapat dikatakan cocok dengan aturan permainan game ‘Plant the Future’.

**Kata kunci: Hunt-and-Kill, maze, puzzle**

**IMPLEMENTATION OF HUNT-AND-KILL ALGORITHM FOR PUZZLE ARRANGEMENTS ON PLANT THE FUTURE GAME**

Name : MUHAMAD HENDRI FEBRIANSYAH

NRP : 05111440000036

Major : Informatics – FTIK ITS

Supervisor I : Waskito Wibisono, S.Kom., M.Eng., Ph.D.

Supervisor II : Ir. Muchammad Husni, M.Kom.

**Abstract**

Technological advances make many changes especially in the development of games or game applications. Play experience starts to change from something static to dynamic, for example in the preparation of mazes or puzzles in game applications. Quite a lot of algorithms can be used to construct puzzles and make them dynamic, but the puzzles themselves have unique rules for each. So the application of puzzle preparation needs to be adjusted with the gameplay or game rules that exist in the game.

In this study the authors offer a way of using maze compilation algorithms that can be used and customized in order to produce puzzles that change and can be solved. 'Plant the Future' is a puzzle game designed for Android-based smartphones. This game has a stage with a puzzle arrangement similar to a simple maze with a game rules that are quite unique. Puzzles vary in dimensions ranging from 4x4 to 11x11. In this game the movement of players is very dependent on the order of the puzzle to get points. Therefore we need a way to make the puzzle arrangement can be won. To arrange a puzzle, Hunt-and-Kill algorithm is used in order to generate an array of puzzles that are not only dynamic but also resolvable.

Trials are performed by playing 30 stages with a puzzle arrangement designed using the Hunt-and-Kill algorithm. Each stage is tested each stage five times played. Once tested, it is observed whether the order of puzzle stages generated by the Hunt-and-Kill algorithm can be solved or not. In the result, the Hunt-and-Kill algorithm can construct the puzzle on 30 stage 'Plant the Future' games well. Puzzles designed with dimensions varying from 4x4 to 11x11 can be completed and won, so the algorithm can be said to match the game rules of 'Plant the Future' game.

**Keywords: Hunt-and-Kill, maze, puzzle**

**KATA PENGANTAR**

****

Alhamdulillahirabbil’alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “IMPLEMENTASI ALGORITMA HUNT-AND-KILL UNTUK PERANCANGAN *PUZZLE* PADA GAME 'PLANT THE FUTURE'”.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan tugas akhir ini penulis bisa mendapatkan ilmu lebih serta memanfaatkan semua ilmu yang telah didapatkan pada saat berkuliah di Departemen Informatika FTIK ITS.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Orang tua, Kakak, Adik serta Saudara-saudara yang selalu mendoakan dan mendukung penulis.
3. Pak Imam Kuswardayan, S.Kom, M.T. selaku pembimbing I yang selalu memberikan arahan, motivasi dan bantuan sekaligus bimbingan kepada penulis selama pengerjaan Tugas Akhir.
4. Ibu Anny Yuniarti, S.Kom., M.Comp.Sc. selaku pembimbing II yang juga telah sangat membantu, dan membimbing saat pengerjaan Tugas Akhir ini.
5. Andreas Galang Anugerah dan Ade Nobi Miranto yang tergabung bersama penulis dalam tim fragments, yang telah membantu penulis selama pengerjaan Tugas Akhir.
6. Bapak dan Ibu Dosen Karyawan Teknik Informatika FTIf ITS yang telah memberikan ilmunya.
7. Teman-teman angkatan 2014 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, kakak-kakak angkatan 2013, 2012, serta adik-adik angkatan 2015 dan 2016 yang membuat penulis untuk selalu belajar.
8. Serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Mei 2018

**DAFTAR** ISI

[Abstrak vii](#_Toc515847622)

[Abstract ix](#_Toc515847623)

[KATA PENGANTAR xi](#_Toc515847624)

[DAFTAR ISI xiii](#_Toc515847625)

[DAFTAR GAMBAR xviii](#_Toc515847626)

[DAFTAR TABEL xx](#_Toc515847627)

[DAFTAR KODE SUMBER xxii](#_Toc515847628)

[BAB I PENDAHULUAN 1](#_Toc515847629)

[1.1. Latar Belakang 1](#_Toc515847630)

[1.2. Rumusan Masalah 3](#_Toc515847631)

[1.3. Batasan Masalah 3](#_Toc515847632)

[1.4. Tujuan 4](#_Toc515847633)

[1.5. Manfaat 4](#_Toc515847634)

[1.6. Metodologi 4](#_Toc515847635)

[2.6.1. Penyusunan Proposal Tugas Akhir 4](#_Toc515847643)

[2.6.2. Studi Literatur 5](#_Toc515847644)

[2.6.3. Analisis dan Desain Perangkat Lunak 5](#_Toc515847645)

[2.6.4. Implementasi dan Pembuatan Sistem 5](#_Toc515847646)

[2.6.5. Pengujian dan Evaluasi 6](#_Toc515847647)

[2.6.6. Penyusunan Laporan Tugas Akhir 6](#_Toc515847648)

[2.7. Sistematika Penulisan 6](#_Toc515847649)

[BAB II TINJAUAN PUSTAKA 9](#_Toc515847650)

[2.1. Wireless Sensor Network 9](#_Toc515847653)

[2.2. Arduino 10](#_Toc515847654)

[2.3. Arduino Integrated Development Environment 12](#_Toc515847655)

[2.4. XBee Shield 14](#_Toc515847656)

[2.5. XBee Modul S2 16](#_Toc515847657)

[2.6. Protokol ZigBee 17](#_Toc515847658)

[2.7. Algoritma Heatshrink 18](#_Toc515847659)

[2.8. Bahasa Pemrograman C 22](#_Toc515847660)

[2.9. MicroSD Card Adapter 22](#_Toc515847661)

[2.10. DIGI XCTU 23](#_Toc515847662)

[BAB III PERANCANGAN PERANGKAT LUNAK 25](#_Toc515847663)

[3.1. Deskripsi Umum Sistem 25](#_Toc515847665)

[3.2. Arsitektur Umum Sistem 26](#_Toc515847666)

[3.3. Perancangan Komunikasi Sistem 27](#_Toc515847667)

[3.4. Perancangan Kompresi dan Dekompresi Data 31](#_Toc515847668)

[3.4.1. Cara Kerja Algoritma Heatshrink 32](#_Toc515847669)

[3.4.2. Konfigurasi Algoritma Heatshrink 33](#_Toc515847670)

[3.5. Perancangan Pengiriman Data 35](#_Toc515847671)

[3.6. Perancangan Dekompresi Data 36](#_Toc515847672)

[3.7. Perancangan Perangkat Keras 37](#_Toc515847673)

[3.7.1. Perancangan Perangkat ZigBee Coordinator 38](#_Toc515847674)

[3.7.2. Perancangan Perangkat ZigBee Router 39](#_Toc515847675)

[BAB IV IMPLEMENTASI 41](#_Toc515847676)

[4.1. Lingkungan Implementasi 41](#_Toc515847678)

[4.1.1. Lingkungan Implementasi Perangkat Keras 41](#_Toc515847679)

[4.1.2. Lingkungan Implementasi Perangkat Lunak 43](#_Toc515847680)

[4.2. Implementasi Perangkat Keras 43](#_Toc515847681)

[4.2.1. Perangkat Node ZigBee Coordinator 44](#_Toc515847682)

[4.2.2. Perangkat Node ZigBee Router 45](#_Toc515847683)

[4.3. Implementasi Inisialisasi Data Pada SD Card 45](#_Toc515847684)

[4.4. Implementasi Read Data Dari SD Card 47](#_Toc515847685)

[4.5. Implementasi Setting Konfigurasi Encoder / Decoder 48](#_Toc515847686)

[4.6. Implementasi Kompresi Data 48](#_Toc515847687)

[4.7. Implementasi Pengiriman Data 48](#_Toc515847688)

[4.8. Implementasi Dekompresi Data 49](#_Toc515847689)

[BAB V PENGUJIAN DAN EVALUASI 50](#_Toc515847690)

[5.1. Lingkungan Uji Coba 50](#_Toc515847692)

[5.2. Uji Coba Fungsionalitas 51](#_Toc515847693)

[5.2.1. Skenario Uji Coba Membaca Data dari SD Card 51](#_Toc515847694)

[5.2.2. Skenario Uji Coba Komunikasi Node ZigBee Router dengan Node ZigBee Coordinator 52](#_Toc515847695)

[5.2.3. Skenario Uji Coba Kompresi Data 53](#_Toc515847696)

[5.2.4. Skenario Uji Coba Pengiriman Data 53](#_Toc515847697)

[5.2.5. Skenario Uji Coba Dekompresi Data 54](#_Toc515847698)

[5.2.6. Hasil Pengujian 54](#_Toc515847699)

[5.3. Uji Coba Performa 55](#_Toc515847700)

[5.3.1. Uji Coba Efektifitas Hasil Kompresi Data 55](#_Toc515847701)

[5.3.2. Uji Coba Waktu Untuk Kompresi dan Decompresi 55](#_Toc515847702)

[5.3.3. Uji Coba Konsumsi Energi 55](#_Toc515847703)

[5.3.4. Hasil Pengujian 55](#_Toc515847704)

[5.4. Pengujian Pengguna 57](#_Toc515847705)

[5.4.1. Skenario Pengujian 57](#_Toc515847706)

[5.4.2. Hasil Pengujian 58](#_Toc515847707)

[BAB VI KESIMPULAN DAN SARAN 59](#_Toc515847708)

[6.1. Kesimpulan 59](#_Toc515847710)

[6.2. Saran 59](#_Toc515847711)

[DAFTAR PUSTAKA 62](#_Toc515847712)

[BIODATA PENULIS 64](#_Toc515847713)

[LAMPIRAN A 66](#_Toc515847714)

[LAMPIRAN B 67](#_Toc515847715)

*[Halaman ini sengaja dikosongkan]*

DAFTAR GAMBAR

[Gambar 2.1 Game Cute Munchies 8](#_Toc504112739)

[Gambar 2.2 Algoritma Hunt-and-Kill 10](#_Toc504112740)

[Gambar 3.1 Poin yang tidak dapat dicapai di ‘Masa Depan’ 12](#_Toc504112741)

[Gambar 3.2 Tempat penanaman bibit di 'Masa Lalu' 13](#_Toc504112742)

[Gambar 3.3 Poin bisa didapatkan dengan adanya pohon 13](#_Toc504112743)

[Gambar 3.4 Flow diagram Algoritma Hunt-and-Kill 17](#_Toc504112744)

[Gambar 3.5 Class diagram rancangan Hunt-and-Kill 19](#_Toc504112745)

[Gambar 3.6 Flow diagram tahap ‘Hunt’ 20](file:///F:\Documents\Tugas%20Akhir\PASCA%20SIDANG\Revisi%20Buku\Dokumen%20Tugas%20Akhir%20Syauki%20Aulia.docx#_Toc504112746)

[Gambar 3.7 Flow diagram tahap ‘Kill’ 22](file:///F:\Documents\Tugas%20Akhir\PASCA%20SIDANG\Revisi%20Buku\Dokumen%20Tugas%20Akhir%20Syauki%20Aulia.docx#_Toc504112747)

[Gambar 3.8 Peletakan *obstacle* 24](file:///F:\Documents\Tugas%20Akhir\PASCA%20SIDANG\Revisi%20Buku\Dokumen%20Tugas%20Akhir%20Syauki%20Aulia.docx#_Toc504112748)

[Gambar 3.9 Array hasil(atas), tampilan realisasi'masa depan'(kiri) dan 'masa lalu'(kanan) 26](file:///F:\Documents\Tugas%20Akhir\PASCA%20SIDANG\Revisi%20Buku\Dokumen%20Tugas%20Akhir%20Syauki%20Aulia.docx#_Toc504112749)

[Gambar 4.1 Implementasi tampilan *Main Menu* 33](#_Toc504112750)

[Gambar 4.2 Implementasi Tampilan *Stage Selection* 34](file:///F:\Documents\Tugas%20Akhir\PASCA%20SIDANG\Revisi%20Buku\Dokumen%20Tugas%20Akhir%20Syauki%20Aulia.docx#_Toc504112751)

[Gambar 4.3 Implementasi tampilan *Gameplay* Stage 35](file:///F:\Documents\Tugas%20Akhir\PASCA%20SIDANG\Revisi%20Buku\Dokumen%20Tugas%20Akhir%20Syauki%20Aulia.docx#_Toc504112752)

[Gambar 4.4 Penerapan konfigurasi stage 37](#_Toc504112753)

[Gambar 5.1 Tampilan *Main Menu* 42](#_Toc504112754)

[Gambar 5.2 Tampilan *Stage Selection* 43](#_Toc504112755)

[Gambar 5.3 Tampilan pergerakakan karakter 44](#_Toc504112756)

[Gambar 5.4 Tampilan perpindahan waktu maze ‘masa depan’ (kiri) dan maze ‘masa lalu’(kanan) 45](#_Toc504112757)

[Gambar 5.5 Pemain mendapatkan poin saat bergerak 45](#_Toc504112758)

[Gambar 5.6 Bibit belum ditanam(kiri) dan setelah ditanam(kanan) 46](#_Toc504112759)

[Gambar 5.7 Pohon muncul di tampilan masa depan 47](#_Toc504112760)

[Gambar 5.8 Tampilan saat pemain menang 47](#_Toc504112761)

[Gambar 5.9 Batas waktu habis dan pemain kalah 48](#_Toc504112762)

[Gambar 5.10 Tampilan stage pengujian 49](#_Toc504112763)

[Gambar B.1 Stage 2 77](#_Toc504112764)

[Gambar B.2 Stage 3-1 77](#_Toc504112765)

[Gambar B.3 Stage 3-2 78](#_Toc504112766)

[Gambar B.4 Stage 4-1 78](#_Toc504112767)

[Gambar B.5 Stage 4-2 79](#_Toc504112768)

[Gambar B.6 Stage 4-3 79](#_Toc504112769)

[Gambar B.7 Stage 5-1 80](#_Toc504112770)

[Gambar B.8 Stage 5-2 80](#_Toc504112771)

[Gambar B.9 Stage 5-3 81](#_Toc504112772)

[Gambar B.10 Stage 5-4 81](#_Toc504112773)

[Gambar B.11 Stage 6-1 82](#_Toc504112774)

[Gambar B.12 Stage 6-2 82](#_Toc504112775)

[Gambar B.13 Stage 6-3 83](#_Toc504112776)

[Gambar B.14 Stage 6-4 83](#_Toc504112777)

[Gambar B.15 Stage 6-5 84](#_Toc504112778)

[Gambar B.16 Stage 7-1 84](#_Toc504112779)

[Gambar B.17 Stage 7-2 85](#_Toc504112780)

[Gambar B.18 Stage 7-3 85](#_Toc504112781)

[Gambar B.19 Stage 7-4 86](#_Toc504112782)

[Gambar B.20 Stage 7-5 86](#_Toc504112783)

[Gambar B.21 Stage 8-1 87](#_Toc504112784)

[Gambar B.22 Stage 8-2 87](#_Toc504112785)

[Gambar B.23 Stage 8-3 88](#_Toc504112786)

[Gambar B.24 Stage 8-4 88](#_Toc504112787)

[Gambar B.25 Stage 8-5 89](#_Toc504112788)

[Gambar B.26 Stage 9-1 89](#_Toc504112789)

[Gambar B.27 Stage 9-2 90](#_Toc504112790)

[Gambar B.28 Stage 9-3 90](#_Toc504112791)

[Gambar B.29 Stage 9-4 91](#_Toc504112792)

[Gambar B.30 Stage 9-5 91](#_Toc504112793)

[Gambar C.1 Form pengujian pengguna 1.A 93](#_Toc504112794)

[Gambar C.2 Form pengujian pengguna 1.B 94](#_Toc504112795)

[Gambar C.3 Form pengujian pengguna 2.A 95](#_Toc504112796)

[Gambar C.4 Form pengujian pengguna 2.B 96](#_Toc504112797)

[Gambar C.5 Form pengujian pengguna 3.A 97](#_Toc504112798)

[Gambar C.6 Form pengujian pengguna 3.B 98](#_Toc504112799)

DAFTAR TABEL

[Tabel 3.1 Konfigurasi stage game 15](#_Toc503524631)

[Tabel 4.1 Penjelasan metode MapGenerator 38](#_Toc503524632)

[Tabel 5.1 Lingkungan pengujian 41](#_Toc503524633)

[Tabel 5.2 Pengujian aplikasi permainan 41](#_Toc503524634)

[Tabel 5.3 Hasil pengujian fungsionalitas 48](#_Toc503524635)

[Tabel 5.4 Hasil pengujian 50](#_Toc503524636)

[Tabel 5.5 Hasil Pengujian Pengguna 51](#_Toc503524637)

*[Halaman ini sengaja dikosongkan]*

DAFTAR KODE SUMBER

[Kode Sumber 4.1 Implementasi pergerakan karakter 30](#_Toc503524638)

[Kode Sumber 4.2 Perpindahan waktu 31](#_Toc503524639)

[Kode Sumber 4.3 Mendapatkan poin 32](#_Toc503524640)

[Kode Sumber 4.4 Penanaman bibit 32](#_Toc503524641)

[Kode Sumber 4.5 Pemilihan stage 33](#_Toc503524642)

[Kode Sumber 4.6 StageController.cs 37](#_Toc503524643)

[Kode Sumber 4.7 Implementasi realisasi *puzzle* 40](#_Toc503524644)

[Kode Sumber A.1 Hunt-and-Kill 76](#_Toc503524645)

*[Halaman ini sengaja dikosongkan]*

# BAB I PENDAHULUAN

Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

## 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 kodisi 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 managemen pengunaan 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.

## Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

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

## Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yakni sebagai berikut:

1. Menggunakan mikrokontroller Arduino Mega.
2. Komunikasi nirkabel menggunakan protokol ZigBee.
3. Menggunakan algoritma Heatshrink untuk melakukan kompresi dan dekompresi data
4. Algoritma Heatshrink dibuat dalam bahasa pemrograman C.

## Tujuan

Tujuan pembuatan tugas akhir ini adalah untuk memanfaatkan algoritma heatshrink dalam kompresi data sehingga dapat menghemat penggunaan energi pada saat melakukan pengiriman data pada *platform wireless sensor network*.

## Manfaat

Manfaat dari penelitian tugas akhir ini adalah adanya algoritma heatshrink yang dapat digunakan untuk kompresi adaptive dalam pengiriman data di *platform wireless sensor network* sehingga dapat mengemat penggunaan energi.

## Metodologi

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:



### Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan terdiri atas hal yang menjadi latar belakang diajukannya usulan 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. Sub bab metodologi berisi penjelasan tahapan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal ginggu penyususunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

### Studi Literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Arduino
2. Bahasa pemrograman C
3. Algoritma Heatshrink
4. Protokol Zigbee

### Analisis dan Desain Perangkat Lunak

Pada tahap ini akan dilakukan analisa, perancangan, dan pendefinisian kebutuhan system untuk mengetahui permasalahan yang akan dihadapi pada tahap implementasi. Kemudian akan dijabarkan kebutuhan-kebutuhan tersebut ke dalam perancangan fitur sistem. Berikut langkah yang akan dilakukan perancangan proses perangkat lunak:

1. Perancangan rangkaian node yang akan dibuat
2. Uji coba komunikasi menggunkan protocol ZigBee pada rangkaian node
3. Implementasi kompresi dan dekompresi pada node
4. Implementasi kompresi adaptive pada node

### Implementasi dan Pembuatan Sistem

Aplikasi ini dibangun menggunakan Arduino IDE (Integrated Development Environment), dengan bahasa pemrograman C dan algoritma Heatshrink.

### Pengujian dan Evaluasi

Tahap pengujian dan evaluasi berisi pengujian aplikasi dan evaluasi berdasarkan hasil pengujian. Pada tahap ini dilakukan pengujian dari fungsionalitas dan performa system WSN yang mana nanatinya akan dijalankan scenario yang sudah 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 data yang dapat di hemat.

### Penyusunan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

## Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. Bab I. Pendahuluan

Bab ini menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan dari pembuatan tugas akhir, metodologi yang digunakan selama penyusunan, serta sistematika penulisan dari pembuatan tugas akhir.

1. Bab II. Tinjauan Pustaka

Bab ini memaparkan hasil studi literatur yang digunakan sebagai dasar untuk menyelesaikan tugas akhir ini, terdiri atas deskripsi mengenai wireless sensor network, mikrokontroller, protokol zigbee, bahasa pemrograman C, dan algoritma Heatshrink

1. Bab III. Perancangan Perangkat Lunak

Bab ini menjelaskan perancangan komunikasi pada sistem, perancangan kompresi data, pengiriman data, perancangan perangkat keras dan lunak.

1. Bab IV. Implementasi

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Implementasi yang akan diterapkan berupa pseudocode.

1. Bab V. Pengujian dan Evaluasi

Bab ini menjelaskan pengujian pada sistem yang dibuat. Pengujian akan dilakukan berupa uji coba fungsionalitas dan uji coba performa.

1. Bab VI. Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan yang didapat dari proses pembuatan tugas akhir beserta saran-saran untuk pengembangan selanjutnya.

1. Daftar Pustaka

Bab ini berisi daftar pustaka yang dijadikan literatur dalam pembuatan tugas akhir.

1. Lampiran

Lampiran yang ada berisikan kelengkapan – kelengkapan yang diperlukan dalam menyusun buku tugas akhir.

[Halaman ini sengaja dikosongkan]

# BAB II TINJAUAN PUSTAKA

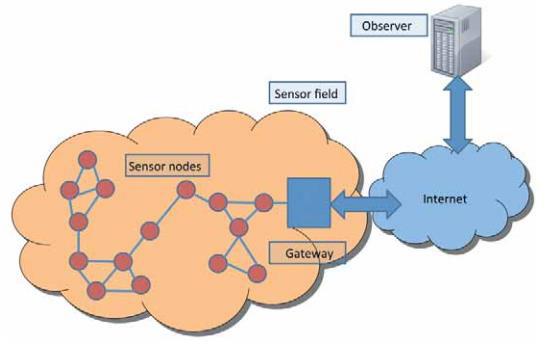
Bab ini membahas teori-teori dasar yang berkaitan dengan pokok bahasan tugas akhir. Bab ini juga menjelaskan modul dan alat yann nantinya akan digunakan pada tahap implementasi program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap alat yang digunakan dan berguna sebagai penunjang dalam pengembangan perangkat lunak.



## Wireless Sensor Network

*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].

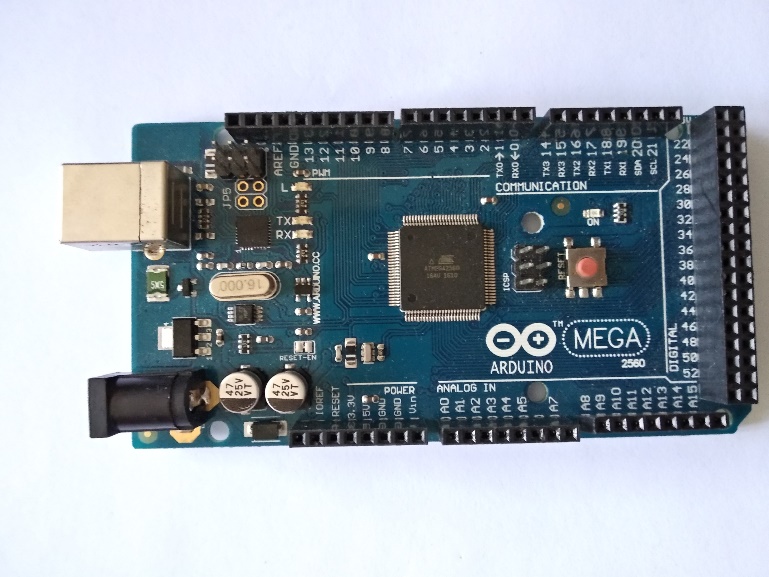
WSN pada umumnya dapat digambarkan sebagai jaringan simpul yang secara kooperatif merasakan dan mengendalikan lingkungan, memungkinkan interaksi antar orang atau computer dan lingkungan sekitar [9]. WSN saat ini biasanya mencakup node sensor, node akuator, gateway dan klien. Sejumlah besar node sensor 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 ditanggani 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 2. Ilustrasi Wireless Sensor Network [8]

## 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 softwarenya 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 Mega

Terdapat tiga jenis memori yang ada pada mikrokontroller papan Arduino berbasis AVR [19], yaitu :

a. Flash memory, berfungsi sebagai tempat untuk menyimpan sketsa Arduino.

b. SRAM adalah tempat sketsa menciptakan dan memanipulasi variable yang sedang dijalankan atau digunakan.

c. 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 data sheet yang ada pada Arduino Mega2560 Chip ATmega2560 (ditunjukkan pada table 2.1).

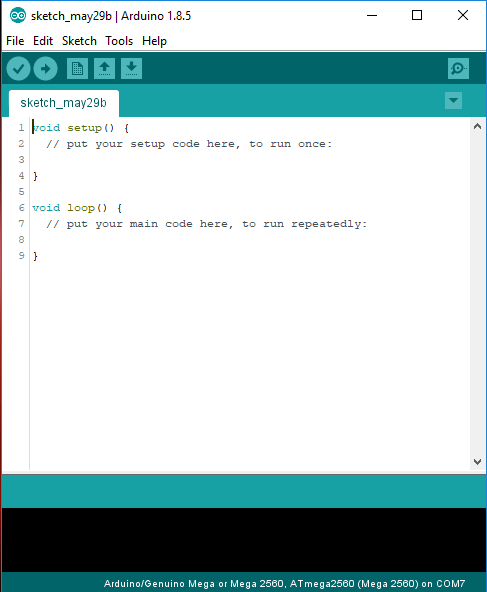
|  |  |
| --- | --- |
| Mikrokontroller | ATmega2560 |
| Operating Voltage | 5V |
| Input Voltage (recommended) | 7 - 12V |
| Input Voltage (limits) | 6 – 20 V |
| Digital I/O Pins | 54 (14 pin merupakan *pulse width modulation output*) |
| Analog Input Pins | 16 |
| DC Current per I/O Pin | 40 mA |
| DC Current for 3.3V Pin | 50 mA |
| Flash memory | 256 KB (8KB digunakan untuk bootloader) |
| SRAM | 8 KB t |
| EPPROM | 4 KB |
| Clock Speed | 16 MHz |

Tabel 2. Data Sheet Arduino Mega2560

## 

## Arduino Integrated Development Environment

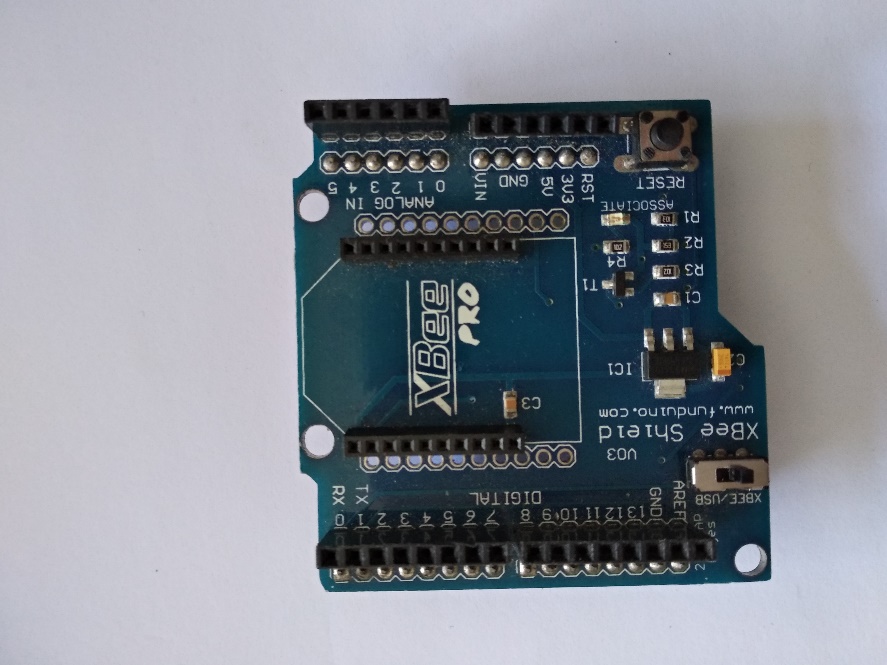
Arduino Integrated Development Environment (IDE) merupakan editor teks untuk menulis kode, area pesan, konsol teks dan memiliki toolbar dengan tombol untuk fungsi umum dan serangkaian menu. Arduino IDE terhubung ke perangkat keras Arduino dan Genuino untuk mengunggah program dan dapat berkomunikasi dengan meraka. 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 mengekspor dan menampilkan kesalahan. Konsol menampilkan output teks oleh Arduino Software (IDE), termasuk pesan kesalahan dan informasi lainnya yang lengkap [11].



Gambar 2. Antarmuka Arduino IDE

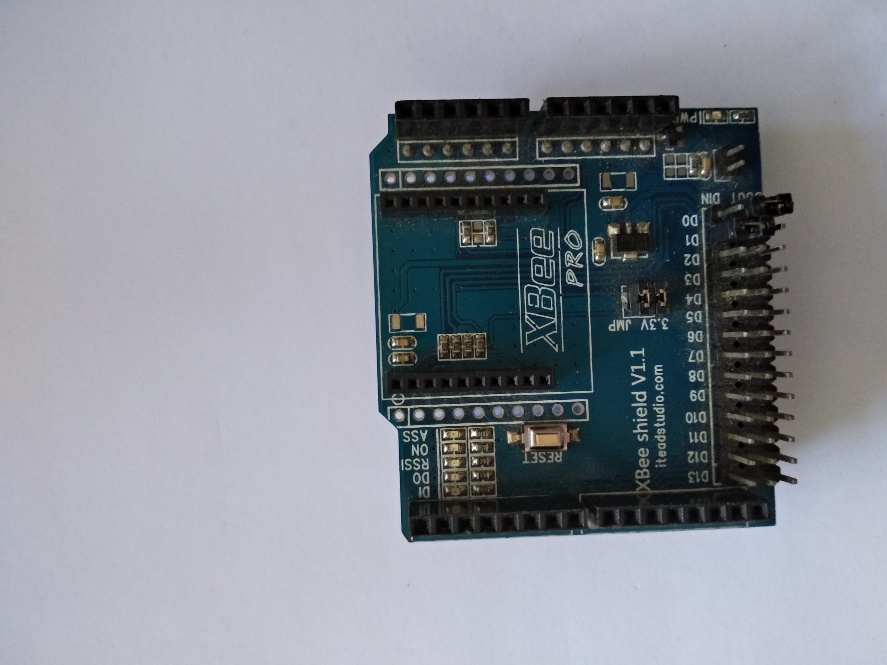
## XBee Shield

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].



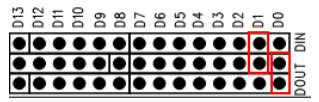
Gambar 2. XBee Shield pabrikan Funduino

Pada penelitian ini, penulis menggunakan dua tipe XBee Shield. Pertama XBee Shield V03 keluaran dari perusahaan Funduino, dimana jumper yang digunakan bertipe switch, sehingga user lebih mudah untuk memilih mode jumper. Terdapat dua jenis jumper, yaitu mode XBee (berfungsi untuk pengiriman data) dan mode USB (berfungsi untuk programming). Perlu di perhatikan ketika memilih mode jumper, ketika ingin melakukan upload code ke arduino jumper harus berada dalam mode USB jika tidak maka code tidak akan bisa terupload. Begitu pula sebaliknya, ketika ingin mengirimkan data via wireless jumper harus berada dalam mode XBee.

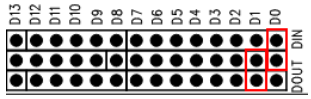


Gambar 2. XBee Shield pabrikan ITead Studio

Adapun XBee Shield jenis kedua merupakan XBee Shield V1.1 keluaran perusahaan ITead Studio. Terdapat dua jenis jumper yaitu mode USB dan mode XBee. Berikut posisi jumper shield tersebut



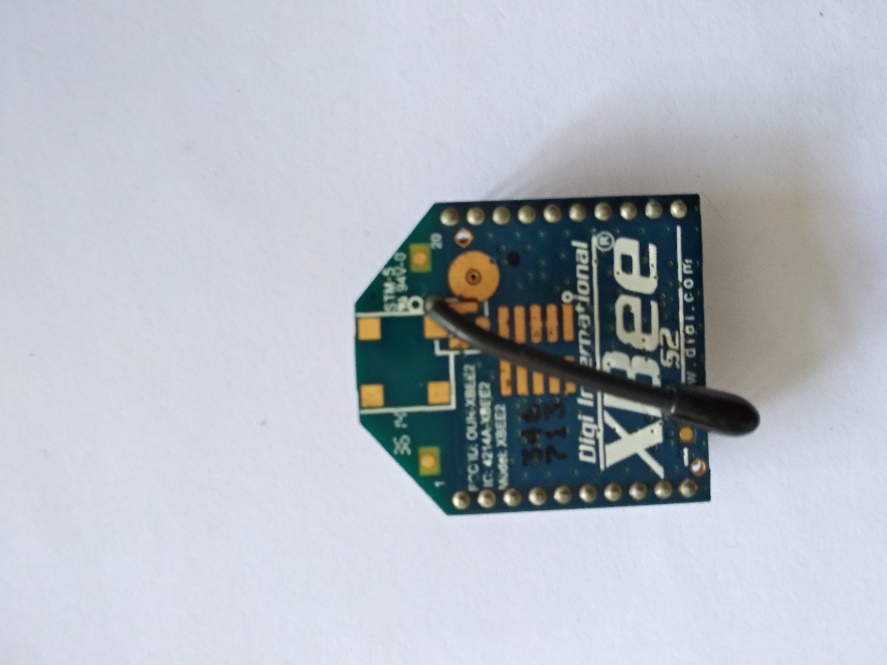
Gambar 2. Konfigurasi Jumper mode USB



Gambar 2. Konfigurasi Jumper mode XBee

## 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 250 kbps. Setiap perangkat output serial dapat mengunakan 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 2. XBee Modul S2

## 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 linkungan 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 menjalanan 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 *default* nya, 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.

## Algoritma 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 kebutuan 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. Proses 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.

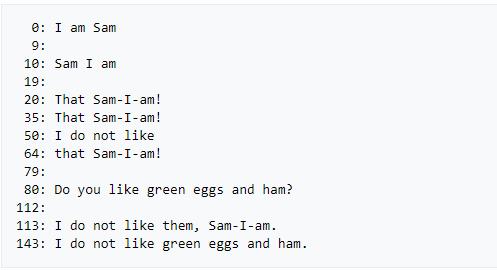
Lempel-Ziv-Storer-Szymanski (LZSS) adalah salah satu jenis algoritma kompresi yang berbasis *dictionary* yang bersifat *lossless* (data dapat di rekonstruksi ulang menjadi data asli). LZSS merupakan salah satu varian dari LZ77 (Lempel Ziv 1977) yang dikembangkan oleh Storer dan Szymansky pada tahun 1982. 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].

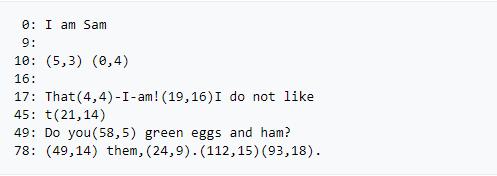
1. Proses Kompresi

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.



Gambar 2. Contoh text asli [18]



Gambar 2. Hasil teks yang ditelah dikompresi [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.

1. Proses Dekompresi

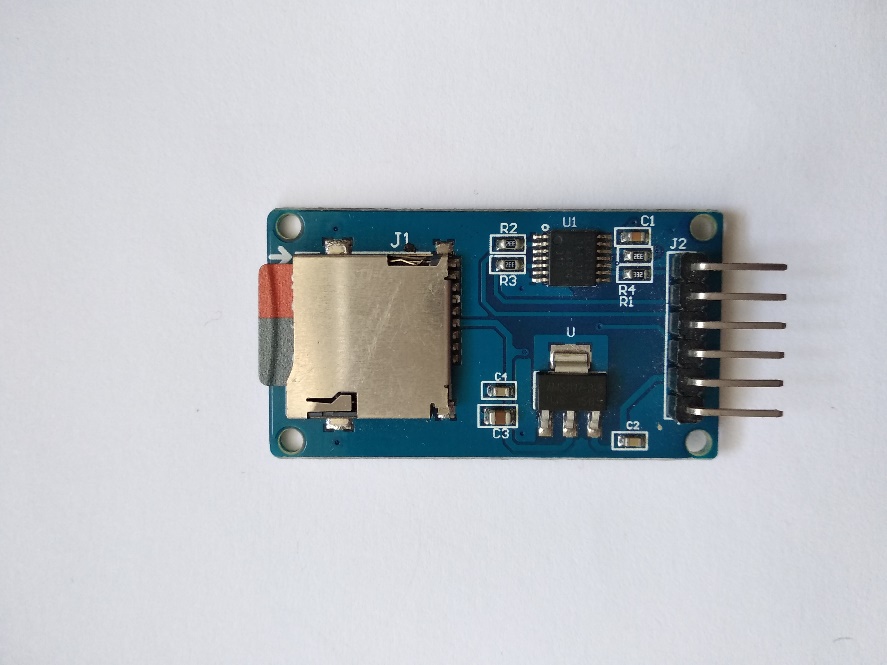
Proses dekompresi dilakukan dengan menguraikan kode secara langsung dengan melibatkan membaca dan menulis ulang hasil tanpa melakukan pencarian apapun. Flag pengkodean dibaca untuk mengetahui karakter mana yang dikodekan. Jika flag menunjukkan bahwa karakter tersebut dikodekan, jumlah karakter dan posisi 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.

## Bahasa Pemrograman C

Bahasa C merupakan salah satu bahasa pemrograman level rendah yang menjadi induk dari bahasa pemrograman modern seperti C++, C#, PHP, Javascript dan masih banyak lagi. Bahasa pemrograman C dibuat pertama kali oleh Dennis M Ritchie dengan tujuan untuk mengembangkan system operasi UNIX yang sebelumnya menggunakan bahasa assembly. Adapun beberapa keunggulan bahasa C dibandingkan dengan bahasa pemrograman yang lain, yaitu: bahasa C termasuk bahasa pemrograman procedural, bahasa C sangat cepat dan efisien, dan Bahasa C merupakan portable language.

## MicroSD Card Adapter

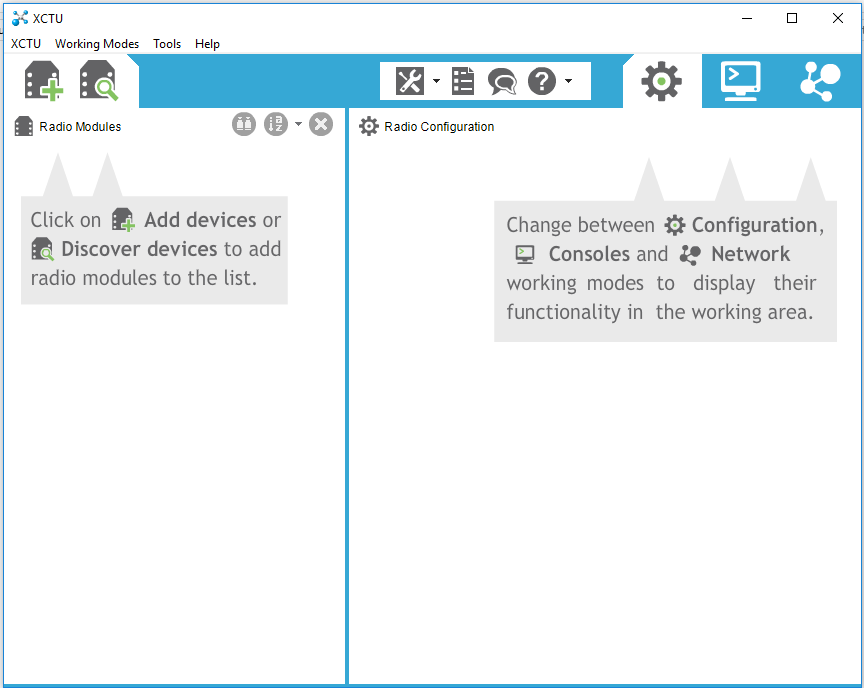
MicroSD Card Adapter ini merupakan modul pembaca kartu MicroSD melalui system file dan SPI antarmuka driver, MCU untuk membaca dan menulis pada kartu microSD. Dengan menggunakan Arduino IDE dan libarary SD card, pengguna dapat menginisialisasi kartu SD card, membaca dan menulisnya.



Gambar 2. MicroSD Card Adapter

## DIGI XCTU

XCTU merupakan aplikasi multi-platform gratis yang dirancang untuk memungkinkan pengembangan modul Digi RF melalui antarmuka grafis yang mudah digunakan. Aplikasi ini dapat digunakan untuk melakukan konfigurasi dan pengujian pada modul XBee® RF. XCTU memiliki semua *tools* yang dibutuhkan pengembang untuk melakukan pengembangan dengan XBee. Terdapat fitur unik didalamnya, seperti tampilan jaringan grafis, yang secara grafis mewakili jaringan XBee yang ada bersama dengan kekuatan sinyal setiap sambungan. Selain itu kita dapat menggunakan XBee API yang secara intuitif membantu dalam membangun dan menafsirkan API *frame* untuk XBee yang menggunakan mode API. Dengan menggunakan aplikasi ini, kita lebih mudah dalam melakukan pengembangan pada platform wireless sensor network yang menggunakan XBee.



Gambar 2. Antarmuka XCTU

*[Halaman ini sengaja dikosongkan]*

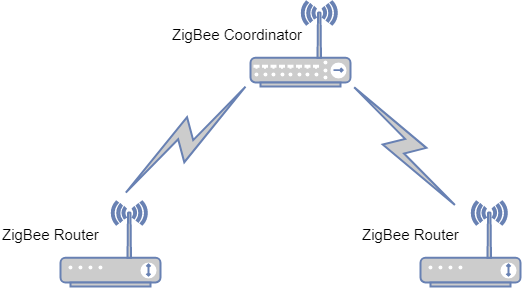
# BAB III PERANCANGAN PERANGKAT LUNAK

Bab ini membahas mengenai dasar dari perancanga sistem yang akan dibangun pada tugas akhir. Perancangan yang dibahas meliputi deskripsi umum sistem, proses perancangan, alur dan implementasinya.



## Deskripsi Umum Sistem

Pada tugas akhir ini akan dibangun suatu sistem kompresi dan dekompresi data pada platform wireless sensor network dengan menggunakan algorima heatshrink. Teknologi wireless sensor network menggunakan mikrokontroller Arduino dan protokol ZigBee sebagai jalur komunikasi. Terdapat tiga node yang akan digunakan, yaitu sebuah node yang bertindak sebagai Coordinator (penerima) dan dua buah node sebagai Router (pengirim). Sebelum data di transfer melalui ZigBee, data yang ada pada node pengirim akan dikompresi terlebih dahulu lalu dikirimkan beserta konfigurasi kompresi yang digunakan. Proses pengiriman akan dilakukan secara bertahap dikarenakan keterbatasan buffer yang ada. Setelah semua data diterima oleh node penerima, data tersebut akan dikompresi sesuai konfigurasi yang telah ditentukan sebelumnya.



Gambar 3. Arsitektur jaringan yang akan dibangun

## Arsitektur Umum Sistem

Teknologi wireless sensor network yang dikembangkan pada tugas akhir ini, menggunakan perangkat XBee S2 yang akan berperan sebagai perangkat yang membantu komunikasi antar node dengan protokol ZigBee sebagai jalur komunikasinya. Pada nantinya, dibutuhkan dua jenis perangkat ZigBee, yaitu node ZigBee Router dan node ZigBee Coordinator. Perangkat node ZigBee Router berfungsi menampung data yang telah ditentukan. Kemudian data tersebut akan dikompresi lalu dikirimkan melalui protokol ZigBee ke node ZigBee Coordinator. Selanjutnya, perangkat node ZigBee Coordinator akan medekompresi data yang telah diterima beserta konfigurasi kompresi yang digunakan.

Gambar arsitektur sistem

Berdasarkan gambar x.x sistem kompresi dan dekompresi data pada platform wireless sensor network memiliki alur proses yang akan dijabarkan sebagai berikut.

1. Pada ZigBee Coordinator, XBee Shield dipasang diatas Arduino Mega 2560, kemudian diatas XBee Shield tersebut dipasang modul XBee S2
2. Pada ZigBee Router, XBee Shield dipasang diatas Arduino Mega 2560, kemudian diatas XBee Shield tersebut dipasang modul XBee S2
3. Pada ZigBee Router, dipasang modul MicroSD Card Adapter berisikan data yang akan dikompresi.

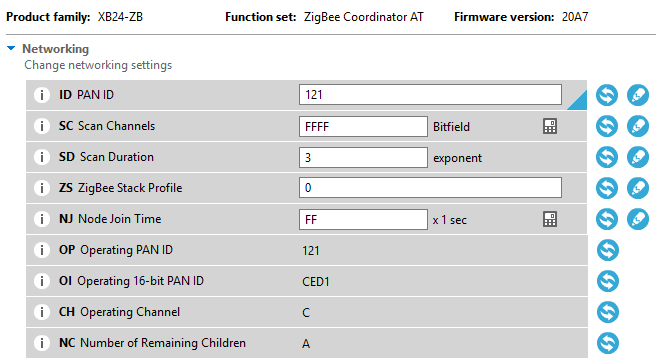
## Perancangan Komunikasi Sistem

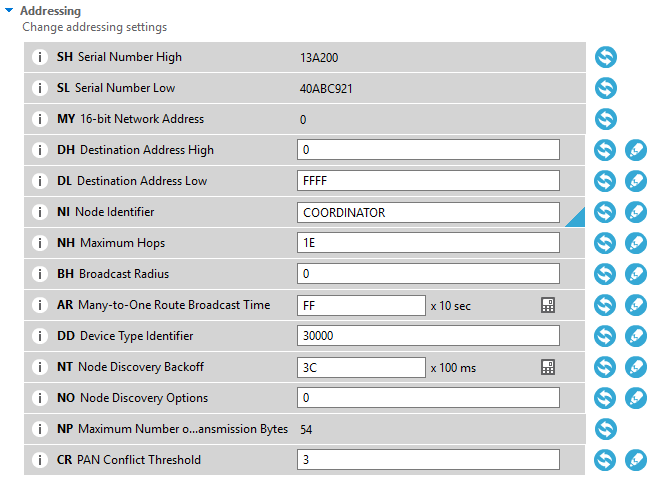
Perancangan komunikasi sistem merupakan salah satu proses penting yang harus dilakukan dalam membangun sistem agar node pada jaringan dapat saling berkomunikasi. Protokol jaringan komunikasi yang digunakan adalah protokol ZigBee, dimana pada jaringan tersebut minimal terdapat sebuah node yang bertindak sebagai Coordinator.

Pada penelitian tugas akhir ini, akan menggunakan dua buah node yang berfungsi sebagai ZigBee Router dan sebuah node sebagai ZigBee Coordinator. Agar setiap node dapat berkomunikasi, harus dilakukan konfigurasi terlebih dahulu terhadap node coordinator dan router dengan memanfaatkan aplikasi XCTU yang disediakan oleh Digi International Inc. Beberapa hal yang harus di perhatikan pada saat melakukan konfigurasi antara lain :

* **Function Set**
* **ID** PAN ID
* **DH** Destination Address High
* **DL** Destination Address Low

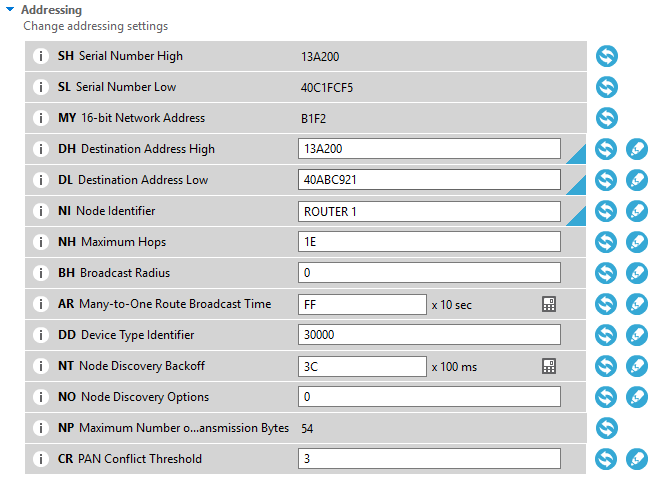
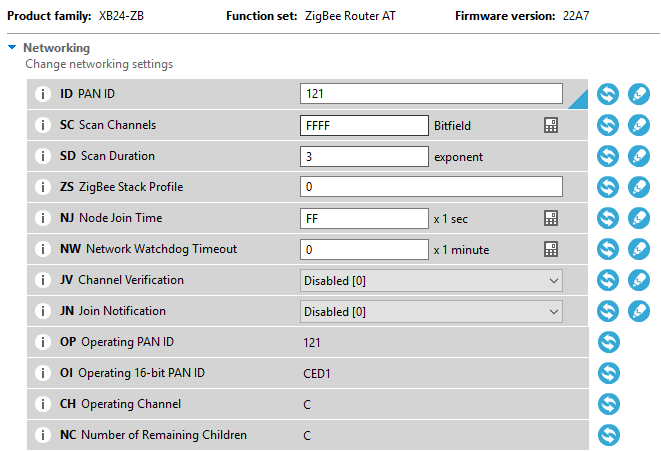
Berikut ini adalah configurasi secara detail pada ZigBee Coordinator dan ZigBee Router





Gambar 3. Konfigurasi pada node ZigBee Coordinator

**Function Set** yang digunakan pada konfigurasi node ZigBee Coordinator adalah **ZigBee Coordinator AT.** Untuk **PAN ID (ID)** nya **121**. PAN ID ini memiliki fungsi yang hampir sama dengan subnet, dimana semua node yang ada pada jaringan ZigBee nilainya harus sama agar dapat saling berkomunikasi. **Destination Address High (DH)** yang digunakan yaitu **0** dan **Destination Address Low (DL)** yaitu **FFFF.** Konfigurasi merupakan*broadcast* sehingga node Coordinator dapat mengirimkan pesan kepada semua node yang ada pada jaringan.



Gambar 3. Konfigurasi pada node ZigBee Router

Setelah mensetting node Coordinator**,** langkah selanjutnya adalah mensetting node Router. **Function Set** yang digunakan pada konfigurasi node ZigBee Router adalah **ZigBee Router AT.** Untuk **PAN ID (ID)** nilainya **121**. **Destination Address High (DH)** yang digunakan adalah **13A200**, dimana nilai ini didapat dari **Serial Number High (SH)** pada node ZigBee Coordinator. Selainnya itu **Destination Address Low (DL)** nilainya **40ABC921** yang merupakan nilai **Serial Number Low (SL)** pada node ZigBee Coordinator. Konfigurasi ini bertujuan agar node ZigBee Router dapat megirimkan pesan pada node ZigBee Coordinator.

## Perancangan Kompresi dan Dekompresi Data

Pada bagian ini akan dijelaskan bagaimana perancangan yang dilakukan sebelum data di kompresi. Untuk melakukan proses kompresi dan dekompresi, pada penelitian ini menggunakan algoritma Heatshrink, dimana algoritma ini berbasis pada algoritma Lempel-Ziv-Storer-Szymanski (LZSS).

Kebutuhan memori merupakan suatu hal yang mendasar dalam melakukan perancangan kompresi dan dekompresi data, sebab jumlah memori yang tersedia pada Arduino sangat terbatas. 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 berkut :

1. Encoding

16 + 2 ∗ byte untuk encoding, ditambah lagi untuk indeks pencarian optional yakni 2 ∗ byte untuk mempercepat pengkodean

1. Decoding

16 + byte untuk decoding, dimana N dapat di atur pada saat pengodean (encoding)

### Cara Kerja Algoritma Heatshrink

Lempel-Ziv-Storer-Szymanski (LZSS) membutuhkan sedikit ruang kerja dimana cache yang diperlukan adalah 2^N byte dari data terakhir (N dapat dikonfigurasi) dan proses dekompresi yang dilakukan cukup sederhana. System yang dijalankan juga cukup sederhana, dump data kedalam, terus putar engkol sampai tidak ada lagi data yang keluar, dump data lebih banyak, ulangi proses tersebut. Notifikasi encoder / decoder ketika akhir input telah tercapai, lakukan sampai selesai. Pada gambar 3.4 menjelaskan diagram alir cara kerja algoritma heatshrink.



Gambar 3. Diagram alir cara kerja algoritma heatshrink

Berikut ini adalah cara kerja pada algoritma Heatshrink :

1. Alokasikan **heatshrink\_encoder** atau **heatshrink\_decoder** pada *state machine* menggunakan fungsi **alloc** atau dapat mengunakan **static alloc** dan panggil fungsi reset untuk memulai inisialisasi
2. Gunakan **sink** untuk memasukkan *input buffer* kedalam *state machine.* Pointer pada input\_size digunakan untuk menunjukkan seberapa banyak byte dari buffer input yang digunakan (jika nilainya 0 maka buffernya penuh)
3. Gunakan **poll** untuk memindahkan output dari *state machine* ke *buffer output.* Pointer pada **output\_size** menunjukkan berapa banyak byte yang dihasilkan dan fungsi return menunjukkan apakah output selanjutnya tersedia (State mechine tidak boleh mengeluarkan data sampai ia menerima input yang cukup)
4. Ulangi langkah 2 dan 3 untuk melakukan stream data melalui *state machine*. Pada saat kompresi data, ukuran input dan output dapat bervariasi secara signifikan. Looping diperlukan untuk buffer input dan output dalam pemrosesan data.
5. Ketika input stream selesai, panggil fungsi **finish** untuk memberitahu bahwa state machine tidak lagi bisa menerima input. Nilai kembalian dari proses yang terlah selesai menunjukkan apakah ada output yang tersisa. Jika ada, panggil fungsi **poll** lagi.
6. Kemudian panggil fungsi **finish** dan flush sisa output hingga selesai, sampai sisa output habis.

### Konfigurasi Algoritma Heatshrink

Heatshrink memilki beberapa opsi konfigurasi yang dapat mempengaruhi penggunaan resource dan seberapa efektif ia dapat memampatkan data. Konfigurasi ini dapat diatur secara dinamis pada saat akan melakukan kompresi dan dekompresi atau dapat pula di setting statis pada file **heatshrink\_config.h.** Adapun konfigurasi yang dimaksudkan adalah sebagi berikut

* **window\_sz2**

Ukuran window menentukan seberapa panjang input yang dapat dicari untuk pola yang berulang. Semakin besar ukuran window maka akan menggunakan memori semakin banyak, tetapi dapat melakukan kompresi lebih efektif dalam mendeteksi pengulangan yang lebih banyak. Sebuah window\_sz2 = 8 akan menggunakan memori 256 byte (2^8), sedangkan window\_sz2 = 10 akan menggunakan memori 1024 byte (2^10). Pengaturan window\_sz yang tersedia adalah antara **4 sampai 15**

* **lookahead\_sz2**

Ukuran lookahead menentukan panjang maksimal unttuk pola berulang yang ditemukan. Jika lookahead\_sz2 adalah 4, ‘a’ 50-bit dari karakter ‘a’ akan direpresentasikan sebagai pola 16-byte berulang (2^4). Jumlah bit yang digunakan unutk ukuran lookahead bersifat tetap, sehingga ukuran lookahead yang besar dapat mengurangi kompresi dengan menambahkan bit yang tidak digunakan ke pola-pola kecil. Pengaturan lookahead\_sz2 yang ada saat ini adalah antara **3 sampai window\_sz – 1.**

* **input\_buffer\_size**

Besar atau kecilnya buffer input yang digunakan untuk decoder ditentukan oleh **input\_buffer\_size.** Ukuran buffer input berdampak pada seberapa banya pekerjaan yang dapat dilakukan decoder dalam satu langkah, dan semakin besar buffer maka memori yang dibutuhkan semakin banyak. Buffer yang sangat kecil (misalnya 1 byte) akan menambah overhead karena banyak banyak melakukan pemanggilan fungsi suspend / resume, akan tetapi **input\_buffer\_size** tidak mempengaruhi seberapa baik dalam melakukan kompresi data.

Pada penelitian ini, konfigurasi yang akan digunakan yaitu sebagai berikut :

* **window\_sz2** menggunakan konfigurasi antara **4 sampai 8**
* **lookahead\_sz2** menggunakan konfigurasi nilai **3 sampai window\_sz2 – 1**
* **input\_buffer\_size** yang digunakan adalah **64**

## Perancangan Pengiriman Data

Pengiriman data adalah suatu hal yang perlu diperhatikan dalam perancangan system ini, sebab keterbatasan buffer yang ada pada protocol ZigBee yaitu sebesar 72 byte menjadi landasan dasar untuk melakukan perancangan pengiriman data yang tepat.

Pada penelitian ini protocol ZigBee menggunakan mode AT atau lebih dikenal mode “Transparan”. Dalam mode AT, data akan segera dikirim ke modul jarak jauh yang diidentifikasi melalui alamat tujuan yang ada pada memori modul XBee. Alamat tujuan dapat di konfigurasi oleh pengguna pada mode Command. Jika XBee menggirimkan data ke Coordinator maka akan di broadcast pada PanID. Informasi paket tidak diperlukan, tetapi prosesnya lebih sederhana, dimana Serial data dikirimkan ke Tx dari satu XBee dan akan diterima oleh Rx tujuan XBee. Mode AT cocok digunakan pada jaringan yang sangat sederhana, karena tidak perlu untuk mengubah alamat tujuan terlalu sering.

Pada gambar 3.5 menjelaskan bagaimana proses pengiriman data dari node ZigBee Router menuju node ZigBee Coordinator



Gambar 3. Diagram alir pengiriman data

## Perancangan Dekompresi Data

Data yang dikirimkan oleh node ZigBee Router akan di terima oleh node ZigBee Coordinator melalui jaringan ZigBee. Potongan – potongan data tersebut akan ditampung terlebih dahulu dan kemudian akan di satukan kembali untuk di proses kembali. Proses dekompresi akan dilakukan ketika data dekompresi yang diterima sudah utuh beserta konfigurasi encoder / decoder yang digunakan. Pada gambar 3.6 menjelaskan bagaimana proses dekompresi data.



Gambar 3. Diagram alir dekompresi data

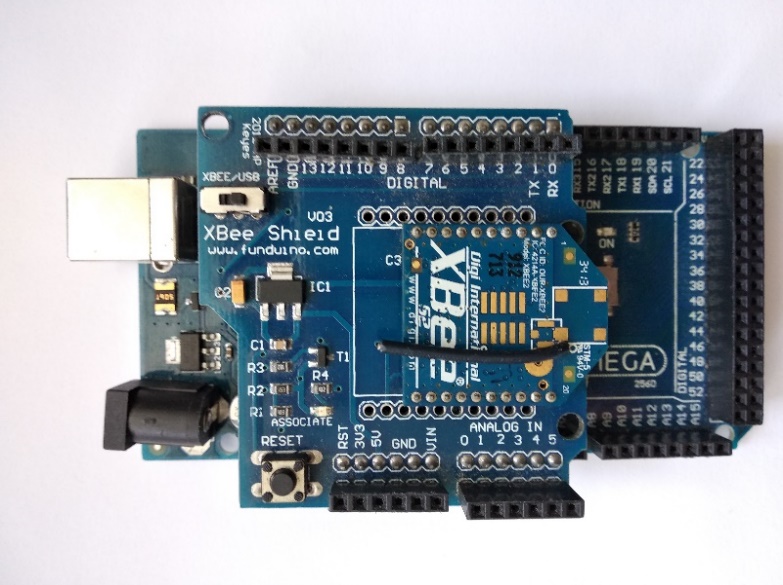
## Perancangan Perangkat Keras

Perancangan perangkat keras secara umum menjelaskan mengenai penempatan perangkat keras yang digunakan dalam membangun sistem, yang mana terdiri dari rangkaian ZigBee Coordinator dan ZigBee Router. Rangkaian perangkat keras pada sistem dapat dilihat pada gambar xx dan gambar xx. Agar sistem dapat berjalan sebagai mestinya, terdapat beberapa komponen yang dibutuhkan antara lain:

1. Tiga buah Arduino Mega
2. Tiga buah baterai 9V
3. Tiga buah XBee Shield
4. Tiga buah XBee S2
5. Dua buah MicroSD Card Adapter
6. Dua buah SD Card

### Perancangan Perangkat ZigBee Coordinator

Pada rangkaian ZigBee Coordinator, XBee Shield V03 pabrikan Funduino menempati tepat di bagian atas Arduino Mega 2560 dengan posisi pin Tx dan Rx yang sama dengan pin Tx dan Rx pada XBee Shield. Kemudian modul XBee S2 Pro di letakkan pada space yang telah disediakan pada XBee Shield. Pastikan kepala modul XBee S2 menghadap arah yang berlawanan dengan konektor USB.

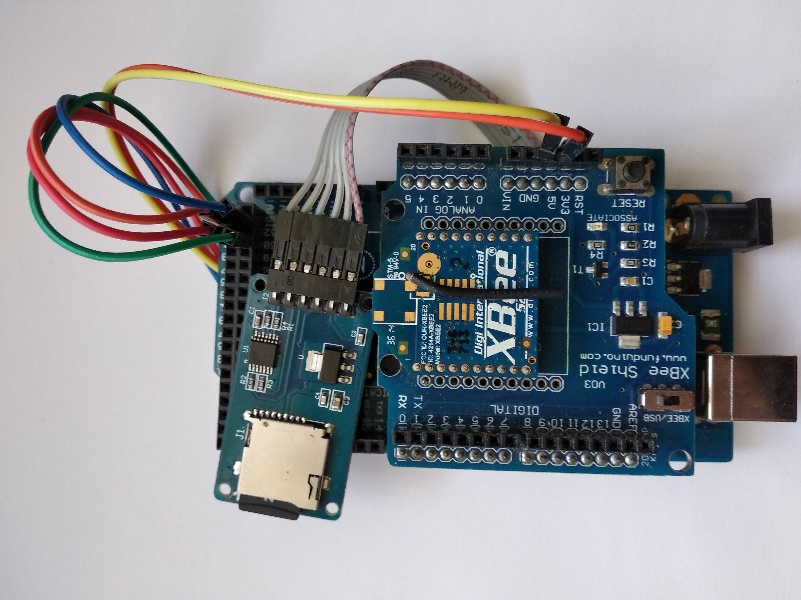


Gambar 3. Node ZigBee Coordinator

### Perancangan Perangkat ZigBee Router

Pada rangkaian ZigBee Router posisinya hampir sama dengan rangkaian ZigBee Coordinator. Xbee Shield diletakkan diatas Arduino Mega dengan posisi pin Tx dan Rx yang sama. Selain itu letakkan modul XBee S2 pada tempat yang telah disediakan. Pada perangkat ZigBee Router membutuhkan MicroSD Card Adapter yang berfungsi untuk menampung data yang akan di kompresi. Pastikan sudah terdapat SD Card pada modul tersebut, kemudian sambungkan pin yang terdapat pada MicroSD Card Adapter kepada pin yang sudah ditentukan, berikut ini rincian pinnya :

* Pin **CS** dihubungkan dengan pin **digital 53**
* Pin **SCK** dihubungkan dengan pin **digital 52**
* Pin **MOSI** dihubungkan dengan pin **digital 51**
* Pin **MISO** dihubungkan dengan pin **digital 50**
* Pin **VCC** dihubungkan dengan pin **tegangan 5V**
* Pin **GND** dihubungkan dengan pin **GND**



Gambar 3. Node ZigBee Router disertai MicroSD

*[Halaman ini sengaja dikosongkan]*

# BAB IV IMPLEMENTASI

Bab ini membahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi berupa *pseudocode* untuk membangun program. Cakupan implementasi dari perancangan system ini meliputi perangkat node ZigBee Router yang bertugas untuk melakukan kompresi data dan mengirimkan hasilnya ke perangkat node ZigBee Coordinator, kemudian akan di dekompresi berdasarkan konfigurasi yang telah diterima sebelumnya. Bahasa pemrograman yang digunakan adalah Bahasa pemrograman C.



## Lingkungan Implementasi

Lingkungan implementasi merupakan suatu lingkungan dimana sistem akan dibangun. Untuk mempermudah penjelasan, lingkungan impelementasi akan terbagi menjadi dua bagian. Pembahasan pertama mengenai lingkungan implementasi perangkat keras dan pembahasan kedua mengenai lingkungan implementasi perangkat lunak.

### Lingkungan Implementasi Perangkat Keras

Pada bagian ini akan dibahas mengenai perangkat keras apa saja yang dibutuhkan untuk membangun system. Linkungan implementasi perangkat keras dari system yang akan dibangun secara lebih lengkap dijelaskan pada Tabel 4.1 dibawah ini

Tabel 4. Linkungan implementasi perangkat keras

|  |  |
| --- | --- |
| Perangkat | Detail Perangkat |
| Perangkat Komputer | Model :   * Lenovo Y410P |
| Manufaktur :   * Lenovo |
| Processor :   * Intel® Core™ i7-4700MQ (2.40GHz 1600MHz 6MB) |
| Memori :   * 8GB PC3-12800 DDR3L SDRAM 1600 MH |
| Perangkat Mikrokontroler | Mikrokontroler :   * ATmega2560 |
| Model :   * Arduino Mega 2560 |
| Tegangan :   * 5 V |
| Memori Flash :   * 256 KB (8KB digunakan untuk bootloader) |
| SRAM :   * 8 KB |
| Perangkat XBee Shield | Model:   * XBee Shield V0.3 (a) * XBee Shield V1.1 (b) |
| Manufaktur :   * Funduino (a) * ITead Studio (b) |
| Tipe jumper :   * Switch (a) * Pasang lepas (b) |
| Perangkat Modul XBee | Model :   * XBee S2 |
| Manufaktur :   * Digi International Inc. |

### Lingkungan Implementasi Perangkat Lunak

Pada bagian ini akan dibahas mengenai perangkat lunak apa saja yang dibutuhkan untuk membangun sistem. Linkungan impementasi perangkat lunak dari sistem yang akan dibangun secara lebihh detail akan dijelaskan pada Tabel 4.2 dibawah ini.

Tabel 4. Lingkungan implementasi perangkat lunak

|  |  |
| --- | --- |
| Perangkat | Detail Perangkat |
| Perangkat Lunak | Sistem Operasi :   * Microsoft Windows 10 Pro 64-bit |
| Software Arduino :   * Arduino IDE 1.8.5 |
| Software XBee :   * DIGI XCTU 6.3.13 |

## Implementasi Perangkat Keras

Implementasi perangkat keras untuk penelitian tugas akhir ini adalah sebagai berikut :

* 3 (tiga) buah Arduino Mega
* 3 (tiga) buah XBee Shield
* 3 (tiga) buah XBee S2
* 2 (dua) buah MicroSD Card Adapter
* 2 (dua) buah SD Card
* 3 (tiga) buah baterai 9 volt

Dua buah Arduino Mega akan berperan sebagai node router yang akan mengirimkan data ke sebuah node coordinator. Data yang akan di kirimkan diambil dari SD Card, dimana data telah didefinisikan sebelumnya. Kemudian data tersebut akan di kompresi pada node Zigbee Router. Setelah data selesai dikompresi, data kemudian dikirimkan ke Node ZigBee Coordinator beserta konfigurasi yang telah digunakan. Perancangan serta implementasi peranagkat node ZigBee Coordinator dan Node ZigBee Router akan dijabarkan lebih mendetail pada point xxx dan xxx

### Perangkat Node ZigBee Coordinator

Perangkat node ZigBee Coordinator tersusun dari Arduino Mega, XBee Shield, dan modul XBee S2. Zigbee Coordinator memiliki peranan dalam nenerima data dari ZigBee Router dan akan mendekompresi data yang telah diterima. Pada gambar xx memperlihatkan perancangan dari node ZigBee Coordinator, sedangkan pada gambar xx merupakan implementasi dari node ZigBee Coordinator.

GAMBAR Perancangan Node ZigBee Coordinator

GAMBAR Implementasi

### Perangkat Node ZigBee Router

Perangkat node ZigBee Router tersusun dari Arduino Mega, XBee Shield, modul XBee S2 dan MicroSD Card Adapter. ZigBee Router memiliki peranan dalam mengambil data dari CD Card, melakukan kompresi data dan mengirimkan data tersebut menuju node ZigBee Coordinator. Pada gambar xx memperlihatkan perancangan dari node ZigBee Router dan gambar xx merupakan implementasi dari node ZigBee Router.

GAMBAR Perancangan Node ZigBee Router

GAMBAR Implementasi

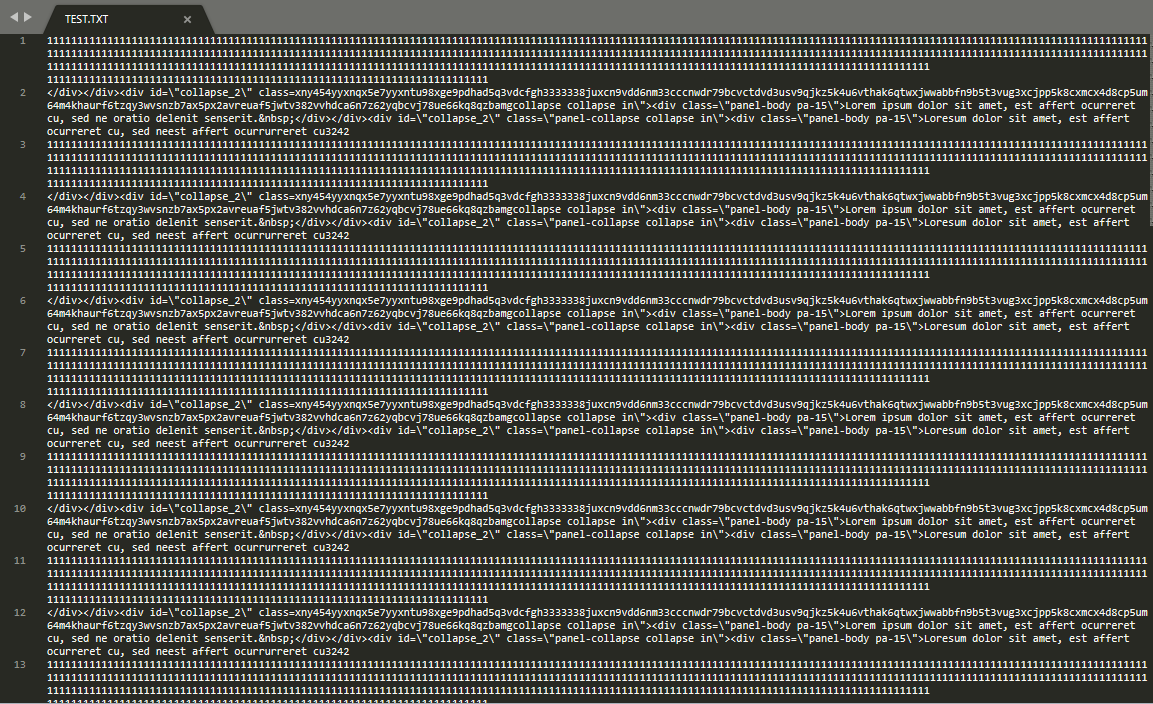
## Implementasi Inisialisasi Data Pada SD Card

Data yang akan dikompresi berupa string diletakkan pada SD Card. Sebelum melakukan kompresi data, terlebih dahulu kita harus menginisialisasi data. Pada penelitian ini terdapat empat jenis data yang akan dikompresi, dimana masing – masing data memiliki panjang yang berbeda. Data tersebut akan disimpan dalam bentuk file txt. Berikut ini adalah keempat data tersebut :

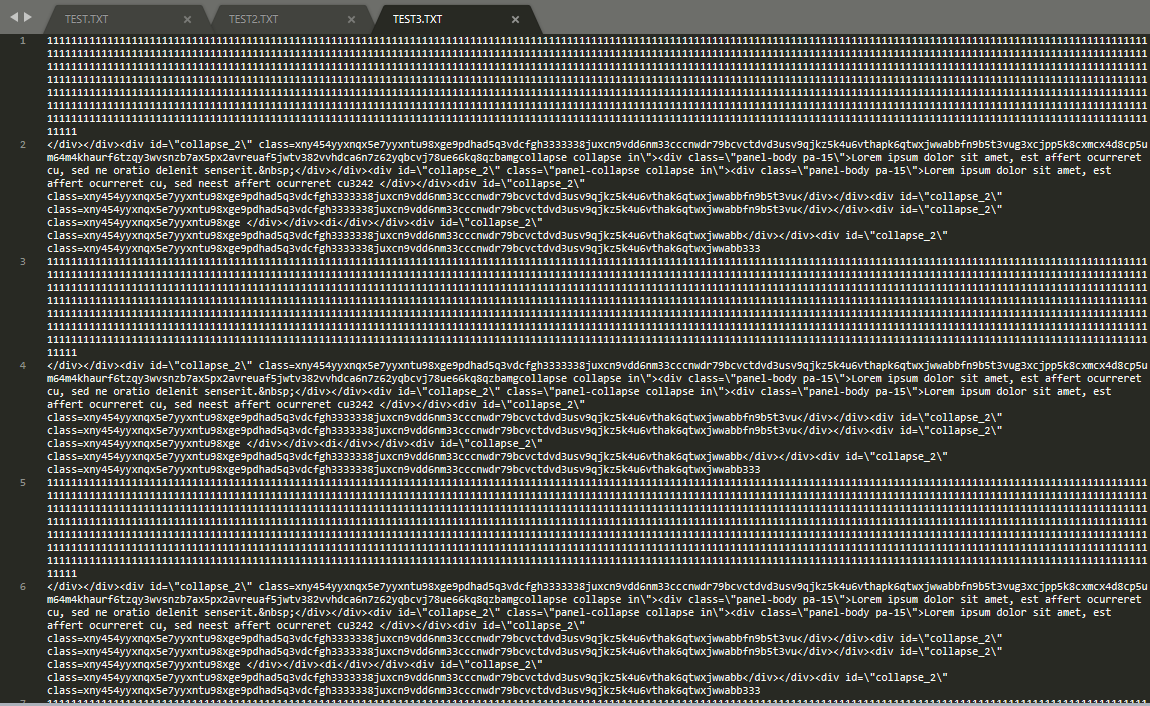
1. Data string dengan panjang 584 karakter
2. Data string dengan panjang 980 karakter
3. Data string dengan panjang 1280 karakter
4. Data string dengan panjang 1345 karakter

Setiap file menyimpan data dengan panjang yang sama. Sebagai contohnya file data\_584.txt menyimpan data dengan panjang 584 karakter, dimana setiap baris data yang ada pada file tersebut panjangnya sama yaitu 584 karakter.

Pada gambar xx merupakan contoh file txt dengan Panjang string 584 karakter per baris. Pada gambar xx merupakan contoh file txt dengan panjang string 980 karakter per baris. Pada gambar xx merupakan contoh file txt dengan panjang string 1280 karakter per baris. Dan pada gambar xx merupakan contoh file txt dengan Panjang string 1345 karakter per baris.



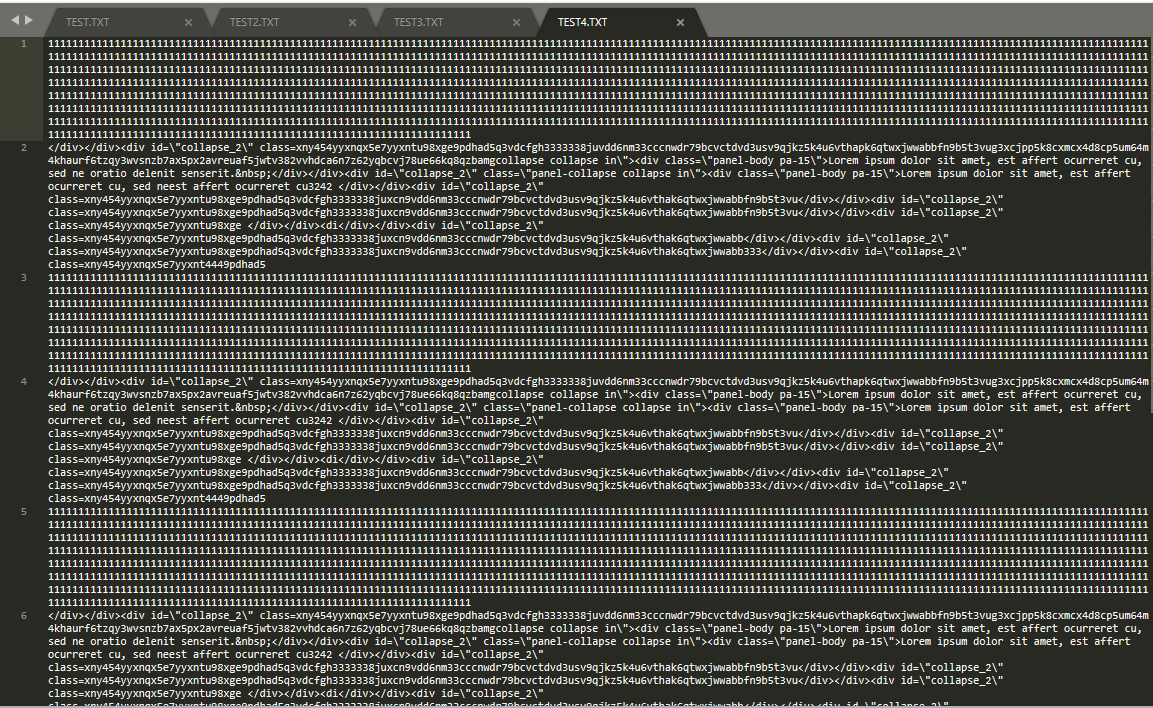
Gambar 4. Data dengan panjang 584 karakter per baris



Gambar 4. Data dengan panjang 980 karakter per baris



Gambar 4. Data dengan panjang 1280 karakter per baris



Gambar 4. Data dengan panjang 1345 karakter per baris

## Implementasi Membaca Data dari SD Card

Setelah data dibagi kedalam masing-masing file, maka data akan siap di kompresi. Pada node Router data akan diambil dari SD Card per baris. Kemudian data tersebut di akan kompresi dengan konfigurasi yang telah ditentukan.

Pseudocode fungsi membaca data dari SD Card

## Implementasi Setting Konfigurasi Encoder / Decoder

Implementasi setting konfigurasi encoder / decoder bertujuan untuk melakukan setting konfigurasi yang akan digunakan encoder dalam proses kompresi data. Adapun konfigurasi yang akan digunakan yaitu sebagai berikut :

* **window\_sz2** menggunakan konfigurasi antara **4 sampai 8**
* **lookahead\_sz2** menggunakan konfigurasi nilai **3 sampai window\_sz2 – 1**
* **input\_buffer\_size** yang digunakan adalah **64**

Pseudocode fungsi set konfigurasi

## Implementasi Kompresi Data

Implementasi kompresi data yang dilakukan pada penelitian ini menggunakan library Heatshrink yang merupakan salah satu algoritma kompresi data yang berbasis pada algoritma Lempel-Ziv-Storer-Szymanski (LZSS).

Pseudocode fungsi kompresi data

## Implementasi Pengiriman Data

Pada penelitian ini, pengiriman data yang dilakukan oleh node ZigBee Router menggunakan mode Router AT. Proses pengiriman data memanfaatkan kelas Serial yang ada pada Arduino. Data hasil kompresi yang akan dikirimkan akan dipecah menjadi beberapa bagian kecil. Dalam satu kali pengiriman jumlah data yang dapat di kirimkan adalah sebanyak 16 buah, dimana terdapat delay 3 detik untuk setiap pengiriman data. Setelah data kompresi dikirimkan semua, settingan konfigurasi encoder / decoder yang digunakan juga akan dikirimkan ke node ZigBee Coordinator.

Pseudocode fungsi pengiriman data

## Implementasi Dekompresi Data

Potongan – potongan data yang dikirimkan oleh node ZigBee Router akan disatukan kembali pada Node ZigBee Coordinator. Setelah data hasil kompresi utuh, maka proses dekompresi data akan siap dilakukan dengan konfigurasi encoder / decoder yang dikirimkan juga oleh node ZigBee Router.

Pseudocode fungsi dekompresi data

*[Halaman ini sengaja dikosongkan]*

# BAB V PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada Tugas Akhir yang telah dikerjakan. Uji coba yang akan dilakukan secara garis besar terdiri dari uji coba fungsionalitas dan uji coba performa. Mekanisme uji coba dilakukan dengan menjalankan serangkaian scenario yang telah ditentukan. Pengujian fungsionalitas meliputi uji coba setiap bagian perangkat keras yang dirangkai pada Arduino dan uji coba keseluruhan sistem. Sedangkan pengujian performa meliputi efektifitas hasil kompresi, waktu yang dibutuhkan unutk kompresi dan dekompresi serta energi yang dapat dihemat oleh system. Bagian akhir dari bab ini akan membahas mengenai evaluasi dari serangkaian uji coba yang telah dilakukan.



## Lingkungan Uji Coba

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1.

Tabel . Spesifikasi Lingkungan Uji Coba

|  |  |  |
| --- | --- | --- |
| Perangkat Keras | | |
| ZigBee Coordinator | Rangkaian dari : |
| - Arduino Mega 2560 |
| - XBee Shield |
| - XBee Modul S2 |
| - Baterai 9 Volt |
| ZigBee Router | Rangkaian dari : |
| - Arduino Mega 2560 |
| - XBee Shield |
| - XBee Modul S2 |
| - MicroSD Card Adapter |
| - Baterai 9 Volt |
| - SD Card |
| Perangkat Lunak | | |
| Sistem Operasi | Microsoft Windows 10 Pro 64-bit |
| Software Arduino | Arduino IDE 1.8.5 |
| Software XBee | DIGI XCTU 6.3.13 |

## Uji Coba Fungsionalitas

Untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan, maka dibutuhkan pengujian fungsionalitas.

Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi benar-benar diimplementasikan dan bekerja sebagaimana seharusnya. Pengujian juga dilakukan untuk mengetahui kesesuaian setiap tahapan atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Pengujian dilakukan dengan metode *black-box*.

### Skenario Uji Coba Membaca Data dari SD Card

Skenario pengujian fungsionalitas digunakan untuk memberikan tahap-tahap dalam pengujian sistem. Skenario tertera pada Tabel 5.2.

Tabel . Skenario Uji Coba Membaca Data dari SD Card

|  |  |
| --- | --- |
| ID | UJ – 01 |
| Nama | Uji Coba Membaca Data dari SD Card |
| Tujuan Uji Coba | Menguji fungsionalitas system untuk membaca data dari SD Card |
| Kondisi Awal | Node ZigBee Router diaktifkan |
| Skenario | Node ZigBee Router membaca data dari SD Card, data dibaca per baris kemudian dilakukan kompresi data. Kegiatan ini akan terus berulang hingga data pada baris terakhir. |
| Masukan | Data raw yang akan dikompresi |
| Keluaran |  |
| Hasil Uji Coba |  |

### Skenario Uji Coba Komunikasi Node ZigBee Router dengan Node ZigBee Coordinator

|  |  |
| --- | --- |
| ID | UJ – 02 |
| Nama | Uji Coba Komunikasi Node ZigBee Router dengan Node ZigBee Coordinator |
| Tujuan Uji Coba | Menguji fungsionalitas sistem untuk mengirimkan data dari node ZigBee Router menuju node ZigBee Coordinator |
| Kondisi Awal | Node ZigBee Router dan Node ZigBee Coordinator diaktifkan |
| Skenario | Node ZigBee Router mengirimkan data ke node Node ZigBee Coordinator |
| Masukan | Data raw yang akan dikompresi |
| Keluaran |  |
| Hasil Uji Coba |  |

### Skenario Uji Coba Kompresi Data

|  |  |
| --- | --- |
| ID | UJ – 03 |
| Nama | Uji Coba Kompresi Data |
| Tujuan Uji Coba | Menguji fungsionalitas system untuk mengkompresi data |
| Kondisi Awal | Node ZigBee Router diaktifkan, data raw yang akan di kompresi sudah disiapkan |
| Skenario | 1. Menginisialisasi Encoder / Decoder dengan konfigurasi HS (4,3) 2. Node Router membaca data dari *sd card* per baris 3. Melakukan kompresi data 4. Ulangi langkah 2 dan 3 sampai data pada baris terakhir 5. Mengulangi pengujian dengan konfigurasi HS (5,4) 6. Mengulangi pengujian dengan konfigurasi HS (6,5) 7. Mengulangi pengujian dengan konfigurasi HS (7,5) 8. Mengulangi pengujian dengan konfigurasi HS (8,6) |
| Masukan | Data raw yang akan dikompresi |
| Keluaran | Data hasil kompresi |
| Hasil Uji Coba |  |

### Skenario Uji Coba Pengiriman Data

|  |  |
| --- | --- |
| ID | UJ – 03 |
| Nama | Uji Coba Pengiriman Data |
| Tujuan Uji Coba | Menguji fungsionalitas system untuk mengirimkan data hasil kompresi dari node ZigBee Router menuju node ZigBee Coordinator |
| Kondisi Awal | Node ZigBee Router diaktifkan, data sudah di kompresi |
| Skenario | 1. Node ZigBee Router mendapatkan data hasil kompresi  2. Node ZigBee Router mengirimkan data hasil kompresi dengan memecahnya menjadi 16 data dalam setiap kali kirim  3. Delay 3 detik  4. Ulangi langkah 2 dan 3 sampai data hasil kompresi dikirim semuanya  5. Kirim konfigurasi yang digunakan |
| Masukan | Data yang sudah di kompresi |
| Keluaran | Data kompresi beserta konfigurasi encoder / decodernya |
| Hasil Uji Coba |  |

### Skenario Uji Coba Dekompresi Data

|  |  |
| --- | --- |
| ID | UJ – 03 |
| Nama | Uji Coba Dekompresi Data |
| Tujuan Uji Coba | Menguji fungsionalitas system untuk melakukan dekompresi data |
| Kondisi Awal |  |
| Skenario | 1. Node ZigBee Coordinator menerima data yang di kirim oleh node ZigBee Router  2. Data yang di terima di bentuk kembali  3. |
| Masukan |  |
| Keluaran |  |
| Hasil Uji Coba |  |

### Hasil Pengujian

Hasil uji fungsionalitas yang telah dilakukan berdasarkan skenario sebelumnya, menunjukkan bahwa semua fungsionalitas permainan berjalan dengan baik dan sesuai dengan sebagaimana mestinya yang telah dibuat pada tahap perancangan. Hasil uji fungsionalitas dapat dilihat pada Tabel 5.3.

Tabel . Hasil pengujian fungsionalitas

|  |  |  |
| --- | --- | --- |
| No | Pengujian | Hasil Pengujian |
| 1 | *Main Menu* dan *Stage Selection* |  |
| 2 | Pergerakan karakter pemain |  |
| 3 | Perpindahan waktu |  |
| 4 | Mendapatkan poin |  |
| 5 | Penanaman bibit |  |
| 6 | Menang dan Kalah |  |

## Uji Coba Performa

Pengujian penerapan algoritma Hunt-and-Kill pada game Plant the Future mencakup seberapa efektif algoritma tersebut digunakan sebagai penyusun *puzzle* yang dapat diselesaikan. Pengujian ini dilakukan oleh satu orang pengguna.

### Uji Coba Efektifitas Hasil Kompresi Data

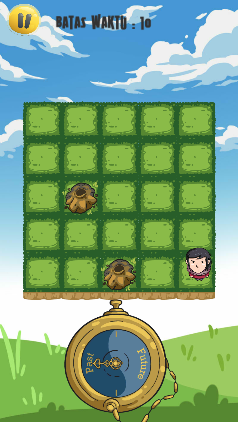
Pengujian dilakukan pada 30 stage pada game ‘Plant the Future’ yang menerapkan algoritma Hunt-and-Kill. Pengujian dilakukan dengan memainkan setiap stage sebanyak lima kali secara beruntun dan diamati apakah stage yang dibuat dapat diselesaikan atau tidak. Penulis secara manual menguji stage yang merupakan *puzzle* hasil dari penerapan algoritma Hunt-and-Kill.

### Uji Coba Waktu Untuk Kompresi dan Dekompresi

### Uji Coba Konsumsi Energi

### Hasil Pengujian

Hasil yang diuji merupakan tampilan dari realisasi array menjadi tampilan permainan. Salah Satu contoh tampilan, hasil dari penggunaan algoritma Hunt-and-Kill dapat dilihat pada gambar 5.10. Seluruh tampilan dari stage yang diuji dapat dilihat pada bagian lampiran tugas akhir ini.



Gambar . Tampilan stage pengujian

Pengujian dilakukan pada seluruh stage yang menerapkan algoritma Hunt-and-Kill yaitu berjumlah 30 stage dengan dimensi puzzle yang berbeda-beda. Hasil pengujian 30 stage sebanyak lima kali masing-masingnya dengan total 150 stage uji coba, menghasilkan hasil seperti ditunjukkan pada Table 5.4.

Tabel . Hasil pengujian

|  |  |  |  |
| --- | --- | --- | --- |
| No | Stage | Jumlah Pengujian Stage | Jumlah berhasil diselesaikan |
| 1 | 2 | 5 | 5 |
| 2 | 3-1 | 5 | 5 |
| 3 | 3-2 | 5 | 5 |
| 4 | 4-1 | 5 | 5 |
| 5 | 4-2 | 5 | 5 |
| 6 | 4-3 | 5 | 5 |
| 7 | 5-1 | 5 | 5 |
| 8 | 5-2 | 5 | 5 |
| 9 | 5-3 | 5 | 5 |
| 10 | 5-4 | 5 | 5 |
| 11 | 6-1 | 5 | 5 |
| 12 | 6-2 | 5 | 5 |
| 13 | 6-3 | 5 | 5 |
| 14 | 6-4 | 5 | 5 |
| 15 | 6-5 | 5 | 5 |
| 16 | 7-1 | 5 | 5 |
| 17 | 7-2 | 5 | 5 |
| 18 | 7-3 | 5 | 5 |
| 19 | 7-4 | 5 | 5 |
| 20 | 7-5 | 5 | 5 |
| 21 | 8-1 | 5 | 5 |
| 22 | 8-2 | 5 | 5 |
| 23 | 8-3 | 5 | 5 |
| 24 | 8-4 | 5 | 5 |
| 25 | 8-5 | 5 | 5 |
| 26 | 9-1 | 5 | 5 |
| 27 | 9-2 | 5 | 5 |
| 28 | 9-3 | 5 | 5 |
| 29 | 9-4 | 5 | 5 |
| 30 | 9-5 | 5 | 5 |

Dari hasil pengujian didapatkan bahwa dari 150 stage yang diuji seluruhnya berhasil diselesaikan. Sehingga Hunt-and-Kill dapat dikatakan berhasil diterapkan pada permainan ‘Plant the Future’ sebagai algoritma penyusun stage.

# BAB VI KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.



## Kesimpulan

Dari proses penerapan algoritma Hunt-and-Kill untuk perancangan *puzzle* pada game Plant the Future dapat diambil kesimpulan sebagai berikut:

1. ‘Plant the Future’ berhasil dirancang dengan baik menerapkan pergerakan karakter permainan yang diadopsi dari permainan yang mirip dan menggunakan *gameplay* aturan permainan yang dirancang sendiri oleh penulis.
2. Aturan perancangan puzzle menggunakan algoritma Hunt-and-Kill diterapkan dengan memodifikasi algoritma yaitu menambahkan beberapa kondisi untuk menyesuaikan Hunt-and-Kill dan *gameplay* game ‘Plant the Future’.
3. Penerapan algoritma Hunt-and-Kill dalam perancangan puzzle dalam game diterapkan dalam *game engine*, dan permainan dapat berjalan sebagaimana mestinya dalam perangkat mobile.

## Saran

Saran yang diberikan terkait pengembangan pada Tugas Akhir ini berdasarkan perancangan, implementasi, dan uji coba adalah:

* + - 1. Menambah parameter tingkat kesulitan dalam penyusunan *puzzle* stage.
      2. Memperbanyak jumlah pengujian terhadap setiap stage.
      3. Untuk pengembangan lebih lanjut disarankan agar menggabungkan penggunaan algoritma Hunt-and-Kill dengan algoritma lain untuk menciptakan tingkat kesulitan dengan parameter yang lebih banyak dan terarah, bukan hanya dari sisi dimensi stage.
      4. Menambah fitur mode permainan dan power up.
      5. Petunjuk permainan lebih diperjelas.

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].

1. 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.

1. 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.
2. 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.
3. P. Ghaffariyan, “An Effective Data Aggregation Mechanism for Wireless Sensor Networks”, 6th International Conference on Wireless Communications Networking and Mobile Computing, 2010.
4. 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
5. 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.
6. Shu Yinbiao dan Kang Lee, “Internet of Things: Wireless Sensor Networks”,

International Electrotechnical Commission, White Paper, 2014.

1. Bröring A, “New Generation Sensor Web Enablement”, PMC, 2011. [Daring]. Tersedia: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231615/>[Diakses 2-Jan-2018]
2. “Arduino.” Wikipedia bahasa Indonesia, eksiklopedia bebas. [Daring]. Tersedia: https://id.wikipedia.org/wiki/Arduino [Diakses 2-Jan-2018]
3. Arduino, “Arduino Software (IDE).” [Daring]. Tersedia: <https://www.arduino.cc/en/Guide/Environment>[Diakses 3-Jan-2018]
4. “Shield Arduino.” [Daring]. Tersedia: <http://www.belajararduino.net/shield-arduino/> [Diakses 3-Jan-2018]
5. “Zigbee Networking with XBee Series 2 and Seeed's Products.” [Daring]. Tersedia: <http://wiki.seeed.cc/Zigbee_Networking_with_XBee_Series_2_and_Seeeds_Products/> [Diakses 3-Jan-2018]
6. 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
7. Scott Vokes, “heatshrink: An Embedded Data Compression Library.” [Daring]. Tersedia: <https://spin.atomicobject.com/2013/03/14/heatshrink-embedded-data->[compression/](https://spin.atomicobject.com/2013/03/14/heatshrink-embedded-data-compression/) [Diakses 3-Jan-2018]
8. “ Algoritma Kompresi LZSS” [Daring]. Tersedia: <https://mahfudharun.wordpress.com/2012/06/14/algoritma-kompresi-lzss/>[Diakses 6-Jan-2018]
9. Adnan Ozsoy dan Martin Swany, “ CULZSS: LZSS Lossless Data Compression on CUDA”, IEEE International Conference on Cluster Computing (CLUSTER), 2011
10. “Lempel–Ziv–Storer–Szymanski.” Wikipedia, the free encyclopedia . [Daring]. Tersedia: https://en.wikipedia.org/wiki/Lempel–Ziv–Storer–Szymanski [Diakses 6-Jan-2018]
11. Arduino, “Memory.” [Daring]. Tersedia: <https://www.arduino.cc/en/Tutorial/Memory>[Diakses 6-Jan-2018]
12. 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]

*[Halaman ini sengaja dikosongkan]*

BIODATA PENULIS

Penulis lahir di Pekanbaru, 25 September 1996, merupakan anak pertama dari tiga bersaudara. Dalam menjalani pendidikan semasa hidup, penulis menempuh pendidikan di TK An-Nur Pekanbaru, SDN 001 Sail Pekanbaru, SMPN 4 Pekanbaru, SMAN 8 Pekanbaru dan S1 Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS) pada rumpun Interaksi Grafika dan Seni (IGS).

Selama menjadi mahasiswa, penulis ikut dalam Himpunan Mahasiswa Teknik Komputer Informatika ITS, dalam departemen Ristek pada tahun kedua, dilanjutkan dengan tahun ketiga di departemen Teknologi sebagai staf ahli, staf ahli departemen Syiar KMI(Keluarga Muslim Informatika) dan staf Schematics 2016 di biro NPC. Penulis telah menghasilkan beberapa karya aplikasi permainan diantaranya: ‘Logic Bullet’,’Logic Bullet Extented’, ‘Get the Trash’, ‘EnCollect’ dan ‘Plant the Future’. Berkat kegemarannya di ranah game development, penulis berkesempatan mengikuti beberapa kompetisi bersama tim fragments. Penulis mengikuti beberapa kompetisi seperti ajang Gemastik 9 yang diselenggarakan di Universitas Indonesia berhasil menjadi finalis dengan membawakan game ‘Logic Bullet’, mendapatkan juara 1(satu) pada ajang Game Development MAGE 2017 dengan membawakan game ‘Logic Bullet Extended’, mendapatkan juara 1(satu) pada ajang Game Development FTIF Festival 2017 dengan membawakan game ‘Get the Trash’ dan mendapatkan medali emas kategori aplikasi permainan pada ajang Gemastik 10 di Universitas Indonesia dengan membawakan game ‘Plant the Future’.

*[Halaman ini sengaja dikosongkan]*

LAMPIRAN A

1. public  class  MapGenerator :  MonoBehaviour  {
2. public  int  sumHope;
3. int  chunt = 1;
4. bool  future  =  true;
5. bool  PlayerWay  =  true;
6. int  row  =  5;
7. int  column  =  5;
8. int  rowPoint;
9. int  columnPoint;
10. int  count;
11. bool  character = true;
12. int  crow;
13. int  ccol;
14. int  *obstacle*  =  0;
15. int  test  =  20;
16. int[, ]  map;
17. int[, ]  pastMap;
18. int[, ]  futureMap;
19. int[, ]  dummyMap;
20. int[]  randomDirection;
21. *obstacle*s[]  obs;
22. int[]  histRow;
23. int[]  histCol;
24. int  hisDir  =  0;
25. int  frow;
26. int  fcolumn;
27. int  through;
28. int  nowTurn;
29. int  direction  =  0;
30. /\* atas = 1     \* kanan = 2     \* kiri = 3     \* bawah = 4     \*/
32. bool  isHunt  =  false;
33. bool  isKill  =  false;
34. bool  first  =  true;
36. struct  *obstacle*s     {
37. public  int  x;
38. public  int  y;
39. }
40. void  HistReset()     {

LAMPIRAN B



Gambar B. Stage 2



Gambar B. Stage 3-1