TUGAS AKHIR – KI141502

RANCANG BANGUN JARINGAN SENSOR NIRKABEL BERBASIS ZIGBEE UNTUK KASUS MONITORING KUALITAS AIR PADA LINGKUNGAN TERDISTRIBUSI

BENITO DANNESWARA WIDYATAMA
NRP 05111540000097

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

Dosen Pembimbing II

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – KI141502

RANCANG BANGUN JARINGAN SENSOR NIRKABEL BERBASIS ZIGBEE UNTUK KASUS MONITORING KUALITAS AIR PADA LINGKUNGAN TERDISTRIBUSI

BENITO DANNESWARA WIDYATAMA
NRP 05111540000097

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

Dosen Pembimbing II

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

UNDERGRADUATE THESIS - KI141502

Prototype Design of Water Quality Monitoring System From Spread Wireless Sensor Networks Zigbee Based

BENITO DANNESWARA WIDYATAMA
NRP 05111540000097

First Advisor
Waskitho Wibisono, S.Kom., M.Kom., Ph.D

Second Advisor

Department of Informatics
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN JARINGAN SENSOR NIRKABEL BERBASIS ZIGBEE UNTUK KASUS MONITORING KUALITAS AIR PADA LINGKUNGAN TERDISTRIBUSI

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
BENITO DANNESWARA WIDYATAMA
NRP: 05111540000097

Disetujui oleh Pembimbing Tugas Akhir:

1. Waskitho Wibisono, S.Kom., M.Kom., Ph.D .....
   (NIP. 197410222000031001) (Pembimbing 1)

SURABAYA
Juni, 2020

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN JARINGAN SENSOR NIRKABEL BERBASIS ZIGBEE UNTUK KASUS MONITORING KUALITAS AIR PADA LINGKUNGAN TERDISTRIBUSI

Nama Mahasiswa	**:**	**BENITO DANNESWARA WIDYATAMA**
NRP	**:**	**05111540000097**
Jurusan	**:**	**Teknik Informatika FTIF-ITS**
Dosen Pembimbing 1	**:**	**Waskitho Wibisono, S.Kom., M.Eng., Ph.D**
Dosen Pembimbing 2	**:**	

Abstrak

.

Dalam beberapa tahun terakhir, Wireless Sensor Network (WSN) mendapatkan banyak perhatian dalam bidang aplikasi monitoring dan kontrol baik pada lingkungan hidup maupun industry. Sebagai salah satu jenis komputasi pervasive menggunakan jaringan sensor, WSN menyediakan beberapa keuntungan baik dalam fungsi maupun biaya.

Pada penelitian ini penulis akan membuat sebuah system untuk mengawasi dan mengevaluasi kualitas air secara real time, yang mana nantinya akan menganalisis data dari jaringan sensor nirkabel tersebar yang berbasis kepada teknologi Zigbee. Dengan adanya system ini, diharapkan dapat membantu pihak-pihak yang membutuhkan kebijakan untuk mengawasi kualitas air yang dimiliki, terkhususnya masyarakat yang memiliki usaha seperti tambak ikan.

Sistem monitoring kualitas air ini menggunakan mikrokontroler Arduino Uno. Kemudian untuk mendapatkan data kualitas air, system monitoring ini menggunakan beberapa sensor, yaitu sensor Gravity TDS Meter, dan sensor temperatur tahan air DS18B20. Adapun untuk berkirim data antar node, system ini

memanfaatkan modul XBee S2 Pro, dan mengirim data ke server menggunakan Python.

Sistem ini diuji coba mulai dari uji coba fungsionalitas hingga menggunakan beberapa scenario yang telah ditentukan. Berdasarkan hasil uji coba fungsionalitas, semua fungsi system berjalan dengan semestinya.

Kata kunci: Wireless Sensor Network, Kualitas Air, Mikrokontroler, Sensor,

PROTOTYPE DESIGN OF WATER QUALITY MONITORING SYSTEM FROM SPREAD WIRELESS SENSOR NETWORK ZIGBEE BASED

Student's Name	**: BENITO DANNESWARA WIDYATAMA**
Student's ID	**: 05111540000097**
Department	**: Teknik Informatika FTIF-ITS**
First Advisor	**: Waskitho Wibisono, S.Kom., M.Eng., Ph.D**
Second Advisor	**:**

Abstract

In the last few years, Wireless Sensor Network (WSN) has a lot of attention on topic of monitoring and control in topics regarding wildlife or industry. As one of a censor-based pervasive computing, WSN provides a lot of advantages both in cost as well as its function.

In this study, the author will create a system to monitor and evaluate water quality in real-time, which will then analyze the data from a network of wireless sensors spread based on Zigbee technology. With this system, it is expected to assist stakeholders in monitoring water quality that they have, especially the ones who have businesses that involves water, like fisheries.

This water quality monitoring system uses Arduino Uno microcontrollers. To get data of water quality, this system uses two sensors, that is, Gravity TDS Meter, and waterproof temperature censor DS18B20. And then, for sending data between nodes, this system using Xbee S2 Pro module, and to send data to server using a Python script.

The system is tested starting from functionality testing to performance test using some predefined scenarios. Based on the results of the functionality test, all system functions run properly.

Keywords: Wireless Sensor Network, Microcontroller, Water Quality, Sensor

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas segala rahmat, dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

"**RANCANG BANGUN SISTEM DETEKSI PERGERAKAN TANAH DENGAN ANALISIS DATA SILENT ZONE DARI JARINGAN SENSOR NIRKABEL TERSEBAR BERBASIS ZIGBEE**".

Tugas akhir ini merupakan salah satu syarat dalam menempuh ujian sidang guna memperoleh gelar Sarjana Komputer. Dengan pengerjaan tugas akhir ini, penulis bisa belajar banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS.

Selesainya tugas akhir ini tidak terlepas dari bantuan dan dukungan beberapa pihak, sehingga pada kesempatan ini penulis menghaturkan rasa syukur serta terima kasih kepada:

1. Allah SWT atas segala nikmat, rahmat, dan kesehata selama ini.
2. Bapak, ibu, dan keluarga penulis yang selalu memberikan dukungan, perhatian, dan kasih saying bagi penulis yang menjadi semangat selama perkuliahan maupun pengerjaan Tugas Akhir ini.
3. Bapak Waskitho Wibisono, S.Kom., M.Kom., Ph.D selaku dosen pembimbing penulis yang telah membimbing dan memberikan motivasi, nasehat dan bimbingan dalam menyelesaikan tugas akhir ini.
4. Bapak Arya Yudhi Wijaya, S.Kom., M.Kom., selaku dosen wali penulis yang telah memberikan arahan, masukan dan motivasi kepada penulis.
5. Bapak Dr. Eng Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknik Informatika ITS.

6. Bapak Dr. Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator tugas akhir.
7. Seluruh dosen dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di ITS.
8. Teman-teman IFLS yang telah menerima penulis dan mejadi keluarga penulis baik di kampus maupun di luar kampus.
9. Teman-teman seperjuangan RMK NCC/KBJ, yang telah menemani dan menyemangati penulis, terkhususnya Ubud, Burhan, Tommy, Hanif, dan lainnya.
10. Teman-teman lab NCC/KBJ, yang telah banyak membantu sekaligus menyemangati, dan senantiasa membersamai selama penulis menjadi administrator.
11. Teman-teman angkatan 2015, yang sudah mendukung penulis selama perkuliahan.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan sehingga dengan kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca untuk perbaikan ke depan.

Surabaya, Juni 2020

Benito Danneswara Widyatama

DAFTAR ISI

LEMBAR PENGESAHAN ..... **v**
Abstrak ..... **vii**
Abstract ..... **ix**
DAFTAR ISI ..... **xiii**
DAFTAR GAMBAR ..... **xvii**
DAFTAR TABEL ..... **xix**
DAFTAR PSEUDOCODE ..... **xx**
DAFTAR KODE SUMBER ..... **xxii**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Permasalahan 3
1.4 Tujuan 4
1.5 Manfaat 4
1.6 Metodologi 4
1.6.1 Penyusunan Proposal Tugas Akhir 4
1.6.2 Studi Literatur 5
1.6.3 Implementasi Perangkat Lunak 5
1.6.4 Pengujian dan Evaluasi 5
1.6.5 Penyusunan Buku 5
1.7 Sistematika Penulisan Laporan 6
BAB II TINJAUAN PUSTAKA **7**
2.1 Mikrokontroler *Arduino UNO* 7
2.2 *Wireless Sensor Network* (WSN) 8
2.3 Modul Sensor TDS Meter Analog 9
2.4 *Sensor Temperatur Digital DS18B20* 11
2.5 Xbee Shield 11
2.6 Xbee Modul S2 12
2.7 Python 13
2.8 *Arduino* IDE 14
2.9 Protokol *ZigBee* 15
2.10 *Framework CodeIgniter* 16
BAB III ANALISIS DAN PERANCANGAN **17**

3.1 Deskripsi Umum Sistem.....17
3.2 Arsitektur Umum Sistem.....18
3.3 Perancangan Antarmuka Sistem.....19
3.4 Perancangan Kebutuhan Arus Listrik.....21
3.5 Perancangan Perangkat Keras 22
3.5.1 Perancangan Perangkat Keras *ZigBee Coordinator*23
3.5.2 Perancangan Perangkat Keras *Zigbee Router*.....23
3.6 Perancangan Basis Data Sistem 24
3.7 Diagram Alir Aplikasi Sistem 25
3.7.1 Diagram Alir Data Keseluruhan Sistem 25
3.7.2 Diagram Alir Pembacaan Sensor TDS Meter dan Sensor Temperatur.....26
3.7.3 Diagram Alir Pengiriman Data 27
3.7.3.1 Diagram Alir Pengiriman Data dari ZigBee Router kepada ZigBee Coordinator.....28
3.7.3.2 Diagram Alir Pengiriman Data dari *ZigBee Coordinator* kepada *Web Service*.....29
3.7.4 Diagram Alir Menampilkan Data pada Website.....29
BAB IV IMPLEMENTASI.....31
4.1 Lingkungan Implementasi.....31
4.2 Implementasi Perangkat Keras.....32
4.2.1 Perangkat *Node ZigBee Coordinator*.....32
4.2.2 Perangkat *Node ZigBee Router*.....33
4.3 Implementasi Perangkat Lunak pada Perangkat Keras ..35
4.4 Implementasi Inisialisasi Sensor TDS Meter dan Temperatur 36
4.5 Implementasi Perangkat Lunak untuk Membaca Sensor Temperatur dan TDS Meter 37
4.6 Implementasi Perangkat Lunak untuk Mengirim Data ..38
4.6.1 Implementasi Menerima Data dari *Node Zigbee Router* oleh *Node Zigbee Coordinator* 39
4.6.2 Implementasi Mengirim Data dari *Node Zigbee Coordinator* kepada *Server*.....39
4.7 Implementasi Basis Data Sistem 40
BAB V HASIL UJI COBA DAN EVALUASI 42

5.1 Lingkungan Uji Coba 42
5.2 Uji Coba Fungsionalitas 43
5.2.1 Pengujian Komunikasi Node ZigBee Router dengan Node ZigBee Koordinator 44
5.2.2 Pengujian Pengiriman Data Kepada Server 45
5.3 Uji Coba Performa 47
5.3.1 Percobaan Pertama 48
5.3.2 Percobaan Kedua 49
5.3.3 Percobaan Ketiga 49
5.3.4 Percobaan Keempat 50
5.3.5 Percobaan Kelima 50
5.3.6 Percobaan Keenam 51
5.3.7 Percobaan Ketujuh 52
5.3.8 Percobaan Kedelapan 52
5.3.9 Percobaan Kesembilan 53
5.3.10 Percobaan Kesepuluh 54
5.4 Evaluasi Pengujian 54
5.4.1 Evaluasi Pengujian Fungsionalitas 54
5.4.2 Evaluasi Pengujian Performa 55
BAB VI KESIMPULAN DAN SARAN 59
7.1 Kesimpulan 59
7.2 Saran 59
DAFTAR PUSTAKA 61
LAMPIRAN A KODE SUMBER 62
BIODATA PENULIS 66

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 1.1 Figur TDS Meter 2
Gambar 2.1 *Arduino UNO*..... 7
Gambar 2.2 Contoh Skema dari WSN 9
Gambar 2.3 TDS Meter Analog 10
Gambar 2.4 Sensor Temperatur Digital DS18B20..... 11
Gambar 2.5 Logo *Python* 14
Gambar 2.6 *Arduino* IDE 15
Gambar 2.7 Logo *CodeIgniter* 16
Gambar 3.1 Sketsa sederhana WSN Pemantau Kualitas Air..... 17
Gambar 3.2 Arsitektur Dasar Sistem..... 18
Gambar 3.3 Arsitektur Detail Sistem 18
Gambar 3.4 Halaman Awal 20
Gambar 3.5 Halaman Live Data..... 20
Gambar 3.6 Halaman Tabel Data 21
Gambar 3.7 *Node ZigBee Coordinator* 23
Gambar 3.8 Node ZigBee Router dengan *Breadboard* 24
Gambar 3.9 Diagram alir data seluruh sistem 24
Gambar 3.10 Diagram alir pembacaan sensor TDS Meter dan Temperatur 27
Gambar 3.11 Diagram alir pengiriman data dari *ZigBee Router* kepada *ZigBee Coordinator* 28
Gambar 3.12 Diagram alir pengiriman data dari *ZigBee Coordinator* kepada *web service* 29
Gambar 3.13 Diagram alir menampilkan data pada *website* 30
Gambar 4.1 Perancangan *Node Zigbee Coordinator* 33
Gambar 4.2 Implementasi *Node Zigbee Coordinator* 33
Gambar 4.3 Perancangan *Node Zigbee Router*..... 34
Gambar 4.4 Implementasi *Node Zigbee Router* 34
Gambar 4.5 Perangkat Lunak X-CTU..... 35
Gambar 4.6 Percobaan Komunikasi Modul *Xbee* 36
Gambar 5.1 Hasil Uji Coba Fungsionalitas Komunikasi *ZigBee* 45
Gambar 5.2 Hasil Uji Coba Fungsionalitas Pengiriman Data kepada *Server* 47

Gambar 5.3 Hasil Uji Coba Fungsionalitas Pengiriman Data kepada *Server* 48

DAFTAR TABEL

Tabel 2.1 Arduino Data Sheet 8
Tabel 3.1 Kebutuhan Arus Listrik Node Zigbee Router 21
Tabel 3.2 Kebutuhan Arus Listrik Node Zigbee Coordinator 22
Tabel 3.3 Perancangan Tabel *Node01* dan *Node02* 25
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak 31
Tabel 5.1 Spesifikasi Lingkungan Uji Coba 43
Tabel 5.2 Skenario Uji Coba Fungsionalitas Komunikasi Node ZigBee Router dengan Node ZigBee Coordinator 44
Tabel 5.3 Skenario Uji Coba Fungsionalitas Pengiriman Data Kepada *Server* 46
Tabel 5.4 Skenario Uji Coba Performa Kondisi Air Normal 49
Tabel 5.5 Skenario Uji Coba Performa Kondisi Air Kotor 49
Tabel 5.6 Uji Coba Performa Pertama 49
Tabel 5.7 Uji Coba Performa Kedua 50
Tabel 5.8 Uji Coba Performa Ketiga 50
Tabel 5.9 Uji Coba Performa Keempat 50
Tabel 5.10 Uji Coba Performa Kelima 50
Tabel 5.11 Uji Coba Performa Keenam 51
Tabel 5.12 Uji Coba Performa Ketujuh 52
Tabel 5.13 Uji Coba Performa Kedelapan 52
Tabel 5.14 Uji Coba Performa Kesembilan 53
Tabel 5.15 Uji Coba Performa Kesepuluh 54
Tabel 5.16 Evaluasi Pengujian Fungsionalitas 55
Tabel 5.17 Evaluasi Pengujian Performa Kondisi Air Biasa 55
Tabel 5.18 Evaluasi Pengujian Performa Kondisi Air Kotor 55
Tabel 5.19 Rumusan Confusion Matrix 57
Tabel 5.20 Confusion Matrix Sistem Monitoring Kualitas Air ... 56

DAFTAR PSEUDOCODE

Pseudocode 4.1 *Pseudocode* Inisialisasi Sensor Orientasi 37
Pseudocode 4.2 *Pseudocode* Mendapatkan Nilai Sensor 38
Pseudocode 4.3 *Pseudocode* Penerimaan data *node ZigBee Router* oleh *node ZigBee Coordinator* 39
Pseudocode A.1 *Pseudocode* Inisialisasi Variabel Global *Node ZigBee Router* 63
Pseudocode A.2 *Pseudocode* Fungsi *Setup* untuk Mempersiapkan Sensor *Node ZigBee Router* 63
Pseudocode A.3 *Pseudocode* Fungsi untuk Pendeteksian Temperatur dan PPM *ZigBee Router* 64
Pseudocode A.4 *Pseudocode* Inisialisasi Variabel Global *Node ZigBee Coordinator* 64
Pseudocode A.5 *Pseudocode* Fungsi untuk Menerima Data *Node ZigBee Coordinator* 65
Pseudocode A.6 *Pseudocode* Inisialisasi *Python* Untuk Menerima Serial Data dari *Node Coordinator* 65
Pseudocode A.7 *Pseudocode* Fungsi untuk *Python* Memasukkan *Serial Data* ke *Server* 66

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 0.1 Inisialisasi Varibel Global Node ZigBee Router 62
Kode Sumber 0.2 Fungsi Setup untuk Mempersiapkan Sensor Node ZigBee Router.....62
Kode Sumber 0.3 Fungsi untuk Pendeteksian Temperatur dan PPM ZigBee Router 63
Kode Sumber 0.4 Inisialisasi Variabel Global Node ZigBee Koordinator 63
Kode Sumber 0.5 Inisialisasi Python Untuk Menerima Serial Data Dari Node Coordinator.....64
Kode Sumber 0.6 Fungsi untuk Python Memasukkan Serial Data ke Server.....65

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kualitas air yang menurun sedang menjadi isu global sangat diperhatikan dunia. Hal ini lama kelamaan muncul karena populasi manusia yang terus bertumbuh, berkembangnya industry dan agrikultur, dan perubahan iklim.[1] Secara keseluruhan, masalah mengenai kualitas air yang paling umum adalah eutrofikasi, dimana adanya kandungan nutrisi yang berlebih, terutama fosfor dan nitrogen dalam air, yang menyebabkan banyaknya tumbuhan dalam air dan banyak hewan yang mati akibat kekurangan oksigen.[2] Air yang terkena dampak eutrofikasi juga tidak dapat digunakan manusia dengan baik. Kualitas air yang buruk dapat berdampak pada kurangnya air yang bisa digunakan dalam banyak hal dalam suatu wilayah. Air yang terpolusi tidak bisa digunakan untuk minum, mandi, ataupun agrikultur.

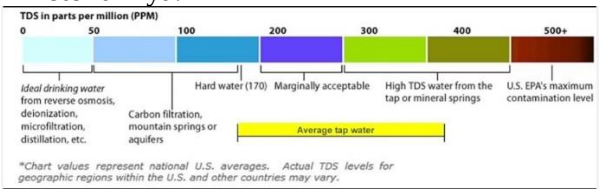
Dalam beberapa tahun terakhir, Wireless Sensor Network (WSN) mendapatkan banyak perhatian dalam bidang aplikasi monitoring dan control baik pada lingkungan hidup maupun industry.[3] Sebagai salah satu jenis komputasi pervasif menggunakan jaringan sensor, WSN menyediakan beberapa keuntungan baik dalam fungsi maupun biaya. Instalasi dan perawatan yang harus dilakukan pada peralatan yang digunakan juga murah dan mudah karena peralatan tidak menggunakan kabel. Karena itulah teknologi WSN dapat menarik perhatian baik dari para akademisi maupun industri.

Dengan teknologi WSN, sebuah kegiatan monitoring kualitas air dapat dilakukan. Teknologi ini dapat digunakan untuk mengawasi dan mengevaluasi kualitas air secara real time dengan mendeteksi data, mengirimkan data, dan menampilkan data.

Dalam pengerjaan tugas akhir ini, penulis akan menggunakan beberapa sensor yang terpasang dalam beberapa lokasi, yang saling berhubungan dengan memanfaatkan teknologi

Wireless Sensor Network (WSN). Sensor-sensor ini berfungsi merekam data kemudian mengirimkan data itu ke *server*.

Total Dissolved Solids, disebut juga TDS, dapat mengindikasikan berapa banyak milligram zat mudah larut yang terlarut dalam satu liter air. Secara singkatnya, semakin tinggi nilai TDS, semakin banyak zat mudah larut yang terlarut dalam air, dan semakin kotor airnya.



Gambar 1.1 Figur TDS Meter

Protokol yang digunakan pada implementasi WSN ini yaitu menggunakan protocol *ZigBee* (IEEE 802.15.4). Adapun perangkat keras yang digunakan untuk *Node ZigBee Coordinator* yaitu *Arduino Uno*, *XBee Pro S2* dan *XBee Shield*. Sedangkan untuk *Zigbee Router*, terdiri dari *Arduino Uno*, *XBee Pro S2*, *XBee Shield*, TDS Meter, dan sensor temperatur. Data yang telah diolah kemudian dikirim ke *web service*, sehingga kemudian dapat ditampilkan di *web server*.

Hasil yang diharapkan dari pengerjaan tugas akhir ini yaitu terciptanya sebuah *Wireless Sensor Network* guna melakukan kegiatan monitoring kondisi air pada danau. Sehingga diharapkan dari rancang bangun WSN ini dapat melakukan pemantauan kondisi air dalam berbagai titik di sebuah sumber air yang dapat dipantau melalui aplikasi web.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam pengerjaan tugas akhir ini adalah sebagai berikut :

a. Bagaimana metode yang digunakan untuk mendeteksi kualitas air secara WSN dengan beberapa node yang diletakkan disekitar lokasi?
b. Bagaimana cara sistem ini mendeteksi kualitas air yang dipantau?
c. Bagaimana metode mengekstrak data dari hasil rekaman sensor TDS Meter dan temperatur?
d. Bagaimana cara sistem ini memantau kualitas air melalui aplikasi *web server*?
e. Bagaimana implementasi protokol ZigBee untuk komunikasi antar node?

1.3 Batasan Permasalahan

Permasalahan yang dibahas di dalam pengerjaan tugas akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

a. Lingkungan uji coba berupa beberapa gelas berisi air yang diletakkan berjauhan
b. Data yang diolah dari sensor TDS Meter adalah hasil pembacaan PPM (*Particle Per Millimeter*)
c. Data yang diolah dari sensor temperatur adalah hasil pembacaan suhu dengan satuan derajat Celsius
d. Sistem akan terus mengirimkan data ke *server* selama sistem dinyalakan
e. Aplikasi server berupa aplikasi basis data menggunakan *MySQL* dengan memanfaatkan *Web Service*
f. Menggunakan mikrokontroller *Arduino*
g. Komunikasi nirkabel menggunakan protokol *ZigBee*

h. Aplikasi pemantauan kualitas air ini berbasis *web server*

1.4 Tujuan

Tujuan pembuatan tugas akhir ini adalah :

a. Mengimplementasikan sebuah *Wireless Sensor Network* (WSN) yang dapat memantau pergerakan kualitas air pada sumber air atau sistem hidroponik
b. Mengirimkan data monitoring secara *real time* kedalam sebuah aplikasi berbasis web

1.5 Manfaat

Manfaat dari pembuatan tugas akhir adalah sebagai system yang dapat melakukan pengawasan kualitas air pada sebuah sumber air atau sistem hidroponik yang luas dari jarak jauh, sehingga dapat membantu banyak pihak, baik kepada masyarakat umum, maupun pihak yang berkepentingan di pemerintahan.

1.6 Metodologi

Adapun langkah-langkah yang ditempuh dalam pengerjaan penelitian ini adalah sebagai berikut :

1.6.1 Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini 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 juga dijabarkan tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan

mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

1.6.2 Studi Literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan rancang bangun WSN pengawasan kualitas air yaitu mengenai pemrograman *Arduino*, *web service*, dan protokol *ZigBee*.

1.6.3 Implementasi Perangkat Lunak

Pembangunan aplikasi akan dilakukan dengan menggunakan Bahasa pemrograman SQL, PHP, dan C/C++. Sedangkan untuk IDE (Integrated Development Environment) yang digunakan yaitu Arduino IDE,dan Microsoft Visual Studio. Framework back end yang digunakan untuk *Web Server* yaitu Code Igniter, dengan menggunakan MySQL sebagai RDBMS.

1.6.4 Pengujian dan Evaluasi

Pengujian sistem pengawasan kualitas air dilakukan dengan menguji coba fungsionalitas dan performa sistem WSN, yang mana nantinya akan dijalankan skenario 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.

1.6.5 Penyusunan Buku

Pada tahap ini disusun buku sebagai dokumentasi dari pelaksanaan tugas akhir yang mencangkup seluruh konsep, teori, implementasi, serta hasil yang telah dikerjakan.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan tugas akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
   Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan tugas akhir.
2. Bab II. Tinjauan Pustaka
   Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan tugas akhir ini. Secara garis besar, bab ini berisi tentang mikrokontroler, sensor yang digunakan, protokol *ZigBee*, dan lainnya
3. Bab III. Analisis dan Perancangan
   Bab ini berisi pembahasan mengenai perancangan dari sistem deteksi pergerakan tanah berbahaya.
4. Bab IV. Implementasi
   Bab ini menjelaskan implementasi yang berbentuk *pseudocode* yang berupa *pseudocode*.
5. Bab V. Pengujian dan Evaluasi
   Bab ini berisikan hasil uji coba fungsionalitas dan uji coba performa. Hasil evaluasi mencakup evaluasi uji coba fungsionalitas dan evaluasi uji coba performa
6. Bab VI. Kesimpulan dan Saran
   Bab ini merupakan bab yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan, masalah-masalah yang dialami pada proses pengerjaan tugas akhir, dan saran untuk pengembangan solusi ke depannya.
7. Daftar Pustaka
   Bab ini berisi daftar pustaka yang dijadikan literatur dalam tugas akhir.
8. Lampiran
   Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode program secara keseluruhan.

BAB II
TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam tugas akhir. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Mikrokontroler *Arduino UNO*

Arduino Uno adalah salah satu produk berlabel *Arduino* yang sebenarnya adalah suatu papan elektronik yang mengandung mikrokontroler ATmega 328, sebuah *keeping* yang secara fungsional bertindak seperti sebuah komputer. Piranti ini dapat dimanfaatkan untuk mewujudkan rangkaian elektronik dari yang sederhana hingga yang kompleks [4]. Dengan penambahan komponen tertentu, piranti ini bisa digunakan untuk pemantauan jarak jauh melalui internet. *Arduino* yang digunakan penulis dalam penelitian ini adalah jenis *Arduino Uno* berbasis ATmega 328, sebagaimana yang ditunjukkan pada Gambar 2.1

Gambar 2.1 *Arduino UNO*

Tabel 2.1 Arduino Data Sheet

Mikrokontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7 – 12 V
Input Voltage (limits)	6 – 20 V
Digital I/O Pins	14 (6 pin merupakan *pulse width modulation output*)
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (Atmega328) dimana 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

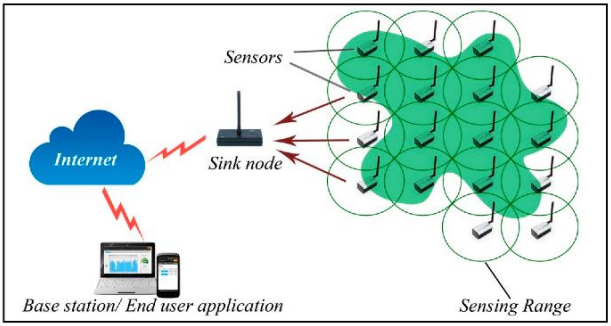
Mikrokontroler ATmega328 memiliki 14 pin digital input/output (6 pin digunakan sebagai PWM output, 6 analog input, 16 MHz sumber *clock signal* untuk *digital circuits*, sebuah konektor USB, sebuah *socket* listrik, sebuah konektor ICSP, dan tombol reset. *Arduino Uno* dapat diberikan *power* melalui konektor USB atau dengan *power supply* eksternal. Pada pengerjaan tugas akhir ini, *Arduino Uno* yang digunakan digunakan sebagai mikrokontroller untuk mengatur sensor TDS meter dan sensor temperatur DS18B20 agar dapat beroperasi dengan baik.

2.2 *Wireless Sensor Network* (WSN)

Wireless Sensor Network (WSN) adalah sekumpulan sensor yang saling bekerja sama mengawasi sebuah lingkungan yang luas. *Node* sensor tidak hanya berkomunikasi satu sama lain, namun mereka juga berkomunikasi dengan sebuah *Base Station* (BS) secara nirkabel, membuat mereka dapat menyebar data yang terekam untuk diproses lebih lanjut.

Kemampuan masing-masing *node* sensor dalam WSN dapat bervariasi, seperti misalnya, *node* sensor simple mungkin hanya dapat merekam satu jenis kejadian, sedangkan peralatan yang lebih kompleks dapat merekam lebih banyak kejadian dengan menggabungkan beberapa sensor. *Node* juga dapat memiliki kapabilitas komunikasi yang berbeda, misal, menggunakan *ultrasound*, *infrared*, atau frekuensi radio.[3]

Pada penelitian ini penulis akan menggunakan teknologi *Wireless Sensor Network* (WSN). Node yang digunakan sebanyak 3 buah, dan menggunakan beberapa komponen yaitu *Arduino*, dan beberapa sensor untuk mendeteksi temperatur, dan kualitas *part per milimeter* (PPM) dari air.



Gambar 2.2 Contoh Skema dari WSN

2.3 Modul Sensor TDS Meter Analog

Sebuah pen *Total Dissolved Solids* (TDS) adalah sebuah peralatan yang digunakan untuk mengukur TDS yang dimiliki

objek yang diukur. TDS sendiri merupakan indicator dari berapa banyak milligram zat mudah larut yang terkadung dalam air, yang menunjukkan berapa bersih air tersebut. Peralatan ini merupakan seperangkat alat sensor yang dapat digunakan menggunakan mikrokontroler *Arduino*.

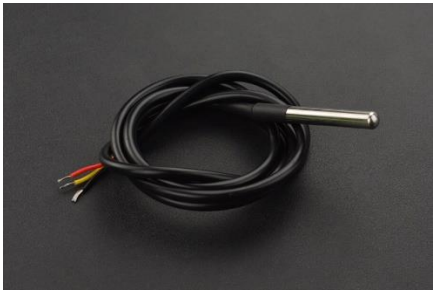
Gambar 2.3 TDS Meter Analog

TDS Meter Analog ini menyediakan *input* dari 3.3 – 5.5 volt, dan *output* 0 – 2.3 volt, yang membuatnya dapat digunakan dengan sistem yang mengoperasikan 5 volt atau 3.3 volt. *Probe* dari TDS Meter Analog ini tahan air, jadi dapat digunakan untuk operasi pembacaan sensor dalam air untuk jangka waktu yang lama.

TDS Meter Analog dapat digunakan dengan baik untuk kegiatan pembacaan kualitas air, yang sangat berguna untuk analisa air di mata air atau pengembangan lading hidroponik. Pada penelitian ini, TDS Meter Analog akan digunakan penulis untuk melakukan pembacaan pada TDS dari sumber air dan EC yang dimiliki.

2.4 *Sensor Temperatur Digital DS18B20*

Sensor Temperatur Digital DS18B20 merupakan sensor dengan kemampuan mendeteksi temperatur yang diproduksi oleh DFRobot. Sensor ini merupakan versi tahan air dari sensor temperatur yang diproduksi oleh DFRobot, yang didesain khusus untuk aktifitas pembacaan sensor yang melibatkan air. Sensor DS18B20 ini menyediakan pembacaan temperatur dalam antarmuka 1 kabel dari 9 hingga 12-bit, jadi sensor ini hanya memerlukan satu kabel untuk dapat tersambung dengan mikroprosesor. Sensor ini dapat digunakan untuk sistem yang menggunakan 3.0-5.5V. Sensor ini juga dapat digunakan untuk meletakkan banyak sensor di banyak tempat.



Gambar 2.4 Sensor Temperatur Digital DS18B20

Dalam tugas akhir ini sensor temperatur digital DS18B20 ini digunakan untuk mendeteksi besarnya suhu yang dimiliki sumber air.

2.5 Xbee Shield

XBee *Shield* merupakan modul tambahan yang dipergunakan untuk menghubungkan antara *board Arduino* dengan XBee S2 yang digunakan. XBee *Shield* ini memungkinkan *board*

Arduino untuk berkomunikasi secara wireless menggunakan XBee S2 dengan dilengkapi LED indikasi RSSI.

Xbee Shield yang digunakan penulis yaitu Xbee Shield keluaran dari perusahaan funduino, yang mana *jumper* yang digunakan berupa tipe *switch*, sehingga user bisa dengan mudah memilih antara mode USB maupun mode Xbee. Sangat dianjurkan bagi user untuk terlebih dahulu membaca dengan seksama *Data Sheet* dari masing-masing Xbee Shield tersebut.

Gambar 2.5 XBee Shield funduino

2.6 Xbee Modul S2

Modul Xbee S2 merupakan perangkat yang berfungsi sebagai protokol komunikasi nirkabel. Perangkat *wireless* yang membutuhkan daya rendah ini cocok untuk implementasi pada jaringan sensor nirkabel karena harganya yang relatif murah. Modul yang beroperasi pada frekuensi *band* ISM (Industrial, Scientific & Medical) 2,4 GHz ini menyajikan kemampuan untuk pengiriman data antar perangkat dengan kemampuan kisaran jarak

yang bervariasi tergantung pada kondisi dan tempat (*indoor* & *outdoor*). Spesifikasi dari modul ini ditunjukkan pada Tabel 1. Protokol Xbee Pro ini disiapkan untuk aplikasi yang memerlukan kecepatan data dan konsumsi daya rendah.

Gambar 2.6 Xbee Modul S2 Pro

Perangkat ini memiliki 20 pin dengan fungsi yang berbeda-beda. Untuk koneksi minimum, dibutuhkan pin VCC, GND, DOUT & DIN. Sedangkan untuk dapat melakukan *update firmware*, dibutuhkan koneksi pin VCC, GND, DIN, DOUT, RTS & DTR. VCC dan GND untuk tegangan suplai, DOUT merupakan pin Transmit (TX), DIN merupakan pin *Receive* (RX), RESET merupakan pin *reset* XBee PRO dan yang terakhir adalah PWMO/RSSI merupakan indikator bahwa ada penerimaan data yang biasanya dihubungkan ke LED yang di-*drive* oleh transistor. Untuk mengaktifkan XBee dibutuhkan suplai tegangan sebesar 3.3 V. XBee merupakan modul *tranceiver*, dengan mekanisme pengiriman data secara *serial*. Pin-pin untuk mengirim dan menerima data ada di pin 2 sebagai pin data OUT (Tx) dan pin 3 sebagai pin data IN (Rx). sehingga penting untuk mengetahui pin Tx dan Rx.

2.7 Python

Python adalah bahasa pemrograman interpretatif multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. *Python* memiliki struktur data *built-in level* tinggi dengan *dynamic typing* dan *dynamic binding*, menjadikan *Python* sangat menarik untuk pengembangan aplikasi yang cepat dan dapat digunakan untuk *scripting* atau menggabungkan beberapa komponen menjadi satu. *Python* diklaim sebagai bahasa yang menggabungkan kapabilitas dan kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. *Python* umumnya digunakan sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai sistem operasi.

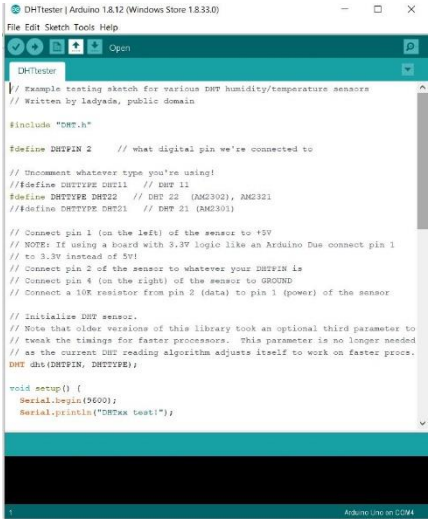
Pada tugas akhir ini, penulis memanfaatkan *Python* untuk kepentingan membaca serial data yang dikeluarkan *Node Coordinator* untuk kemudian dikirimkan ke *web server*.

Gambar 2.5 Logo *Python*

2.8 *Arduino* IDE

Lingkungan pengembangan *Arduino* terdiri dari *text editor* untuk menulis kode *Arduino*, area pesan, *console text*, *toolbar* dengan tombol untuk fungsi secara umum digunakan pada kode

Arduino, dan beberapa menu. *Arduino* IDE yang ditunjukkan pada Gambar 1.3 di bawah ini akan terkoneksi dengan perangkat keras *Arduino* ketika *Arduino* disambungkan dengan kabel USB untuk mengunggah program dan saling berkomunikasi. Program untuk *Arduino* ditulis pada *sketch*. *Sketch* yang disimpan dalam *format file* .ino.



Gambar 2.6 *Arduino* IDE

2.9 Protokol *ZigBee*

ZigBee merupakan sebuah standar komunikasi *wireless* yang dikembangkan dari standar IEEE 802.15.4 [8]. *ZigBee* dikembangkan oleh aliansi perusahaan yang disebut dengan *ZigBee Alliance*. Lebih dari 300 perusahaan yang tergabung melakukan pengembangan dari standar komunikasi *ZigBee*. *ZigBee* memiliki beberapa keunggulan dan kelemahan sebagai standar komukasi *wireless*. Keunggulan dari *ZigBee* adalah harga yang murah dan rendahnya konsumsi energi yang dibutuhkannya. Keuntungan ini yang membuat *ZigBee Alliance* selalu mencoba mengembangkan standar komunikasi ini. Akan tetapi *ZigBee*

memiliki kelemahan dalam kecepatan *transfer* data. Kecepatan *transfer* data maksimal pada standar komunikasi *ZigBee* adalah 250Kpbs, oleh karena itu *ZigBee* jarang sekali digunakan untuk melakukan *transfer data* dalam jumlah yang besar.

2.10 *Framework CodeIgniter*

CodeIgniter adalah sebuah *framework* PHP yang dibangun dengan konsep MVC (*Model-View-Controller*) berbasis *open-source*. *CodeIgniter* adalah kerangka kerja pengembangan situs web berbasis MVC yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan juga untuk meningkatkan pengalaman bekerja menggunakan aplikasi dengan menyediakan sintaks yang ekspresif, jelas, dan menghemat waktu. MVC sendiri adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, *controller*, dan *user interface*.

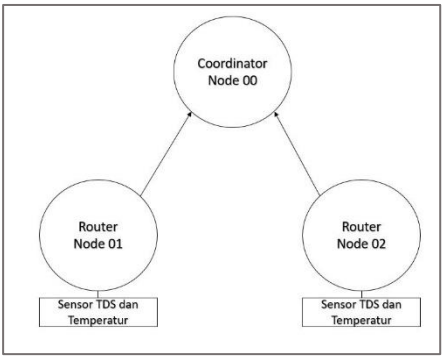
Gambar 2.7 Logo *CodeIgniter*

BAB III
ANALISIS DAN PERANCANGAN

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

3.1 Deskripsi Umum Sistem

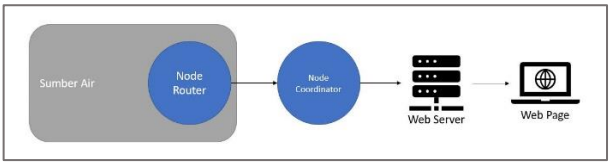
Pada tugas akhir ini akan dibangun suatu sistem pemantau kualitas air pada beberapa sumber air, yang mana akan menggunakan teknologi *Wireless Sensor Network*, dibangun menggunakan mikrokontroller *Arduino*, dan berstandar protokol komunikasi *ZigBee*. Aplikasi pemantau kualitas air ini dirancang untuk dapat melakukan pemantauan jarak jauh, kemudian data yang didapatkan oleh Node Router dikirim ke *server database.* Sensor Analog TDS Meter mampu mendeteksi *Electrical Conductivity* (EC) dari air yang kemudian akan diolah menjadi *Total Dissolved Solids* (TDS), dan sensor temperatur DS18B20 yang mampu mendeteksi temperatur air. Kemudian data hasil pembacaan sensor tersebut akan dikirimkan ke *web server* untuk disimpan dan ditampilkan.



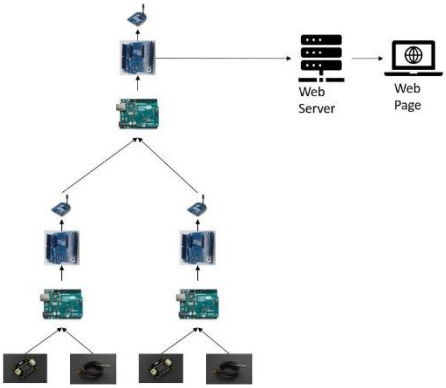
Gambar 3.1 Sketsa sederhana WSN Pemantau Kualitas Air

3.2 Arsitektur Umum Sistem

Dalam pengembangan teknologi *Wireless Sensor Network* pada tugas akhir ini, perangkat *XBee S2 Pro* akan berperan sebagai perangkat yang membantu komunikasi antar *node* dengan standar komunikasi *ZigBee*. Dan nantinya dibutuhkan dua jenis perangkat *ZigBee*, yaitu *node ZigBee Coordinator* dan *node ZigBee Router*. Perangkat *node ZigBee Router* berfungsi untuk membaca kondisi *Particle Per Milimeter* (PPM) dan suhu air, dimana akan dilakukan pengecekan secara simultan. Selanjutnya, data yang telah direkam oleh perangkat *ZigBee Router* akan dikirimkan ke perangkat *node ZigBee Coordinator*. Pada perangkat *ZigBee Coordinator*, data yang diterima akan dikirimkan kepada *server*.



Gambar 3.2 Arsitektur Dasar Sistem



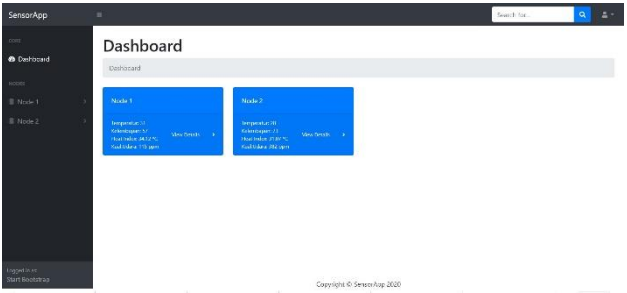
Gambar 3.3 Arsitektur Detail Sistem

Berdasarkan pada Gambar 3.3 sistem pemantauan kualitas air ini memiliki alur proses yang dijabarkan sebagai berikut :

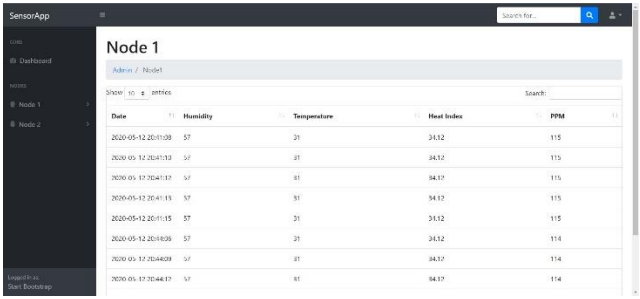
1. Pada *ZigBee Coordinator*, *XBee Shield* dipasang diatas mikrokontroler *Arduino Uno*, kemudian diatas *XBee Shield* tersebut modul *XBee S2* dipasang.
2. Pada *ZigBee Router*, *XBee Shield* dipasang diatas mikrokontroler *Arduino Uno*, kemudian diatas *XBee Shield* tersebut, modul *XBee S2* dipasang.
3. Pada *ZigBee Router*, sensor TDS Meter dan sensor temperatur DS018B20 dipasang diatas *XBee Shield*.
4. Pada *ZigBee Router*, rangkaian mikrokontroler diletakkan disamping sumber air, dengan kedua sensor dicelupkan kedalam air untuk melakukan pembacaan data.
5. Pada *ZigBee Coordinator*, disambungkan ke komputer induk agar dapat melakukan pengiriman data yang diterima kedalam *server*.
6. Pada tahap ini, sistem pemantauan kualitas air sudah siap dioperasikan guna mendapatkan data ppm dan temperatur.
7. Hasil pembacaan dari semua *node ZigBee Router* akan dikirim kepada *ZigBee Coordinator* melalui jaringan *ZigBee*.
8. Data yang diterima oleh *ZigBee Coordinator* selanjutnya akan dikirimkan ke *server* melalui *cursor* dan *dbConn*. Pengiriman data dilakukan dengan cara menjalankan *script* Python menggunakan IDLE dari komputer. *Script* itu bertugas membaca data *serial* yang dikeluarkan *node ZigBee Coordinator*.
9. Situs pemantauan Sistem Monitoring Kualitas Air akan melakukan *request* pula berupa HTTP GET pada *server* untuk mendapatkan data, sehingga dapat ditampilkan.

3.3 Perancangan Antarmuka Sistem

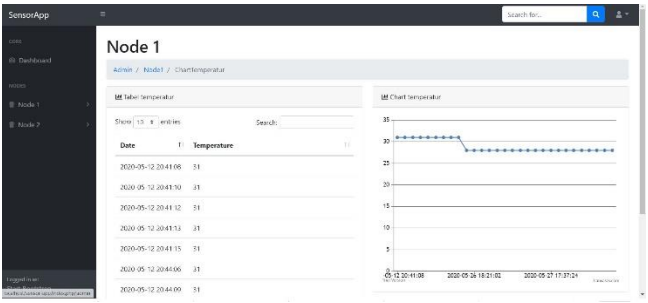
Pada penelitian ini akan dirancang antarmuka untuk pengguna, yang mana terdiri dari antar muka situs web. Pada antarmuka situs web, penulis menggunakan kerangka kerja *backend CodeIgniter*, sehingga sangat membantu dalam menyediakan *web service* yang dibutuhkan. Bahasa pemrograman yang penulis gunakan yaitu PHP 5.4.



Gambar 3.4 Halaman Awal



Gambar 3.5 Halaman Live Data



Gambar 3.6 Halaman Tabel Data

3.4 Perancangan Kebutuhan Arus Listrik

Perancangan kebutuhan arus listrik adalah salah satu proses penting yang harus dilakukan dalam membangun sistem. Hal ini terjadi karena masing-masing komponen dalam suatu sistem membutuhkan catu daya yang berbeda-beda, agar seluruh komponen bisa mendapatkan arus listrik minimum yang baik dan dapat secara optimal menjalankan sistem, maka perlu dipersiapkan kebutuhan arus listrik masing-masing komponen. Perancangan kebutuhan arus listrik dapat dilihat pada tabel dibawah ini

Tabel 3.1 Kebutuhan Arus Listrik *Node Zigbee Router*

Nama Perangkat	**Kebutuhan Arus Listrik**	
	Arus Listrik	**Tegangan Listrik**
Arduino Uno	40 mA	5V DC
Sensor TDS Meter Analog dan DS08B20	40 mA	5V DC
Xbee Shield	50 mA	5V DC
Xbee S2 Pro	40 mA	5V DC
Total	**130 mA**	**-**

Tabel 3.2 Kebutuhan Arus Listrik *Node Zigbee Coordinator*

Nama Perangkat	Kebutuhan Arus Listrik	
	Arus Listrik	Tegangan Listrik
Arduino Uno	50 mA	5V DC
Xbee Shield	50 mA	5V DC
Xbee S2 Pro	40 mA	5V DC
Total	**140 mA**	**-**

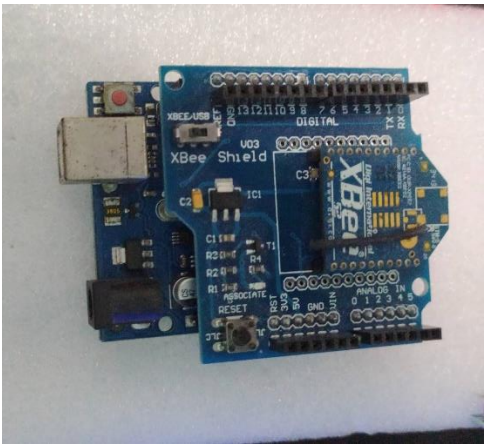
3.5 Perancangan Perangkat Keras

Perancangan perangkat keras secara umum menjelaskan mengenai perancangan penempatan perangkat keras yang digunakan untuk membangun sistem, yang mana terdiri dari rangkaian *ZigBee Coordinator* dan *ZigBee Router*. Rangkaian perangkat keras pada sistem ini seperti pada Gambar 3.7 dan Gambar 3.8, terdapat komponen yang digunakan agar sistem dapat berjalan sebagai berikut :

1. Dua sensor TDS Meter Analog
2. Dua sensor temperatur DS18B20
3. Tiga *Arduino Uno*
4. Dua *Power Bank*
5. Tiga *Xbee Shield*
6. Tiga *Xbee S2 Pro*
7. Dua *Breadboard*

3.5.1 Perancangan Perangkat Keras *ZigBee Coordinator*

Untuk rangkaian *ZigBee Coordinator*, *XBee Shield* dipasang diatas *Arduino Uno*, yang mana pin Tx dan Rx pada shield tersebut harus menempati pin Tx dan Rx *Arduino Uno*. Kemudian modul *XBee S2 Pro* diposisikan pada tempat yang telah disediakan pada *XBee Shield*.



Gambar 3.7 ***Node ZigBee Coordinator***

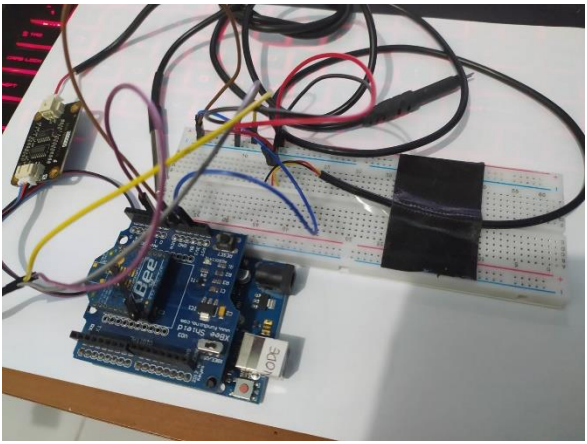
3.5.2 Perancangan Perangkat Keras *Zigbee Router*

Untuk rangkaian *ZigBee Router*, *XBee Shield* dipasangkan tepat diatas *Arduino Uno*, yang mana pin Tx dan Rx pada *XBee Shield* menempati posisi yang sama dengan pin Tx dan Rx *Arduino Uno*. Modul *XBee S2 Pro* diposisikan pada tempat yang telah disediakan pada *XBee Shield*. Kemudian, sensor TDS Meter dan DS08B20 dipasang pada XBee melalui pin yang terdapat pada *XBee Shield*.

Untuk rincian pemasangan sensor pada mikrokontroler, awalnya *probe* dari TDS Meter dipasangkan pada modul TDS Meter. Kemudian baik dari pin milik modul TDS dan sensor

temperatur dipasangkan pada *breadboard*, untuk nantinya disambungkan ke *Arduino Uno*. Disini pin **VCC** dari kedua sensor dijadikan satu pada *breadboard*, begitu juga dengan pin **GND**. Kemudian pin antara *breadboard* dengan pin yang terdapat pada *XBee Shield* dihubungkan dengan rincian:

- Pin **A** pada TDS Meter dihubungkan dengan pin **analog 1**
- Pin **DATA** pada DS08B20 dihubungkan dengan pin **digital 7**
- Pin **VCC** dihubungkan dengan pin tegangan **5V**
- Pin **GND** dihubungkan dengan pin **GND**



Gambar 3.8 Node ZigBee Router dengan *Breadboard*

3.6 Perancangan Basis Data Sistem

Basis data yang digunakan pada aplikasi situs web Sistem *Monitoring* Kualitas Air ini adalah *MySQL*. Pada sistem ini akan diperlukan dua table yaitu *node01* dan *node02*, yang digunakan untuk menyimpan data hasil *monitoring* kedua *node* yang akan

ditampilkan pada halaman *live monitoring*. Perancangan tabel dapat dilihat pada Table 3.3

Tabel 3.3 Perancangan Tabel *Node01* dan *Node02*

No.	Nama Atribut	Tipe Data	Keterangan
1	id	int	Primary key untuk kedua tabel.
2	date	timestamp	Waktu record data diterima
3	ppm	float	Nilai dari PPM yang diterima dari pembacaan sensor TDS
4	ec	float	Nilai dari EC yang diterima dari pembacaan sensor TDS
5	temperature	float	Nilai dari temperatur yang diterima dari pembacaan sensor suhu

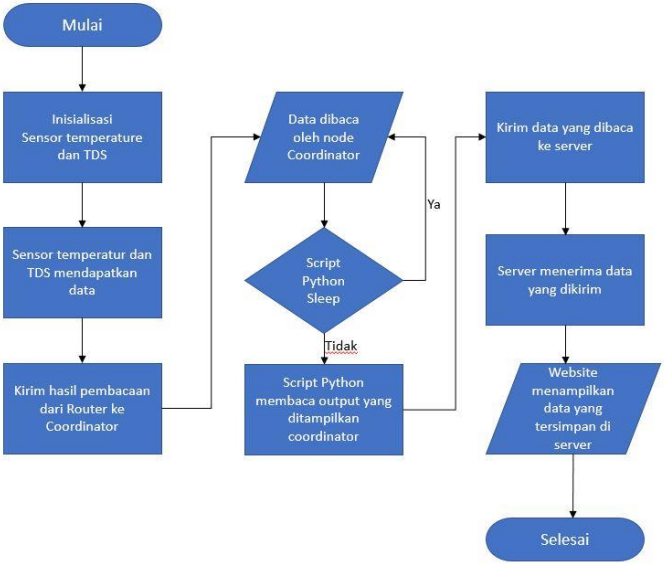
3.7 Diagram Alir Aplikasi Sistem

Diagram alir aplikasi sistem dibuat untuk memudahkan dalam merancang dan memahami seluruh proses yang terjadi di dalam sistem. Diagram alir aplikasi sistem terdiri dari diagram alir keseluruhan sistem, diagram alir pembacaan sensor TDS Meter dan sensor temperatur, diagram alir pengiriman data, dan diagram alir menampilkan data di website

3.7.1 Diagram Alir Data Keseluruhan Sistem

Perancangan alir data keseluruhan sistem dilakukan untuk dapat lebih memahami dan memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, serta menunjukkan tentang fungsi-fungsi utama atau proses yang ada.

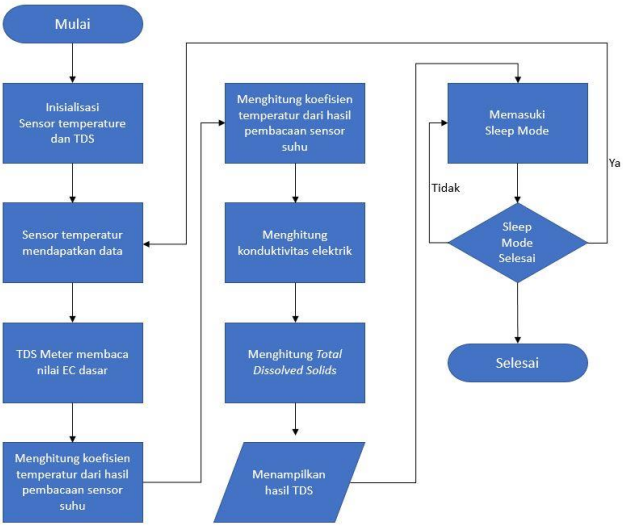
Pada Gambar 3.9, sistem diawali dengan *node Router* menginisialisasi sensor temperatur dan TDS Meter, kemudian sensor temperatur mulai merekam suhu air untuk perhitungan TDS dengan TDS Meter. Kemudian hasil pembacaan sensor akan dikirim kepada *Node Coordinator*. Disini data yang dikirim ke *node Coordinator* akan dibaca oleh *node* itu. Jika *script python* sedang tidak dalam kondisi *Sleep*, *script* itu akan membaca *output* yang dikeluarkan *node Coordinator*, untuk kemudian dikirimkan ke *server*. Data sensor yang telah dikirimkan ke *server* tersebut kemudian akan ditampilkan di situs web.



Gambar 3.9 Diagram alir data seluruh sistem

3.7.2 Diagram Alir Pembacaan Sensor TDS Meter dan Sensor Temperatur

Pada **Error! Reference source not found.** dalam proses pembacaan sensor TDS dan suhu, terlebih dahulu sensor temperatur mendapatkan nilai temperatur air, yang mana kemudian nilai tersebut dimasukkan dalam perhitungan koefisien temperatur. Nilai koefisien temperatur kemudian dimasukkan untuk perhitungan konduktivitas elektrik dengan nilai dasar yang belum terkalibrasi yang sebelumnya dibaca oleh TDS Meter. Kemudian, dari nilai konduktivitas elektrik yang didapatkan, nilai *Total Dissolved Solids* dapat didapatkan. Pada akhirnya, *ZigBee Router* akan menerima data sensor berupa nilai TDS, konduktivitas elektrik (EC), dan temperatur air. Sensor kemudian akan memasuki masa *sleep mode* selama beberapa waktu, untuk kemudian melakukan pembacaan lagi setelah masa *sleep mode* terlewati.



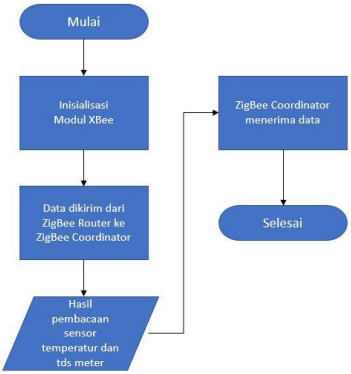
Gambar 3.9 Diagram alir pembacaan sensor TDS Meter dan Temperatur

3.7.3 Diagram Alir Pengiriman Data

Pada proses pengiriman data, Sistem Monitoring Kualitas Air ini memiliki dua diagram alir. Adapun ketiga diagram tersebut yaitu Diagram alir pengiriman data dari zigbee Router kepada zigbee Koordinator, dan Diagram alir pengiriman data dari zigbee Koordinator kepada Web Service.

3.7.3.1 Diagram Alir Pengiriman Data dari ZigBee Router kepada ZigBee Coordinator

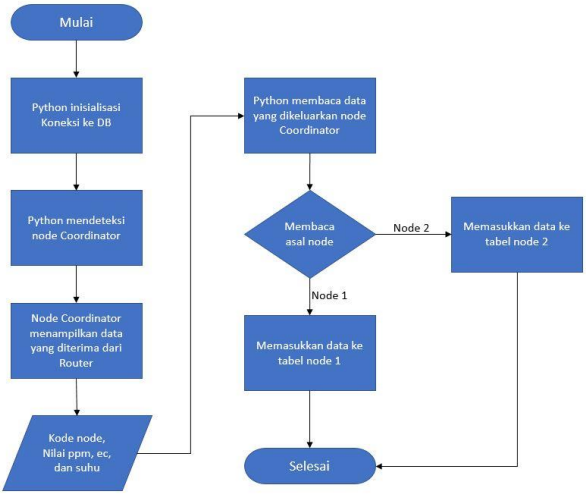
Pada Gambar 3.10 sebagai proses pengiriman data dari *ZigBee Router* kepada *ZigBee Coordinator*, terlebih dahulu dilakukan inisialisasi untuk semua modul *XBee*. Ada beberapa metode / cara atas inisialisasi modul *XBee*. Adapun yang penulis gunakan dalam penelitian ini yaitu dengan menggunakan aplikasi X-CTU. Setelah itu, kedua *ZigBee Router* mengirim data sensor yang sudah diproses secara simultan kepada *ZigBee Coordinator* melalui komunikasi ZigBee, yang mana dikirim dengan memanfaatkan *serial port*.



Gambar 3.10 Diagram alir pengiriman data dari *ZigBee Router* kepada *ZigBee Coordinator*

3.7.3.2 Diagram Alir Pengiriman Data dari *ZigBee Coordinator* kepada *Web Service*

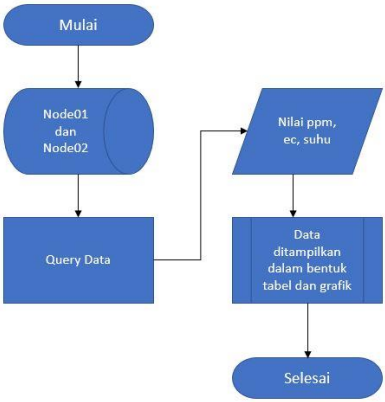
Pada Gambar 3.11 menunjukkan proses pengiriman data dari *ZigBee Coordinator* kepada *database server* dalam *web service* melalui sebuah *script python*. Setelah dilakukan inisialisasi koneksi ke *database* dan mendeteksi *node coordinator*, maka tahap yang akan dilakukan oleh *script python* selanjutnya adalah membaca *serial* yang dikeluarkan oleh *Arduino*. Dalam *serial* yang ditampilkan *node ZigBee Coordinator*, terdapat kode yang menunjukkan dari *node* mana pembacaan itu berasal. Berdasarkan kode itu *script* ini akan memasukkan data sesuai tabelnya masing-masing.



Gambar 3.11 Diagram alir pengiriman data dari ZigBee Koordinator kepada web service

3.7.4 Diagram Alir Menampilkan Data pada Website

Pada Gambar 3.12 merupakan proses aliran data dalam menjalankan situs web aplikasi *monitoring* kualitas air, yang mana sistem melakukan request HTTP GET kepada *server*. Data disimpan ke dalam *database* dengan kondisi yang diberikan pada *server* yang menerima data dari *ZigBee Coordinator*. Pada bagian *backend website*, penulis menggunakan *framework backend* CodeIgniter dan *database* MySQL.



Gambar 3.12 Diagram alir menampilkan data pada *website*

BAB IV
IMPLEMENTASI

Bab ini berisi penjelasan mengenai implementasi dari perancangan yang sudah dilakukan pada bab sebelumnya. Implementasi berupa *pseudocode* untuk membangun program. Cakupan implementasi dari perancangan sistem meliputi perangkat *node ZigBee Router* dalam membaca data PPM dan temperatur air, perangkat *node ZigBee Coordinator* dalam menerima data dari *ZigBee Router*, sekaligus proses pengiriman data, kemudian pengimplementasian antarmuka. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Arduino*, dan PHP

4.1 Lingkungan Implementasi

Lingkungan implementasi perangkat keras dan perangkat lunak akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Lingkungan Implementasi Perangkat Lunak

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel Core i7-7700HQ CPU @ 2.80GHz (8 CPUs)
	Memori	8 GB 2400 MHz DDR4
	Perangkat Sistem	Arduino UNO, TDS Meter sensor, DS08B20 waterproof temperature sensor, Xbee Shield, Xbee Module.
Perangkat Lunak	Sistem Operasi	Windows 10
	Perangkat Pengembang	Arduino Ide 1.8.1 IDLE Microsoft Visual Studio Code

4.2 Implementasi Perangkat Keras

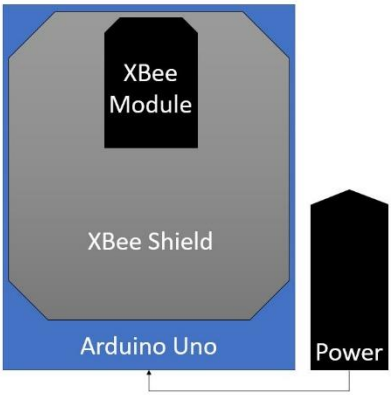
Implementasi perangkat keras sistem pengawasan kualitas air adalah sebagai berikut:

- 3 (tiga) buah *Arduino Uno*.
- 2 (dua) buah Sensor *TDS Meter*.
- 2 (satu) buah Sensor temperatur DS18B20.
- 3 (tiga) buah *XBee Shield*
- 3 (tiga) buah *XBee S2 Pro*
- 2 (dua) buah *Power Bank*.

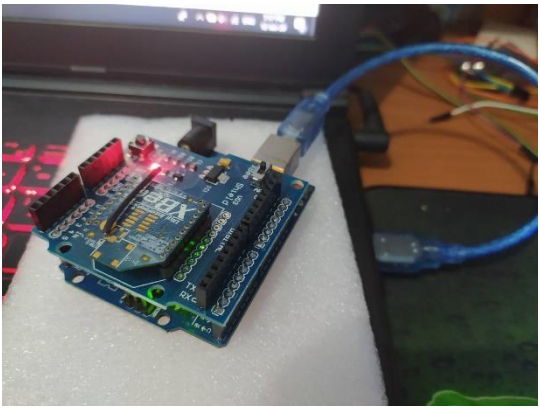
Dua *Arduino Uno* berperan sebagain *node router* yang merekam data suhu dan *particle per milimeter* menggunakan sensor suhu dan TDS Meter. Data yang didapatkan dari *node router*, kemudian dikirim ke *node coordinator* menggunakan protokol *ZigBee*. Ketika *node ZigBee Coordinator* telah menerima data dari *node ZigBee Router*, selanjutnya dengan bantuan dari *script python* yang dijalankan akan mengunggah data tersebut ke *web service*. Perancangan serta implementasi perangkat keras *node Zigbee Coordinator* akan dijabarkan lebih mendetail pada bagian 4.2.1, sedangkan implementasi perangkat keras *node ZigBee Router* akan dijabarkan lebih mendetail pada bagian 4.2.2.

4.2.1 Perangkat *Node ZigBee Coordinator*

Perangkat *ZigBee Coordinator* tersusun dari *Arduino Uno*, *XBee Shield*, dan modul *XBee S2 Pro*. *ZigBee Coordinator* berperan aktif dalam menerima data dari *ZigBee Router* dan juga mengirimkan data *serial* untuk dibaca *script python*. Pada gambar 4.1 diperlihatkan perancangan dari *ZigBee Coordinator*, sedangkan pada Gambar 4.2 diperlihatkan implementasi dari node *ZigBee Coordinator*.



Gambar 4.1 Perancangan *Node Zigbee Coordinator*

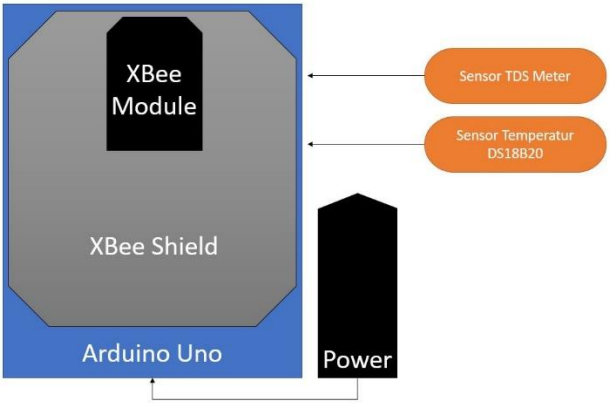


Gambar 4.2 Implementasi *Node Zigbee Coordinator*

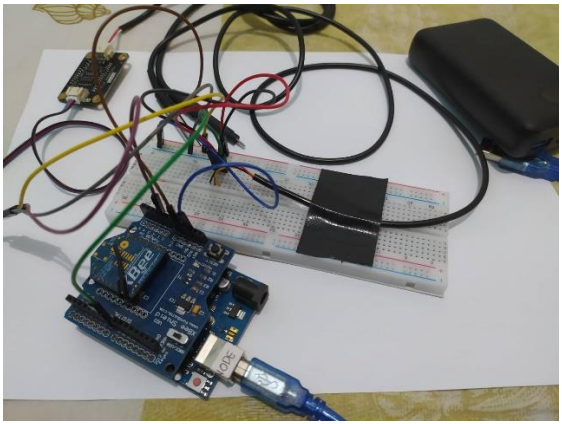
4.2.2 Perangkat *Node ZigBee Router*

Perangkat *ZigBee Router* tersusun dari *Arduino Uno*, sensor TDS Meter, sensor temperatur DS18B20, *Xbee Shield*, dan modul *Xbee S2 Pro*. *ZigBee Router* berperan aktif dalam merekam data

sensor PPM dan suhu sekaligus mengirim data tersebut menuju *ZigBee Coordinator*. Pada Gambar 4.3 diperlihatkan perancangan dari *Zigbee Router*, dan pada Gambar 4.4 implementasi *ZigBee Router*.



Gambar 4.3 Perancangan Node *Zigbee Router*

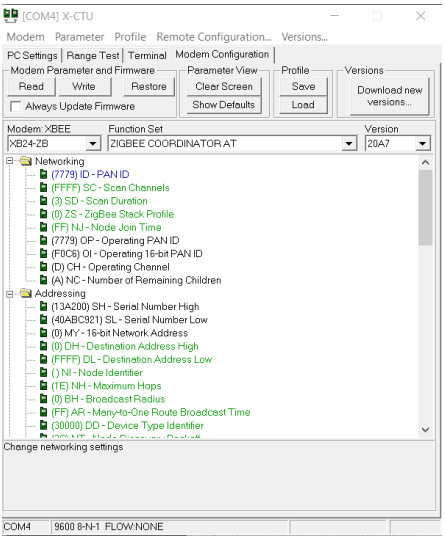


Gambar 4.4 Implementasi *Node Zigbee Router*

4.3 Implementasi Perangkat Lunak pada Perangkat Keras

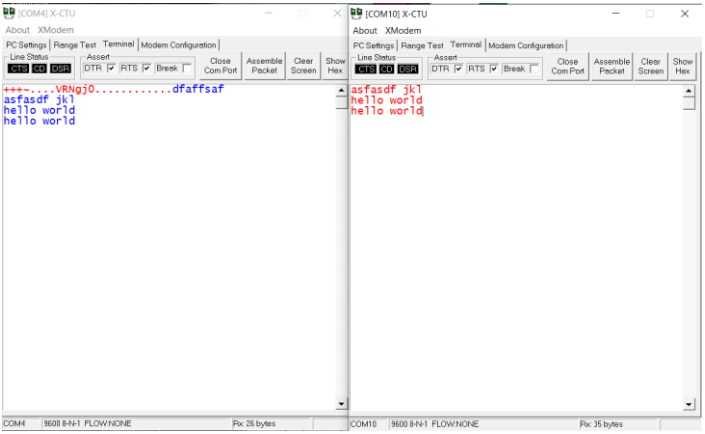
Implementasi perangkat lunak di *Arduino Uno* pada sistem pengawasan kualitas air terdapat dalam beberapa fungsi, yaitu sebagai pusat penerimaan sensor TDS Meter dan sensor suhu tahan air DS18B20, sebagai pusat pengiriman dan penerimaan data melalui *Xbee S2 Pro*, sebagai pusat pengiriman data ke *server*.

Sebelum mengimplementasikan keseluruhan perangkat lunak pada perangkat keras, yang amat penting untuk diperhatikan adalah melakukan uji coba kerja modul *XBee*. Dalam melakukan pengiriman data dari *node Router* ke *Coordinator*, modul *XBee* harus dikonfigurasi terlebih dahulu, agar dapat saling berkomunikasi satu sama lain. Untuk itu, dilakukan konfigurasi terlebih dahulu dengan aplikasi X-CTU, yaitu aplikasi yang didesain untuk membuat pengembang dapat berinteraksi dengan modul *XBee* dengan antarmuka grafis yang mudah digunakan. Aplikasi X-CTU yang digunakan penulis adalah versi lama dari X-CTU.



Gambar 4.5 Perangkat Lunak X-CTU

Untuk implementasi konfigurasi modul *XBee*, yang perlu diperhatikan adalah jenis modem yang digunakan, pengaturan fungsi yang digunakan setiap modem, dan identitas dari tiap modem. Pada penelitian ini, digunakan konfigurasi modem XBee XB24-ZB dengan versi 20A7. Satu modem yang akan digunakan sebagai *node Coordinator* diatur fungsinya sebagai *ZigBee Coordinator API*. Sedangkan untuk modem lainnya yang akan menjadi *node Router*, digunakan fungsi *ZigBee Router AT*. PAN ID yang digunakan setiap *node* harus sama, agar dapat saling berkomunikasi satu sama lain. Pengujian komunikasi kemudian dilakukan dengan membuka *tab Terminal* pada kedua jendela *node Coordinator* dan *Router*. Jika karakter yang diketikkan pada salah satu *node* dapat ditampilkan pada *node* lainnya, itu berarti komunikasi *ZigBee* sudah berhasil dijalankan.



Gambar 4.6 Percobaan Komunikasi Modul *Xbee*

4.4 Implementasi Inisialisasi Sensor TDS Meter dan Temperatur

Pseudocode 4.1 merupakan *pseudocode* implementasi inisialisasi sensor TDS Meter dan temperatur tahan air yaitu untuk mengonfigurasi *register-register* yang terdapat pada sensor

tersebut. Hal pertama yang dilakukan adalah melakukan inisialisasi kedua sensor, menyalakan kedua sensor untuk kemudian memanggil fungsi untuk melakukan pembacaan sensor.

```
Prosedur Sistem Pengawasan Kualitas Air
Inisialisasi Sensor Orientasi
Masukan: -
OneWire onewire(pin::one_wire_bus)
DallasTemperature dalllasTemperature(&oneWire)

Function Setup
{
 dallasTemperature.begin()
}

Function Loop
{
 readTdsQuick()
 delay1000
}

Keluaran: -
```

Pseudocode 4.1 *Pseudocode* Inisialisasi Sensor Orientasi

4.5 Implementasi Perangkat Lunak untuk Membaca Sensor Temperatur dan TDS Meter

Implementasi perangkat lunak sensor temperatur dan TDS meter pada Arduino dilakukan dengan menggunakan *library* dari *Dallas Temperature*, *OneWire*, dan *GravityTDS*. *Library* dari *Dallas Temperature* digunakan untuk melakukan pengukuran suhu dari sensor temperatur DS18B20. Kemudian, *library OneWire* digunakan agar mikrokontroler Arduino Uno dapat mengakses sensor yang hanya memiliki satu kabel, hal ini berguna karena sensor temperatur DS18B20 hanya memiliki satu kabel. Selain itu, *library GravityTDS* digunakan untuk mikrokontroler mengakses sensor TDS Meter. Pada penilitian ini digunakan *baud rate* 9600 untuk *Arduino*. *Baud rate* adalah banyaknya symbol yang

ditransfer selama 1 detik untuk transmisi data *serial*. Implementasi perangkat lunak untuk membaca sensor temperatur dan PPM pada *Arduino* dapat dilihat pada Pseudocode 4.2

```
Prosedur Sistem Pengawasan Kualitas Air
Mendapatkan Nilai Sensor
Masukan: -
Function readTdsQuick()
{
 dallasTemperature.requestTemperatures()
 sensor::waterTemp =
dallasTemperature.getTempCByIndex(0)
 float rawEc = analogRead(pin::tds_sensor) *
device::aref / 1024.0
 float temperatureCoefficient = 1.0 + 0.02 *
(sensor::waterTemp - 25.0)
 sensor::ec = (rawEc/temperatureCoeffient) *
sensor::ecCalibration
 sensor::tds = (133.42 * pow(sensor::ec, 3) -
255.86 * sensor::ec * sensor::ec + 857.39 *
sensor::ec

 print nodeId sensor::tds sensor::ec
sensor::waterTemp
}
Keluaran: ID node, nilai ppm, ec, suhu
```

Pseudocode 4.2 *Pseudocode* Mendapatkan Nilai Sensor

4.6 Implementasi Perangkat Lunak untuk Mengirim Data

Data yang akan dikirim berupa angka yang menandakan dari *node* mana dia berasal, data sensor dari tiap *node ZigBee Router* kepada *node ZigBee Coordinator*, dan data yang dikirim oleh *node router* ke *ZigBee Coordinator* kepada *server*.

4.6.1 Implementasi Menerima Data dari *Node Zigbee Router* oleh *Node Zigbee Coordinator*

Pada penelitian ini, *mode* pada *node ZigBee Router* kita konfigurasi menjadi *mode Router AT*, sehingga untuk proses berkirim data kita memanfaatkan kelas *Serial* pada *Arduino*.

```
Prosedur Sistem Pengawasan Kualitas Air
Penerimaan data Node ZigBee Router oleh Node ZigBee
Coordinator
Masukan: -
Function loop
{
 If (Serial.available() > 0)
  {
   Serial.write(Serial.read())
  }
}
Keluaran: ID node, nilai ppm, ec, suhu
```

Pseudocode 4.3 *Pseudocode* Penerimaan data *node ZigBee Router* oleh *node ZigBee Coordinator*

4.6.2 Implementasi Mengirim Data dari *Node Zigbee Coordinator* kepada *Server*

Dengan memanfaatkan sebuah *script* menggunakan *Python*, pada penelitian ini data sensor yang diterima *node Coordinator* akan dikirim ke *server* dengan bantuan *cursor* yang dibuka *Python* menuju *database*. Pada *script* ini digunakan modul *dbConn*. *DbConn* merupakan sebuah *helper class* yang berfungsi menyambungkan *python* dengan *database* PostgreSQL menggunakan *adapter Psycopg*. Kemudian digunakan *class cursor* untuk mengeksekusi perintah PostgreSQL dalam *session database*. Cara *script* ini mengirim data ke *server* sendiri adalah dengan membaca *serial* yang dikeluarkan *node Coordinator*, kemudian memotong-motongnya dengan perintah *data.split*

berdasarkan adanya spasi, kemudian mengirimnya menggunakan *cursor* dan *dbConn*. Implementasi perangkat lunak untuk mengirim data ditunjukkan oleh Pseudocode 4.4

```
Prosedur Sistem Pengawasan Kualitas Air
Mengirim Data ke Server
Masukan: ID node, nilai ppm, ec, suhu
dbConn =
MySQLdb.connect("server","username","password","tab
el")

while True:
  cursor = dbConn.cursor()
  device = 'serial node'
  Arduino = serial.Serial(device, baud)

  Time.sleep(1)
  Data = Arduino.readline()
  Print data
  Pieces = data.split(" ")

  If pieces[0] == 2 then
    Cursor.execute(perintah sql ke node2)
    DbConn.commit()
    Cursor.close()
  Elif pieces[0] == 1 then
    Cursor.execute(perintah sql ke node2)
    DbConn.commit()
    Cursor.close()
  Else
    Print "node tak ada"
  Time.sleep(10)

Keluaran: ID node, nilai ppm, ec, suhu
```

Pseudocode 4.4 *Pseudocode* Mengirim Data ke *Server*

4.7 Implementasi Basis Data Sistem

Sesuai dengan tahap perancangan di Bab sebelumnya, sistem ini akan menggunakan 2 tabel yang digunakan untuk

menyimpan *log* dari kondisi air. Pada Gambar 4.7 terdapat 5 atribut. Atribut *date*, *ppm*, dan *ec* didapatkan dari *node ZigBee Router*.

Name	Type	Collation	Attributes	Null	Default
id	int(11)			No	*None*
date	timestamp			No	CURRENT_TIMESTAMP
ppm	float			No	*None*
ec	float			No	*None*
temperature	float			No	*None*

Gambar 4.7 Implementasi Basis Data Sistem

.

BAB V
HASIL UJI COBA DAN EVALUASI

Bab ini membahas pengujian dan evaluasi pada sistem pengawasan kualitas air yang dikembangkan. Sistem akan diuji coba fungsionalitas dan performanya dengan menjalankan skenario yang telah ditentukan. Hasil evaluasi menjabarkan tentang rangkuman hasil pengujian pada bagian akhir bab ini. Uji coba akan dilakukan dengan dua tipe pengujian yaitu uji coba fungsionalitas dan uji coba performa. Pengujian fungsionalitas meliputi uji coba setiap bagian perangkat keras yang dirangkai pada Arduino dan uji coba keseluruhan sistem. Pengujian performa meliputi akurasi untuk menentukan seberapa baik kualitas air.

5.1 Lingkungan Uji Coba

Uji coba fungsionalitas dilakukan untuk mengetahui fungsi dasar masing-masing komponen baik perangkat keras yang dirangkai berjalan sebagaimana mestinya. Pada subbab ini akan dijelaskan mengenai uji coba Sistem Pengawasan Kualitas Air. Rangkaian *ZigBee Router* yang terdiri dari *Arduino Uno*, *Gravity TDS Sensor*, sensor temperatur DS18B20, *XBee Shield*, dan modul *XBee S2* yang dihubungkan dengan *Power Bank*. Setiap node *ZigBee Router* tersebut ditaruh disebelah sebuah gelas yang terisi air, yang mana kemudian kedua *probe* sensor dimasukkan kedalam air. Kemudian untuk rangkaian *ZigBee Coordinator* yang terdiri dari *Arduino Uno*, *XBee Shield*, dan modul *Xbee S2* dihubungkan dengan laptop. *Node ZigBee Coordinator* tersebut diletakkan disebelah laptop, dan terhubung dengan laptop. Data dari sensor temperatur dan TDS Meter kemudian dibandingkan dengan *threshold* untuk menentukan keputusan. Apabila sistem mendeteksi besarnya jumlah *Particle Per Milimeter* (PPM) melebihi jumlah aman air, sebuah peringatan akan muncul di halaman web pada aplikasi web. Pengiriman data dilakukan melalui *script Python* yang menangkap data *serial* yang

dikeluarkan oleh *node Coordinator*. Lingkungan uji coba memiliki spesifikasi seperti pada tabel dibawah ini:

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

Perangkat Keras (*Server*)	
Hosting	localhost
Perangkat ZigBee	
ZigBee Koordinator	Rangkaian dari :
	- Arduino UNO
	- Xbee Shield
	- Xbee S2 Modul
ZigBee Router	Rangkaian dari:
	- Arduino UNO
	- Gravity TDS Meter Sensor
	- Sensor Temperatur Tahan Air DS18B20
	- Xbee Shield
	- Modul Xbee S2
	- Power Bank
	- Breadboard

5.2 Uji Coba Fungsionalitas

Uji coba fungsionalitas bertujuan untuk menguji apakah fungsi-fungsi utama pada sistem ini berhasil di implementasikan dan berjalan sesuai dengan yang diharapkan.

5.2.1 Pengujian Komunikasi *Node ZigBee Router* dengan *Node ZigBee Coordinator*

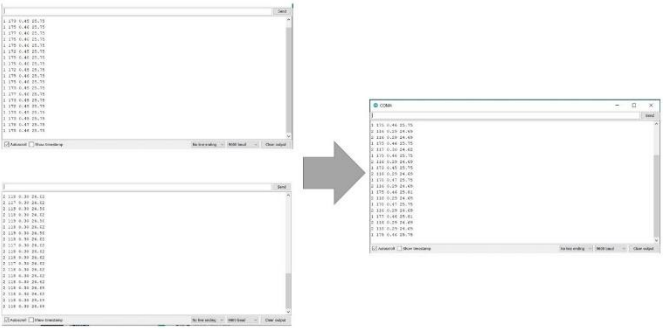
Pengujian fungsi komunikasi antara *node ZigBee Router* dengan *node ZigBee Coordinator* dilakukan dengan cara semua *node ZigBee Router* mengirimkan data pembacaan sensor *Particle per Milimeter* dan temperatur yang dihasilkan oleh masing-masing sensor TDS Meter dan suhu, kemudian dikirim secara terus menerus dan secara bersamaan kepada *node ZigBee Coordinator*.

Tabel 5.2 Skenario Uji Coba Fungsionalitas Komunikasi *Node ZigBee Router* dengan *Node ZigBee Coordinator*

ID	**UJ-01**
Nama	Uji Coba Komunikasi Node Zigbee Router dengan Node ZigBee Coordinator
Tujuan Uji Coba	Menguji fungsionalitas sistem untuk mengirimkan data sensor dari node Zigbee Router menuju node Zigbee Coordinator.
Kondisi Awal	Node ZigBee Router dan ZigBee Koordinator diaktifkan
Skenario	Semua node dijalankan, kemudian node ZigBee Router akan menghasil nilai sensor ppm dan suhu, setelah itu dikirm ke node ZigBee Coordinator
Masukan	Data sensor ppm, ec, dan suhu
Keluaran	Data sensor ppm, ec, dan suhu dari 2 node.
Hasil Uji Coba	Node ZigBee Coordinator berhasil menerima semua nilai sensor ppm, ec, dan suhu dari 2 node ZigBee Router

Tabel 5.2 diatas merupakan rincian prosedur pengujian Komunikasi *Node ZigBee Router* dengan *Node ZigBee Coordinator*. Pada gambar Gambar 5.1 ditampilkan proses

komunikasi antar 2 *node ZigBee Router* dengan *node ZigBee Coordinator*.



Gambar 5.1 Hasil Uji Coba Fungsionalitas Komunikasi *ZigBee*

5.2.2 Pengujian Pengiriman Data Kepada *Server*

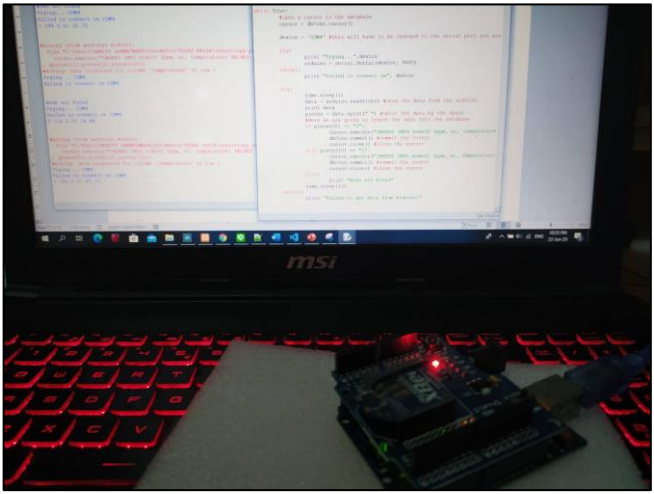
Pengujian fungsi pengiriman data dari *node ZigBee Coordinator* kepada *server* dilakukan dengan cara *node ZigBee Coordinator* mengirimkan data ke *database server* dengan menggunakan *script Python* yang menggunakan *cursor* dan *dbConn* yang telah dipersiapkan.

Tabel 5.3 Skenario Uji Coba Fungsionalitas Pengiriman Data Kepada *Server*

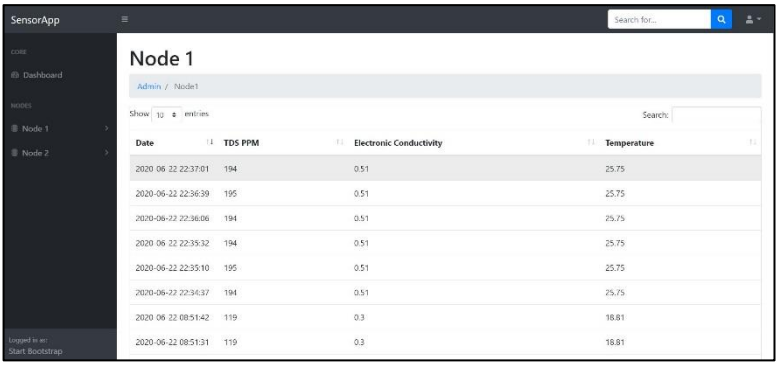
ID	**UJ-03**
Nama	Uji Coba Pengiriman Data kepada Server
Tujuan Uji Coba	Menguji fungsionalitas sistem untuk mengirimkan data dari node ZigBee Coordinator menuju server.
Kondisi Awal	Node ZigBee Coordinator diaktifkan

Skenario	Node ZigBee Coordinator mengirimkan data kepada database server melalui script yang telah disediakan
Masukan	Data String
Keluaran	Data String
Hasil Uji Coba	Node ZigBee Coordinator berhasil mengirim data kepada database server dan berhasil ditampilkan di halaman utama.

Tabel 5. diatas merupakan rincian prosedur pengujian Pengiriman Data Kepada Server. Pada Gambar 5. dan Gambar 5. ditampilkan hasil dari uji coba tersebut.



Gambar 5.2 Hasil Uji Coba Fungsionalitas Pengiriman Data kepada *Server*



Gambar 5.3 Hasil Uji Coba Fungsionalitas Pengiriman Data kepada *Server*

5.3 Uji Coba Performa

Uji coba performa sistem dilakukan untuk mengetahui seberapa akurat dan tepat sistem yang dibangun. Uji coba performa sistem yang dilakukan meliputi berapa keputusan yang dihasilkan bersesuaian dengan kondisi sebenarnya, seberapa lama *delay* yang dibutuhkan untuk sistem peringatan, dan seberapa banyak sistem peringatan yang bekerja dengan semestinya.

Pada uji coba performa ini dilakukan sebanyak sepuluh kali percobaan. Di setiap uji cobanya, dibuat skenario pada 1 menit pertama untuk kondisi air biasa, kemudian setelahnya dikondisikan terjadi penambahan kadar garam, dan setelahnya lagi dikondisikan tambahan kadar garam lagi. Pada setiap aktifitas tambahan, seharusnya sistem akan memberi tampilan naiknya kondisi air dari bagus menjadi sedang setelah terjadinya pengiriman data ke *server*. Kemudian jika kadar garam ditambah lagi, tampilan yang ditampilkan pada halaman web akan berbeda, yaitu menginformasikan kondisi air lebih buruk daripada sebelumnya.

Tabel 5.4 Skenario Uji Coba Performa Kondisi Air Normal

Nama	Uji coba akurasi kondisi air normal
Tujuan	Mengetahui akurasi sistem menentukan kondisi air normal/bagus
Skenario	Uji coba dilakukan dengan membiarkan lingkungan uji coba tanpa dikondisikan apapun.

Tabel 5.5 Skenario Uji Coba Performa Kondisi Air Kotor

Nama	Uji coba akurasi kondisi air kotor
Tujuan	Mengetahui akurasi sistem menentukan kondisi dimana air dalam kondisi kotor
Skenario	Uji coba dilakukan dengan mengkondisikan lingkungan uji coba dengan memasukkan sedikit garam yang dapat meningkatkan kadar ppm.

5.3.1 Percobaan Pertama

Tabel 5.6 Uji Coba Performa Pertama

Uji Coba Performa - Pertama				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	06.21	-	-	06.21
Tambahan #1	06.22	06.23	06.24	06.23
Tambahan #2	06.23	06.24	-	06.24

Pada Tabel 5. dapat dilihat hasil pengujian performa pertama. Di dapatkan informasi bahwa sistem berhasil mengkondisikan status awal dan pengiriman data *server* ketika kondisi normal. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengirim data ke *server* dan

menampilkan pergantian kondisi. Namun pada longsor susulan, sistem berhasil mengirimkan data ke *server*, namun pergantian status pada *node* kedua baru terjadi.

5.3.2 Percobaan Kedua

Tabel 5.7 Uji Coba Performa Kedua

Uji Coba Performa - Kedua				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	06.42	-	-	06.42
Tambahan #1	06.43	06.44	06.43	06.43
Tambahan #2	06.44	06.45	06.45	06.45

Pada Tabel 5.7 dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, dan status kedua *node* berhasil berubah.

5.3.3 Percobaan Ketiga

Tabel 5.8 Uji Coba Performa Ketiga

Uji Coba Performa - Ketiga				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	06.54	-	-	-
Tambahan #1	06.55	06.55	06.55	06.55
Tambahan #2	06.56	06.56	06.56	06.56

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, dan status kedua *node* berhasil diganti.

5.3.4 Percobaan Keempat

Tabel 5.9 Uji Coba Performa Keempat

Uji Coba Performa - Keempat				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	07.00	-	-	7.01
Tambahan #1	07.02	07.02	07.02	07.02
Tambahan #2	07.03	07.03	07.03	07.03

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, dan status kedua *node* berhasil diganti.

5.3.5 Percobaan Kelima

Tabel 5.10 Uji Coba Performa Kelima

Uji Coba Performa - Kelima

Keadaan	Waktu Terjadi	Pergantian Status N1	Pergantian Status N2	Kirim ke Server
Normal	07.13	-	-	7.13
Tambahan #1	07.14	07.14	07.14	07.14
Tambahan #2	07.15	-	07.16	07.16

Pada Tabel 5. dapat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Namun pada penambahan partikel kedua, *node* pertama tidak melakukan pergantian status, meskipun *node* kedua melakukan pergantian dan data berhasil masuk ke *server*.

5.3.6 Percobaan Keenam

Tabel 5.11 Uji Coba Performa Keenam

Uji Coba Performa - Keenam				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	07.23	-	-	7.23
Tambahan #1	07.24	07.24	-	07.24
Tambahan #2	07.25	07.25	07.25	07.25

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan

mengirim data ke *server*. Namun status di *node* kedua kembali ke awal, jadi perubahan tidak dihitung. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, dan status kedua *node* berhasil diganti.

5.3.7 Percobaan Ketujuh

Tabel 5.12 Uji Coba Performa Ketujuh

Uji Coba Performa - Ketujuh				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	07.33	-	-	7.33
Tambahan #1	07.34	07.35	07.36	07.35
Tambahan #2	07.36	07.37	-	07.36

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, tetapi *node* kedua statusnya tidak berganti. Dalam testing ini juga terjadi delay yang agak lama untuk kedua *node*, terutama *node* kedua.

5.3.8 Percobaan Kedelapan

Tabel 5.13 Uji Coba Performa Kedelapan

Uji Coba Performa - Kedelapan				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	07.43	-	-	7.43

Tambahan #1	07.44	07.44	07.44	07.44
Tambahan #2	07.45	07.45	07.45	07.45

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, dan status kedua *node* berhasil diganti.

5.3.9 Percobaan Kesembilan

Tabel 5.14 Uji Coba Performa Kesembilan

Uji Coba Performa - Kesembilan				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	07.53	-	-	7.53
Tambahan #1	07.54	07.54	07.54	07.54
Tambahan #2	07.56	07.55	07.55	07.55

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke *server*. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua *node*, dan mengirim data ke *server*. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke *server*, dan status kedua *node* berhasil diganti.

5.3.10 Percobaan Kesepuluh

Tabel 5.15 Uji Coba Performa Kesepuluh

Uji Coba Performa - Kesepuluh				
Keadaan	**Waktu Terjadi**	**Pergantian Status N1**	**Pergantian Status N2**	**Kirim ke Server**
Normal	08.03	-	-	08.03
Tambahan #1	08.04	08.04	08.04	08.04
Tambahan #2	08.05	08.05	08.05	08.05

Pada Tabel 5. dapat dilihat hasil pengujian performa kedua. Di dapatkan informasi bahwa sistem berhasil mengirimkan data ke server. Kemudian tatkala kondisi penambahan partikel pertama, sistem berhasil mengganti status kedua node, dan mengirim data ke server. Dan pada penambahan partikel kedua, sistem berhasil melakukan pengiriman data ke server, dan status kedua node berhasil diganti.

5.4 Evaluasi Pengujian

Berdasarkan hasil pengujian fungsionalitas dan pengujian performa maka pada subbab ini akan menjelaskan evaluasi pengujian seperti berikut

5.4.1 Evaluasi Pengujian Fungsionalitas

Hasil pengujian fungsionalitas dapat dilihat pada subbab sebelumnya. Berdasarkan Gambar 5.1, Gambar 5.2, dan Gambar 5.3, dapat dilihat hasil uji coba fungsionalitas pada aplikasi dapat dijalankan dengan baik. Sehingga dapat ditarik kesimpulan bahwa fungsionalitas aplikasi telah bekerja sesuai dengan harapan dan menampilkan proses serta hasil yang benar.

Tabel 5.16 Evaluasi Pengujian Fungsionalitas

Nama	Hasil
Pengujian Komunikasi Node Zigbee Router dengan Node Zigbee Coordinator	Berhasil
Pengujian Pengiriman Data Kepada Server	Berhasil

5.4.2 Evaluasi Pengujian Performa

Dari sepuluh kali pengujian coba performa, maka didapatkan rangkuman hasil pengujian performa sebagai berikut

Tabel 5.17 Evaluasi Pengujian Performa Kondisi Air Biasa

Keadaan Air Biasa		
Fitur	**Total Seharusnya Terjadi**	**Total Terjadi**
Pergantian Status	10	10

Tabel 5.18 Evaluasi Pengujian Performa Kondisi Air Kotor

Keadaan Air Kotor		
Fitur	**Total Seharusnya Terjadi**	**Total Terjadi**
Pergantian Status	40	36

Dengan menganalisis semua pengujian performa yang telah dilakukan, maka didapatkan akurasi untuk sistem monitoring kualitas air ini. Untuk menentukan besaran nilai akurasi dan presisi sistem, maka dibuat pembagian seperti pada Tabel 5.19 dan Tabel 5. dibawah ini

Tabel 5.19 Rumusan Confusion Matrix

Keputusan Sistem	Kondisi Sebenarnya	
	True	**False**
True	TP	FP
False	FN	TN

TP = *True Positive*
FN = *False Negative*
FP = *False Positive*
TN = *True Negative*

$$Presisi = \frac{TP}{TP + FP}$$

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

Tabel 5.20 Confusion Matrix Sistem Monitoring Kualitas Air

Keputusan Sistem	Kondisi Sebenarnya	
	Terdeteksi	**Tidak Terdeteksi**
Terdeteksi	36	4
Tidak Terdeteksi	0	10

$$Presisi = \frac{36}{36 + 4} = 0.9 = 90\%$$

$$Akurasi = \frac{36 + 10}{36 + 10 + 4 + 0} = 0.92 = 92\%$$

Berdasarkan data dan perhitungan diatas, nilai presisi dari sistem yaitu 90%, dan nilai keakuratan keseluruhan sistem dalam menampilkan keputusan yang benar yaitu 92%.

[Halaman ini sengaja dikosongkan]

BAB VI
KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat juga saran yang ditujukan untuk pengembangan perangkat lunak nantinya.

7.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian yang dilakukan, maka dapat disimpulkan sebagai berikut

1. Metode yang digunakan untuk mendeteksi kualitas air dapat dilakukan dengan beberapa *node router* yang mengirimkan data *serial* menggunakan *ZigBee* kepada satu *node coordinator* yang kemudian mengirimkan datanya kepada *server*.
2. Sistem dapat mendeteksi kualitas air dengan mengandalkan sensor TDS Meter yang dilengkapi sensor suhu
3. Sistem dapat memantau kualitas air melalui aplikasi dengan *script python*, yang kemudian diolah dalam web.

7.2 Saran

Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan. Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang.

1. Pada penelitian selanjutnya, agar hasil yang lebih lengkap dan informatif, baiknya menambahkan sensor lainnya, seperti sensor turbiditas / kekeruhan dan tingkat keasaman air / pH.

2. Pengembangan sistem pengawasan kualitas air ini disarankan untuk menggunakan jumlah *node ZigBee Router* yang lebih banyak, agar dapat mengawasi wilayah yang lebih luas.
3. Guna pengembangan sistem kualitas air kedepannya, ada baiknya difasilitasi dengan aplikasi berbasis *Android* yang dapat memantau secara langsung data kondisi air, sekaligus dapat memberikan notifikasi jika ada salah satu variabel berada dalam kondisi yang tidak layak.

DAFTAR PUSTAKA

[1] "Water Quality." [Online]. Available: https://www.un.org/waterforlifedecade/quality.shtml [Accessed: 4-Jun-2020].

[2] "UN-Water Policy Brief: Water Quality." [Online]. Available: https://www.unwater.org/publications/un-water-policy-brief-water-quality/ [Accessed: 4-Jun-2020].

[3] Dargie, W. dan Poellabauer, C. 2010. Fundamentals of Wireless Sensor Networks: Theory and Practice. John Wiley and Sons.

[4] A. Kadir, Buku Pintar Pemrograman Arduino, Yogyakarta: Mediakom, 2014.

[5] "Gravity: Analog TDS Sensor/Meter for Arduino - DFRobot" [Online]. Available: https://www.dfrobot.com/product-1662.html [Accessed: 06-Jun-2020].

[6] "Waterproof DS18B20 Digital Temperature Sensor For Arduino - DFRobot" [Online]. Available: https://www.dfrobot.com/product-689.html [Accessed: 07-Jun-2020].

[7] "XBee 2mW PCB Antenna - Series 2 (ZigBee Mesh) - WRL-11217 - SparkFun Electronics." [Online]. Available: https://www.sparkfun.com/products/retired/11217. [Accessed: 17-Jun-2020].

[8] S. R. Ahamed, "The role of zigbee technology in future data communication system," *J. Theor. Appl. Inf. Technol.*, vol. 5, no. 2, p. 129, 2009.

[9] "CodeIgniter", *Wikipedia*. [Accessed: 07-Jun-2020].

LAMPIRAN A KODE SUMBER

```
#include <DallasTemperature.h>
#include <OneWire.h>
#include <GravityTDS.h>

namespace pin {
  const byte tds_sensor = A1;
  const byte one_wire_bus = 7; // Dallas
Temperature Sensor
}

namespace device {
  float aref = 5;
}

namespace sensor {
  float ec = 0;
  unsigned int tds = 0;
  float waterTemp = 0;
  float ecCalibration = 1;
}
```

Pseudocode A.1 *Pseudocode* Inisialisasi Variabel Global *Node ZigBee Router*

```
OneWire oneWire(pin::one_wire_bus);
DallasTemperature dallasTemperature(&oneWire);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dallasTemperature.begin();

}
```

Pseudocode A.2 *Pseudocode* Fungsi *Setup* untuk Mempersiapkan Sensor *Node ZigBee Router*

```
void loop() {
```

```
  // put your main code here, to run repeatedly:
  readTdsQuick();
  delay(1000);
}

void readTdsQuick() {
  dallasTemperature.requestTemperatures();
  sensor::waterTemp =
dallasTemperature.getTempCByIndex(0);
  float rawEc = analogRead(pin::tds_sensor) *
device::aref / 1024.0; // read the analog value
more stable by the median filtering algorithm, and
convert to voltage value
  float temperatureCoefficient = 1.0 + 0.02 *
(sensor::waterTemp - 25.0); // temperature
compensation formula: fFinalResult(25C) =
fFinalResult(current)/(1.0+0.02*(fTP-25.0));
  sensor::ec = (rawEc/temperatureCoefficient) *
sensor::ecCalibration; //temperature and
calibration compensation
  sensor::tds = (133.42 * pow(sensor::ec, 3) -
255.86 * sensor::ec * sensor::ec + 857.39 *
sensor::ec) * 0.5; // convert voltage value to tds
value
  Serial.print(1);
  Serial.print(" ");
  Serial.print(sensor::tds);
  Serial.print(" ");
  Serial.print(sensor::ec, 2);
  Serial.print(" ");
Serial.println(sensor::waterTemp, 2);
  Serial.println();
}
```

Pseudocode A.3 *Pseudocode* Fungsi untuk Pendeteksian Temperatur dan PPM *ZigBee Router*

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}
```

Pseudocode A.4 *Pseudocode* Inisialisasi Variabel Global *Node ZigBee Coordinator*

```
void loop() {
  // put your main code here, to run repeatedly:
  if (Serial.available() > 0) {
    Serial.write(Serial.read());
  }
}
```

Pseudocode A.5 *Pseudocode* Fungsi untuk Menerima Data *Node ZigBee Coordinator*

```
import serial
import MySQLdb
import time

#establish connection to MySQL. You'll have to
change this for your database
dbConn =
MySQLdb.connect("localhost","root","","sensor_data"
) or die ("could not connect to database")

while True:
       #open a cursor to the database
       cursor = dbConn.cursor()

       device = 'COM4' #this will have to be changed
to the serial port you are using

       try:
              print "Trying...",device
              arduino = serial.Serial(device, 9600)
       except:
              print "Failed to connect on", device
```

Pseudocode A.6 *Pseudocode* Inisialisasi *Python* Untuk Menerima *Serial Data* Dari *Node Coordinator*

```
time.sleep(1)
data = arduino.readline() #read the data from the
arduino
print data
```

```
pieces = data.split(" ") #split the data by the
space
  if pieces[0] == "2":
    cursor.execute("INSERT INTO node2 (humidity,
temperature, heat_index, ppm) VALUES
(%s,%s,%s,%s)",
(pieces[1],pieces[2],pieces[3],pieces[4]))
    dbConn.commit() #commit the insert
    cursor.close() #close the cursor
  elif pieces[0] == "1":
    cursor.execute("INSERT INTO node1 (humidity,
temperature, heat_index, ppm) VALUES
(%s,%s,%s,%s)",
(pieces[1],pieces[2],pieces[3],pieces[4]))
    dbConn.commit() #commit the insert
    cursor.close() #close the cursor
  else:
    print "Node not Found"
    time.sleep(2)
```

Pseudocode A.7 *Pseudocode* Fungsi untuk *Python* Memasukkan *Serial Data* ke *Server*

BIODATA PENULIS

Benito Danneswara Widyatama merupakan anak dari pasangan Bapak Agung Widyanto BKY dan Ibu Prathama Pradnja Paramita. Lahir di Jakarta pada tanggal 09 April 1996. Penulis menempuh pendidikan formal dimulai dari SDK Santa Theresia Balikpapan (2002-2006), SDK Fransiskus 2 Bandar Lampung (2006-2008), SMP Negeri 1 Purwokerto (2008-2009), SMPK Kolese Santo Yusup 2 Malang (2009-2011), SMAK Kolese Santo Yusup Malang (2011-2015) dan S1 Teknik Informatika ITS (2015-2020). Bidang studi yang ditekuni oleh penulis pada saat berkuliah di Teknik Informatika ITS adalah Komputasi Berbasis Jaringan (KBJ). Penulis aktif dalam organisasi seperti ITS Foreign Language Society (2015-2017). Penulis juga aktif dalam kegiatan kepanitiaan yaitu IFLS Inochi 2015-2016. Penulis memiliki hobby scale-modeling dan music. Penulis dapat dihubungi melalui email: benito.dannes@gmail.com.