

**RANCANG BANGUN WIRELESS SENSOR NETWORK UNTUK
MONITORING PENCEMARAN AIR SUNGAI MENGGUNAKAN
TOPOLOGI MESH NETWORK**

TUGAS AKHIR



Oleh :

Nama : Yoga Eka Saputra

NIM : 09.41020.0034

Program : S1 (Strata Satu)

Jurusan : Sistem Komputer

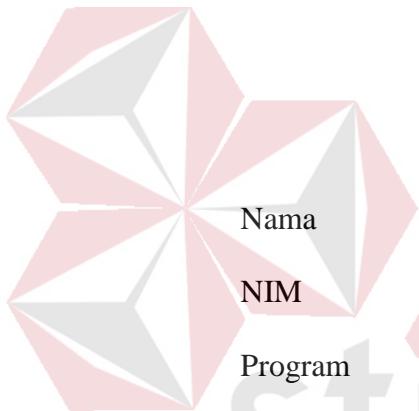
**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA
2014**

**RANCANG BANGUN *WIRELESS SENSOR NETWORK* UNTUK
MONITORING PENCEMARAN AIR SUNGAI MENGGUNAKAN
*TOPOLOGI MESH NETWORK***

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Komputer



Oleh :

Nama : Yoga Eka Saputra
NIM : 09.41020.0034
Program : S1 (Strata Satu)
Jurusan : Sistem Komputer

INSTITUT BISNIS
INFORMATIKA
stikom
SURABAYA

**SEKOLAH TINGGI
MANAJEMEN INFORMATIKA & TEKNIK KOMPUTER
SURABAYA
2014**

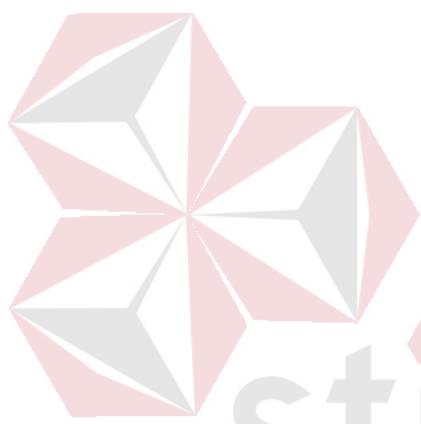


INSTITUT BISNIS
DAN INFORMATIKA

stikom
SURABAYA

*Belajarlah dengan sungguh-sungguh atas pelajaran
yang diberikan oleh dosen anda.*

*Jika tidak mampu,
maka belajarlah dengan sungguh-sungguh
atas pelajaran kehidupan yang diberikan oleh dunia kampus.*



INSTITUT BISNIS
DAN INFORMATIKA

stikom
Kupersembahkan kepada
Bapak dan Ibu tercinta
Bapak Dosen Wali yang selalu membimbingku
Beserta semua orang yang menyayangiku

Tugas Akhir
RANCANG BANGUN WIRELESS SENSOR NETWORK UNTUK
MONITORING PENCEMARAN AIR SUNGAI MENGGUNAKAN
TOPOLOGI MESH NETWORK

dipersiapkan dan disusun oleh

Yoga Eka Saputra

NIM :09.41020.0034

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji
pada :Agustus 2014

Pembimbing

I. Dr. Jusak

Penguji

I. Helmy Widyantara, S.Kom., M.Eng.

II. Ira Puspasari, S.Si., M.T.

Susunan Dewan Penguji

INSTITUT BISNIS
DAN INFORMATIKA

Stikom
SURABAYA

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Pantjawati Sudarmaningtyas, S.Kom., M.Eng.
Pembantu Ketua Bidang Akademik

PERNYATAAN

Dengan ini saya menyatakan dengan benar, bahwa Tugas Akhir ini adalah asli karya saya, bukan plagiat baik sebagian maupun apalagi keseluruhan. Karya atau pendapat orang lain yang ada dalam Tugas Akhir ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya.

Apabila dikemudian hari ditemukan adanya tindakan plagiat pada karya Tugas Akhir ini, maka saya bersedia untuk dilakukan pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Surabaya, 26 Agustus 2014



ABSTRAK

Kualitas sumber air dari sungai-sungai penting di Indonesia umumnya sudah tercemar oleh limbah organik yang berasal dari limbah penduduk, industri dan lain sebagainya. Pencemaran air berdampak luas, misalnya dapat meracuni sumber air minum, meracuni makanan hewan, ketidakseimbangan ekosistem sungai dan danau dan sebagainya.

Pemantauan pencemaran air sungai secara *real time* ini menggunakan teknologi *Wireless Sensor Network* (WSN). (WSN) merupakan jaringan nirkabel yang terdiri dari beberapa alat sensor yang saling bekerja sama untuk memonitor fisik dan kondisi lingkungan. Setiap node dalam WSN ini terdiri dari dengan modul mikrokontroler arduino uno, Xbee series 2 sebagai komunikasi nirkabel, dan sensor *analog pH meter kit*, sensor DS18B20 dan RTC (*Real Time Clock*).

Topologi *MESH* merupakan jalur komunikasi dimana masing-masing node dapat berkomunikasi dengan yang lainnya. Dalam *mesh*, node mempertahankan jalur komunikasi untuk kembali ke *gateway*, sehingga jika salah satu node router *down*, secara otomatis router data akan dilewatkan melalui jalur yang berbeda.

Dengan hasil pengujian didapatkan pengukuran pH air dengan sensor *analog pH meter kit* memiliki presentase kesalahan antara 0,37% sampai dengan 1,86% dan pengukuran suhu air menggunakan sensor DS18B20 memiliki presentase kesalahan antara 0,606% sampai dengan 0,73% Dengan hasil tersebut sistem ini dapat berjalan dengan baik dengan tingkat akurasi 98,14% untuk sensor pH dan 99,27% untuk sensor suhu.

KATA PENGANTAR

Pertama-tama penulis panjatkan puji dan syukur kepada Tuhan Yang Maha Esa karena berkat, rahmat, dan karuniaNyalah penulis dapat menyelesaikan penulisan Tugas Akhir ini dengan sebaik-baiknya. Penulis mengambil judul “Rancang Bangun Wireless Sensor Network Untuk Monitoring Pencemaran Air Sungai Menggunakan Topologi Mesh Network” ini sebagai salah satu syarat dalam menyelesaikan Tugas Akhir di Sekolah Tinggi Manajemen Informatika & Teknik Komputer Surabaya.

Dalam pelaksanaan Tugas Akhir serta pembuatan laporan Tugas Akhir ini, banyak sekali pihak yang telah membantu penulis sehingga Tugas Akhir dapat terlaksana dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada :

1. Seluruh guru yang telah mendidik penulis mulai kecil hingga saat ini sehingga penulis mendapatkan ilmu pengetahuan yang dapat menjadikan penulis menjadi seperti saat ini.
2. Pimpinan STIKOM Surabaya yang telah banyak memberikan motivasi serta teladan yang dapat membantu penulis selama menempuh pembelajaran hingga saat ini.
3. Bapak Dr. Jusak, selaku Kepala Program Studi Sistem Komputer STIKOM Surabaya dan dosen pembimbing pertama yang telah membantu serta mendukung setiap kegiatan sehingga pelaksanaan Tugas Akhir ini dapat berjalan dengan baik.

4. Bapak Yuwono Marta Dinata, S.T., M.Eng., selaku dosen pembimbing kedua sehingga penulis dapat melaksanakan Tugas Akhir ini dengan baik.
5. Bapak Harianto, S.Kom., M.Eng. selaku dosen wali dan para dosen lainnya yang telah membantu penulis jika mengalami berbagai macam kesulitan sehingga penulis dapat termotivasi untuk terus berusaha hingga Tugas Akhir ini terlaksana sesuai dengan harapan.
6. Teman-teman penulis yang telah mendampingi, memberi tempat saat penulis membutuhkan yang juga membantu dalam pelaksanaan Tugas Akhir ini.
7. Seluruh pihak yang tidak dapat penulis tuliskan satu persatu yang telah membantu penulis secara langsung maupun tidak langsung.

Banyak hal dalam laporan Tugas Akhir ini yang masih perlu diperbaiki lagi. Oleh karena itu penulis mengharapkan saran dan kritik yang dapat membangun dari semua pihak agar dapat menyempurnakan penulisan ini kedepannya. Penulis juga memohon maaf yang sebesar-besarnya jika terdapat kata-kata yang salah serta menyinggung perasaan pembaca. Akhir kata penulis ucapan banyak-banyak terima kasih yang sebesar-besarnya kepada para pembaca, semoga tulisan ini dapat bermanfaat bagi para pembaca.

Surabaya, Agustus 2014

Penulis

DAFTAR ISI

Halaman

ABSTRAKSI	v
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I	
PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	4
1.5 Sistematika Penulisan	4
BAB II	
LANDASAN TEORI	7
2.1 <i>Wireless Sensor Network (WSN)</i>	7
2.1.1 Arsitektur WSN	10
2.2 Arduino	11
2.2.1 Arduino Uno SMD R3.....	12
2.2.2 Daya (<i>Power</i>)	13
2.2.3 Memori	14
2.2.4 <i>Input</i> dan <i>output</i>	14
2.3 <i>Software Arduino IDE</i>	16

2.4 Bahasa Pemrograman Arduino	17
2.5 Sensor Analog pH Meter Kit	24
2.6 <i>Digital Temperature DS18B20 (WATERPROOF)</i>	26
2.7 <i>Real Time Clock (RTC)</i>	27
2.8 Xbee Series 2 Chip Antenna dan Xbee Pro Series 2 Wire Antenna	28
2.9 Komunikasi Serial Xbee Series 2	30
2.10 AT / <i>Transparent Mode</i>	31
2.11 <i>Coordinator, Router, dan End Device</i>	31
2.12 Xbee Usb Adapter dan Software X-CTU	32
2.13 Topologi <i>Point to Point</i>	39
2.14 Topologi <i>Point to Multipoint</i>	41
2.15 Xbee <i>Shield</i>	42
2.16 Topologi <i>Mesh Network</i>	44
2.17 <i>Visual Basic</i>	44
BAB III	
METODE PENELITIAN DAN PERANCANGAN SISTEM	47
3.1 Metode Penelitian	47
3.2 Desain Topologi	49
3.3 Perancangan Perangkat Keras	50
3.4 Pemrograman mikrokontroler Arduino pada <i>software</i> Arduino IDE	53
3.4.1 Membuat Skrip untuk <i>Node Router 1</i>	54
3.4.2 Membuat Skrip untuk <i>Node Router 2</i>	62
3.4.3 Membuat Skrip untuk <i>Node Coordinator</i>	63
3.4.4 Membuat Skrip untuk <i>Node End Device</i>	70

3.4.5 Pemrograman <i>visuak basic</i>	72
BAB IV	
HASIL PENGUJIAN DAN PENGAMATAN	78
4.1 Kebutuhan <i>Hardware</i> dan <i>Software</i>	78
4.1.1 Kebutuhan Perangkat Keras (<i>Hardware</i>)	78
4.1.2 Kebutuhan Perangkat Lunak (<i>Software</i>).....	78
4.2 Pengujian Sistem	79
4.3 Pengujian Sensor pH, sensor Suhu, Modul RTC dan <i>Visual Basic</i>	79
4.3.1 Peralatan	79
4.3.2 Prosedur Pengujian	80
4.3.3 Hasil Pengujian	81
4.4 Pengujian Protokol Komunikasi	92
4.4.1 Peralatan	92
4.4.2 Prosedur Pengujian	92
4.4.3 Hasil pengujian	93
4.5 Pengujian Jarak Kemampuan Pengiriman dan Penerimaan Xbee.....	100
4.5.1 Peralatan	101
4.5.2 Prosedur Pengujian	101
4.5.3 Hasil pengujian	102
4.5.4 Analisis Hasil Pengujian	104
BAB V	
KESIMPULAN DAN SARAN	106
5.1 Kesimpulan	106
5.2 Saran	107

DAFTAR PUSTAKA	108
LAMPIRAN	109
BIODATA PENULIS	124



DAFTAR TABEL

	Halaman
Tabel 2.1 Spesifikasi Arduino Uno SMD R3	13
Tabel 4.1 Hasil Pengujian Sensor pH, suhu dan Alat Ukur pada <i>router</i> 1	88
Tabel 4.2 Hasil Pengujian Sensor pH, suhu dan Alat Ukur pada <i>router</i> 2	88
Tabel 4.3 Hasil Pengujian Sensor pH, suhu dan Alat Ukur pada <i>router</i> 1	89
Tabel 4.4 Pengiriman data dari <i>router</i> 1 dan <i>router</i> 2 pada jarak 20 meter	102
Tabel 4.5 Pengiriman data dari <i>router</i> 1 dan <i>router</i> 2 pada jarak 40 meter	102
Tabel 4.6 Pengiriman data dari <i>router</i> 1 dan <i>router</i> 2 pada jarak 60 meter	102
Tabel 4.7 Pengiriman data dari <i>router</i> 1 dan <i>router</i> 2 pada jarak 80 meter	102
Tabel 4.8 Pengiriman data dari <i>router</i> 1 dan <i>router</i> 2 pada jarak 100 meter	103
Tabel 4.9 Selisih pengiriman data antara jarak 20 meter dan 100 meter dengan rumus $ Data\ 100\ meter - Data\ 20\ meter $	103

DAFTAR GAMBAR

Halaman

Gambar 2.1 Arsitektur WSN	7
Gambar 2.2 Komponen-komponen Penyusun <i>Node</i> dalam WSN	8
Gambar 2.3 Perkembangan Dimensi <i>Node</i> Sensor Terhadap Waktu	10
Gambar 2.4 Gambaran Arsitek <i>Wireless Sensor Network</i>	11
Gambar 2.5 Arduino Uno SMD R3 Sisi depan (Kiri) dan Belakang (Kanan)....	12
Gambar 2.6 Tampilan <i>Software Arduino IDE</i>	17
Gambar 2.7 Sensor Analog pH Meter Kit	25
Gambar 2.8 Digital <i>Temperature</i> DS18B20	26
Gambar 2.9 RTC	28
Gambar 2.10 Xbee Series 2 <i>Chip Antenna</i>	29
Gambar 2.11 Xbee Series 2 <i>Wire Antenna</i>	29
Gambar 2.12 Diagram Sistem Aliran Data UART pada Xbee	31
Gambar 2.13 Xbee Usb <i>Adapter</i> dan Kabel Mini Usb.....	33
Gambar 2.14 Tampilan <i>Software X-CTU</i>	33
Gambar 2.15 Kotak Diagram Respon	34
Gambar 2.16 <i>Tab Terminal</i>	35
Gambar 2.17 <i>Assemble Packet</i>	35
Gambar 2.18 <i>Tab PC Settings</i>	36
Gambar 2.19 <i>Tab Modem Configuration</i>	37
Gambar 2.20 Firmware Xbee Telah Terbaca.....	38
Gambar 2.21 Kotak Dialog <i>Get New Versions</i>	38
Gambar 2.22 <i>Save</i> dan <i>Load Modem Profile</i>	39

Gambar 2.23 Konfigurasi <i>Node Coordinator</i>	40
Gambar 2.24 Konfigurasi <i>Node End Device</i>	41
Gambar 2.25 Konfigurasi <i>Point to Multipoint</i>	42
Gambar 2.26 Xbee <i>Shield</i>	43
Gambar 2.27 Jumper pada Xbee <i>Shield</i>	43
Gambar 2.28 Topologi <i>Mesh Network</i>	44
Gambar 2.29 Tampilan Utama <i>Visual Basic 6.0</i>	45
Gambar 3.1 Blok Diagram Sistem	48
Gambar 3.2 Topologi <i>Point to Multipoint</i>	49
Gambar 3.3 Perancangan Perangkat Keras	51
Gambar 3.4 Diagram Alir Pembuatan Skrip pada <i>Node router 1</i>	54
Gambar 3.5 Format Penulisan Pesan	60
Gambar 3.6 Diagram Alir Pembuatan Skrip pada <i>Node router 2</i>	62
Gambar 3.7 Diagram Alir Proses <i>Node Coordinator</i>	64
Gambar 3.8 Diagram Alir Proses <i>Node End Device</i>	70
Gambar 3.9 Tampilan <i>software</i> dari visual basic.....	73
Gambar 3.10 <i>flowchart</i> program monitoring pencemaran air.....	74
Gambar 3.11 <i>flowchart</i> penerimaan dari data yang ber ID C1	76
Gambar 3.12 <i>flowchart</i> penerimaan dari data yang ber ID C2	77
Gambar 4.1 Pengujian Sensor ph dan Sensor Suhu	83
Gambar 4.2 Hasil Pengukuran Sensor pH, sensor suhu dan Pembacaan Waktu RTC pada <i>Node Router 1</i>	86
Gambar 4.3 Hasil Pengukuran Sensor pH, sensor suhu dan Pembacaan Waktu RTC pada <i>Node Router 2</i>	87

Gambar 4.4 Hasil tampilan software <i>visual basic</i>	90
Gambar 4.5 Data masuk yang diterima.....	91
Gambar 4.6 Data pH dari <i>node router 1</i> dan <i>node router 2</i>	91
Gambar 4.7 Data suhu dari <i>node router 1</i> dan <i>node router 2</i>	92
Gambar 4.8 Komunikasi dalam Sistem WSN Monitoring Pencemaran Air Sungai	93
Gambar 4.9 Pengiriman dan Penerimaan Data antar <i>Node Router</i>	98
Gambar 4.10 Hasil Pengiriman/Penerimaan/Rata-rata Data pada <i>Node Coordinator</i>	99
Gambar 4.11 <i>Error Checking</i>	104



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Kualitas sumber air dari sungai-sungai penting di Indonesia umumnya tercemar oleh limbah organik yang berasal dari limbah penduduk, industri dan lain sebagainya. Dengan kemampuan sensor yang ada sekarang serta pencemaran lingkungan yang terjadi disekitar kita maka digunakan sensor untuk mendeteksi pencemaran yang ada, terutama pencemaran air. Pencemaran air adalah suatu perubahan keadaan di suatu tempat penampungan air seperti danau, sungai, lautan dan air tanah akibat aktivitas manusia. Pencemaran air berdampak luas, misalnya dapat meracuni sumber air minum, meracuni makanan hewan, ketidakseimbangan ekosistem sungai dan danau, pengrusakan hutan akibat hujan asam, dan sebagainya. Menurut Wardhana (1995), beberapa komponen pencemar air yang berasal dari industri, rumah tangga dan pertanian dapat dikelompokkan sebagai bahan buangan yaitu bahan buangan padat, bahan buangan organik dan olahan bahan makanan, bahan buangan anorganik, bahan buangan cairan berminyak, bahan buangan berupa gas, dan bahan buangan zat kimia. Menurut Warlina (2004), cara mengatasi pencemaran air dapat dilakukan usaha preventif, misalnya dengan membuang sisa-sisa makanan dan bahan *organic* ke dalam tong sampah dan jangan dibuang di sungai.

Parameter kualitas air minum yang ditetapkan dalam PERMENKES 492/2010 terdiri dari parameter fisik, parameter kimiawi, parameter mikrobiologis. Parameter fisik berupa bau, warna, jumlah zat padat terlarut, kekeruhan, rasa dan suhu. Parameter kimiawi berupa alumunium, besi, pH, seng,

sulfat, tembaga dan mangan. Parameter mikrobiologis berupa total koliform (MPN). Ada 2 parameter utama yang diamati pada tugas akhir ini, yaitu pH dan suhu. Persyaratan kualitas air minum antara lain mempunyai kadar pH antara 6,5 – 8,5 (merupakan batas minimum dan maksimum) dan perbedaan antara suhu air dan suhu alam disekitarnya yang diperbolehkan adalah sebesar $\pm 3^{\circ}\text{C}$. Contoh: suhu alam = 25°C . berarti suhu air yang diperbolehkan berkisar $22\text{--}28^{\circ}\text{C}$. Jika melebihi atau kurang dari kisaran tersebut, bisa dikatakan air tersebut telah tercemar. Suhu memegang peranan penting dalam berbagai aktivitas kimia dan fisika perairan. Aktivitas kimia dan fisika seringkali mengalami peningkatan dengan naiknya suhu. Mahida (1986) menyatakan bahwa tingkat oksidasi senyawa organik jauh lebih besar pada suhu tinggi dibanding pada suhu rendah. Semakin tinggi kenaikan suhu air semakin sedikit oksigen yang terlarut didalamnya.

Wireless sensor network (WSN) merupakan jaringan nirkabel yang terdiri dari beberapa alat sensor yang saling bekerja sama untuk memonitor fisik dan kondisi lingkungan seperti *temperature*, air, suara, getaran atau gempa, polusi udara dan lain-lain ditempat yang berbeda. Perkembangan *wireless sensor* pada awalnya digunakan oleh pihak militer sebagai aplikasi untuk keperluan pengawasan. Namun saat ini banyak digunakan oleh masyarakat umum antara lain untuk aplikasi lingkungan, memonitor tempat tinggal dan digunakan untuk aplikasi kesehatan (Faludi, 2010).

Namun, terdapat kekurangan dalam WSN yaitu WSN memiliki energi dan *bandwidth* yang terbatas. Salah satu cara agar energi dan komunikasi WSN dapat lebih efisien adalah dengan mengatur topologi jaringannya dengan menyesuaikan

dengan kondisi lingkungannya. Dalam WSN, topologi yang dapat digunakan adalah topologi *star, mesh, dan tree* (Y.K. Huang dkk,2012).

Topologi *MESH* merupakan jalur komunikasi dimana masing-masing node dapat berkomunikasi dengan yang lainnya. Dalam sebuah jaringan mesh, node mempertahankan jalur komunikasi untuk kembali ke *gateway*, sehingga jika salah satu node router *down*, secara otomatis router data akan dilewatkan melalui jalur yang berbeda.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan yaitu bagaimana merancang bangun jaringan WSN sebagai perangkat pendukung untuk mempermudah monitoring pencemaran air sungai menggunakan topologi *MESH network*.

1.3 Batasan Masalah

Batasan masalah dari pembahasan tugas akhir ini adalah :

1. Menggunakan modul mikrokontroller Arduino Uno.
2. Mengukur kadar pH menggunakan sensor Analog pH Meter Kit.
3. Mengukur suhu menggunakan sensor suhu DS18B20.
4. Menggunakan modul *Real Time Clock* (RTC) sebagai pencatatan waktu.
5. Menggunakan modul Xbee *series 2* sebagai komunikasi *wireless* dan pengiriman data dengan jarak maksimum 100 meter.
6. Menggunakan *software* arduino IDE untuk memprogram mikrokontroler pada arduino uno.
7. Pengujian dilakukan tidak pada kondisi lingkungan yang sebenarnya, tetapi pengujian akan dilakukan menggunakan *prototype*.

8. Perancangan dibatasi sampai pada pembuatan jaringan dan perancangan sistem monitoring pH dan suhu air.
9. Topologi yang digunakan dalam tugas akhir ini menggunakan topologi *MESH network*.

1.4 Tujuan

Dalam rancang bangun WSN untuk monitoring pencemaran air sungai ini, bertujuan untuk :

Merencanakan dan membangun jaringan WSN dengan perangkat pendukung *sensor node* untuk mempermudah monitoring tingkat pencemaran pada air sungai menggunakan topologi *MESH network*.

1.5 Sistematika Penulisan

Pembahasan Tugas Akhir ini secara Garis besar tersusun dari 5 (lima) bab, yaitu diuraikan sebagai berikut:

BAB I. PENDAHULUAN

Pada Bab ini akan dibahas mengenai latar belakang masalah, batasan masalah, tujuan penulisan, dan sistematika penulisan.

BAB II. LANDASAN TEORI

Pada Bab ini akan dibahas teori penunjang dari permasalahan, yaitu mengenai WSN, Arduino Uno, *software arduino IDE*, ZigBee (Xbee *series 2*), *software X-CTU*, sensor pH Meter Kit, sensor suhu DS18B20 dan modul RTC.

BAB III. METODE PENELITIAN DAN PERANCANGAN SISTEM

Pada Bab ini akan dibahas tentang blog diagram sistem WSN serta metode yang dilakukan dalam perancangan sistem monitoring pencemaran air sungai. Meliputi diagram alur proses pengukuran kadar pH, pengukuran suhu air, pembacaan tanggal serta waktu, pengiriman atau pembacaan data dan skrip *software* arduino IDE untuk pembacaan sensor Analog pH Meter Kit, sensor suhu DS18B20, modul RTC dan modul mikrokontroller arduino uno untuk mengukur kadar pH, suhu air serta mencatat waktu saat kejadian. Kemudian ketiga data tersebut dikirimkan melalui wireless (Xbee *series* 2), dan konfigurasi Xbee *series* 2 dalam mode AT pada *software* X-CTU dan *flow cart software* visual basic untuk menampilkan data yang diterima oleh arduino.

BAB IV. HASIL DAN PEMBAHASAN

Pada Bab ini akan dibahas mengenai pengujian pengukuran kadar pH pada prototipe dengan sensor Analog pH Meter Kit dan mengukur suhu air menggunakan sensor suhu DS18B20 dari *node* sensor 1 dan *node* sensor 2 bersama waktu yang dicatat oleh RTC. Ketiga data tersebut kemudian akan dikirimkan ke *node coordinator*, setelah dari *node coordinator* data dikirim ke *node end device*. Data tersebut kemudian akan dikirimkan ke computer dan ditampilkan oleh *software* *visual basic*.

BAB V. KESIMPULAN DAN SARAN

Berisi kesimpulan yang didapat dari hasil penelitian berdasarkan rumusan masalah serta saran untuk perkembangan penelitian selanjutnya.

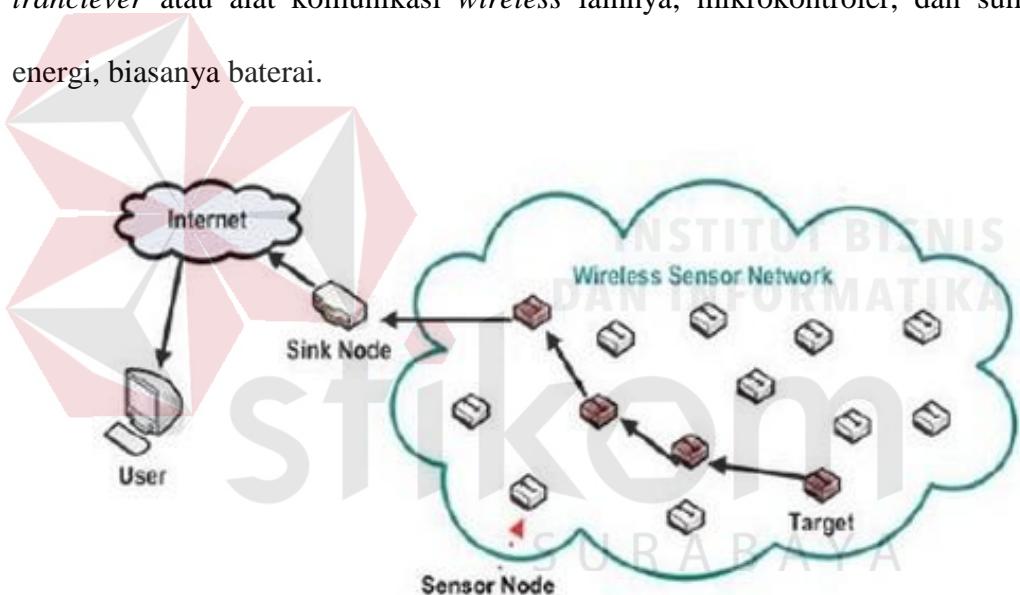


BAB II

LANDASAN TEORI

2.1 Wireless sensor network(WSN)

Wireless Sensor Network (WSN) atau jaringan sensor nirkabel adalah suatu infrastruktur jaringan *wireless* yang menggunakan sensor untuk memonitor kondisi fisik atau kondisi lingkungan yang dapat terhubung ke jaringan. Masing-masing *node* dalam jaringan sensor nirkabel biasanya dilengkapi dengan radio *transceiver* atau alat komunikasi *wireless* lainnya, mikrokontroler, dan sumber energi, biasanya baterai.

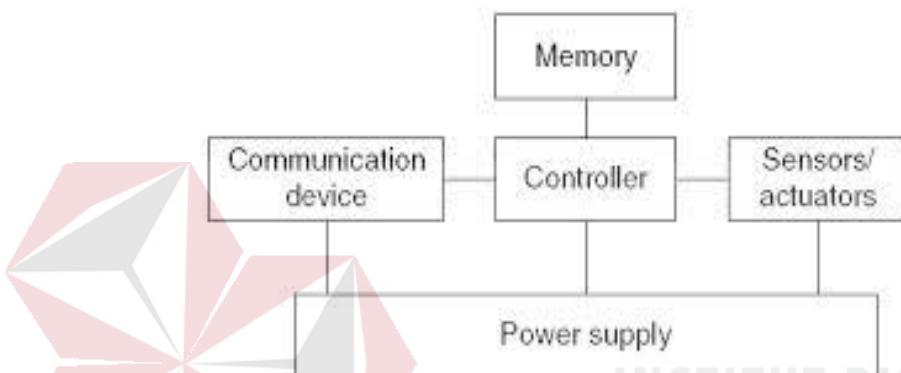


Gambar 2.1 Arsitektur WSN (digilib.ittelkom.ac.id)

Pada Gambar 2.1 menunjukkan gambaran umum WSN, dapat dilihat *node* sensor yang berukuran kecil disebar dalam di suatu area sensor. *Node* sensor tersebut memiliki kemampuan untuk merutekan data yang dikumpulkan ke *node* lain yang berdekatan. Data dikirimkan melalui transmisi radio akan diteruskan menuju BS (*Base Station*) atau *sink node* yang merupakan penghubung antara *node* sensor dan *user*. Informasi tersebut dapat diakses melalui berbagai *platform*

seperti koneksi internet atau satelit sehingga memungkinkan *user* untuk dapat mengakses secara *realtime* melalui *remote server*. (E, Sugiarto, & Sakti, 2009)

Setiap *node* dalam WSN terdiri dari lima komponen yaitu kontroller/mikrokontroler, memori, sensor/akuator, perangkat komunikasi dan catu daya. Umumnya catu daya yang dipakai adalah baterai. Komponen – komponen dari sebuah *node* ditunjukkan pada Gambar 2.2 :



Gambar 2.2 Komponen – Komponen Penyusun *Node* dalam WSN

1. *Communication device* (perangkat komunikasi)

Berfungsi untuk menerima/mengirim data dengan menggunakan protokol IEEE 802.15.4 atau IEEE 802.11b/g kepada *device* atau *node* lainnya.

2. Mikrokontroler

Berfungsi untuk melakukan fungsi perhitungan, mengontrol dan memproses *device – device* yang terhubung dengan mikrokontroler.

3. Sensor

Berfungsi untuk men-sensing besaran-besaran fisis yang hendak diukur. Sensor adalah suatu alat yang mampu untuk mengubah suatu bentuk energi ke bentuk energi lain, dalam hal ini adalah mengubah dari energi besaran yang

diukur menjadi energi listrik yang kemudian diubah oleh ADC menjadi deretan pulsa terkuantisasi yang kemudian bisa dibaca oleh mikrokontroler.

4. *Memory*

Berfungsi sebagai tambahan memori bagi sistem *Wireless Sensor*, pada dasarnya sebuah unit mikrokontroler memiliki unit memori sendiri.

5. *Power supply*

Berfungsi sebagai sumber energi bagi sistem *Wireless Sensor* secara keseluruhan. Dengan adanya teknologi WSN, memungkinkan peneliti untuk mendapat informasi yang maksimal tanpa harus berada di area sensor. Informasi dapat diakses dari jarak jauh melalui *gadget* seperti laptop, *remote control*, *server* dan sebagainya.

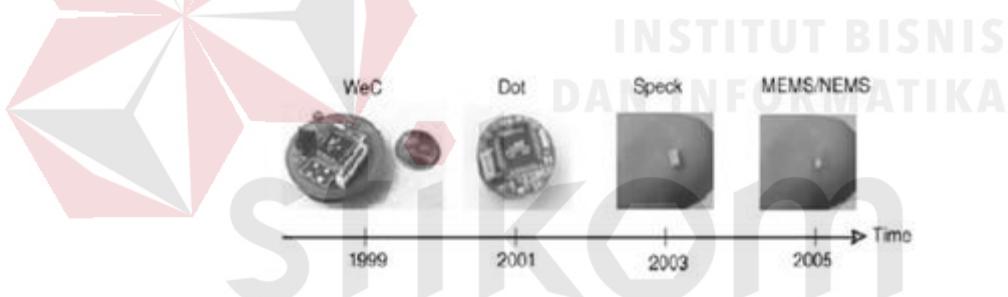
Beberapa keuntungan yang bisa diperoleh dari teknologi WSN adalah Simpel / praktis / ringkas karena tidak perlu ada instalasi kabel yang rumit dan dalam kondisi geografi tertentu sangat menguntungkan dibanding *Wired Sensor*. Sensor menjadi bersifat *mobile*, artinya pada suatu saat dimungkinkan untuk memindahkan sensor untuk mendapat pengukuran yang lebih tepat tanpa harus khawatir mengubah desain ruangan maupun susunan kabel ruangan.

1. Meningkatkan efisiensi secara operasional.
2. Mengurangi total biaya sistem secara signifikan.
3. Dapat mengumpulkan data dalam jumlah besar.
4. Konfigurasi *software* mudah.
5. Memungkinkan komunikasi digital 2 arah.
6. Menyediakan konektivitas internet yang secara global, kapanpun dimanapun informasi tersebut dapat diakses melalui *server*, laptop dan sebagainya.

2.1.1 Arsitektur WSN

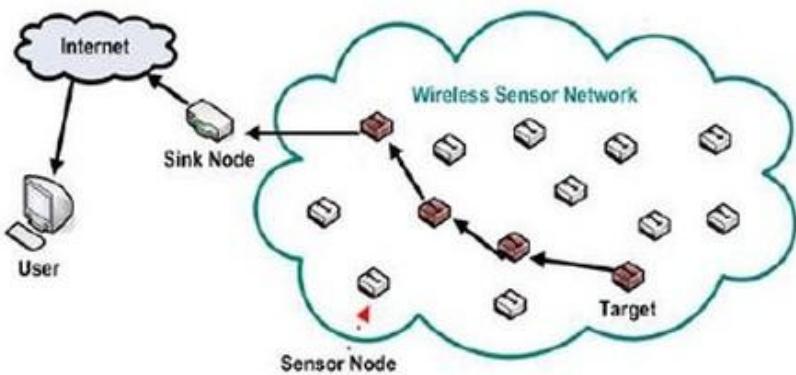
Pada WSN, *node* sensor disebar dengan tujuan untuk menangkap adanya gejala atau fenomena yang hendak diteliti. Jumlah *node* yang disebar dapat ditentukan sesuai kebutuhan dan tergantung beberapa faktor misalnya luas area, kemampuan *sensing node*, dan sebagainya. Tiap *node* memiliki kemampuan untuk mengumpulkan data dan meroutingkannya kembali ke *Base Station*. *Node* sensor dapat mengumpulkan data dalam jumlah yang besar dari gejala yang timbul dari lingkungan sekitar.

Perkembangan *node* sensor mengikuti trend teknologi nano, dimana ukuran *node* sensor menjadi semakin kecil dari tahun ke tahun. *Node* sensor dapat direpresentasikan dalam Gambar 2.3 :



Gambar 2.3 Perkembangan Dimensi *Node* Sensor Terhadap Waktu.

Dan untuk arsitektur WSN secara umum dapat direpresentasikan dalam Gambar 2.4 :



Gambar 2.4 Gambaran Arsitek Wireless Sensor Network (digilib.ittelkom.ac.id)

Pada Gambar 2.4 dapat dilihat, *node* sensor yang berukuran kecil disebar dalam di suatu area sensor. *Node* sensor tersebut memiliki kemampuan untuk merutekan data yang dikumpulkan ke *node* lain yang berdekatan. Data dikirimkan melalui transmisi radio akan diteruskan menuju BS (*Base Station*) yang merupakan penghubung antara *node* sensor dan *user*. Informasi tersebut dapat diakses melalui berbagai *platform* seperti koneksi internet atau satelit sehingga memungkinkan *user* untuk dapat mengakses secara realtime melalui *remote server* (Rolis, 2012).

2.2 Arduino

Arduino adalah prototipe platform elektronik opensource yang terdiri dari mikrokontroler, bahasa pemrograman, dan IDE. Arduino adalah alat untuk membuat aplikasi interaktif, yang dirancang untuk mempermudah proyek bagi

pemula tetapi masih cukup fleksibel bagi para ahli untuk mengembangkan proyek-proyek yang kompleks. (Banzi, 2009)

2.2.1 Arduino Uno SMD R3

Arduino Uno adalah papan mikrokontroler berbasis ATmega328. Dalam bahasa Italy “Uno” berarti satu, maka peluncuran arduino ini diberi nama Uno. Arduino ini berisi semua yang diperlukan untuk mendukung mikrokontroler, untuk mengaktifkan cukup menghubungkannya ke komputer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai. (arduino.cc, 2013)



Gambar 2.5 Arduino Uno SMD R3 Sisi Depan (Kiri) dan Belakang(Kanan)

Secara umum arduino terdiri dari dua bagian, yaitu:

1. *Hardware*: papan input/output (I/O)
2. *Software*: software arduino meliputi IDE untuk menulis program, driver untuk koneksi dengan komputer, contoh program dan *library* untuk pengembangan program. (Djuandi, 2011)

Berikut adalah Tabel 2.1 spesifikasi dari arduino uno smd R3:

Tabel 2.1 Spesifikasi Arduino Uno SMD R3

Mikrokontroler	ATmega328
Tegangan pengoperasian	5V
Tegangan input yang disarankan	7-12V
Batas tegangan input	6-20V
Jumlah pin I/O digital	14 (6 di antaranya menyediakan keluaran PWM)
Jumlah pin input analog	6
Arus DC tiap pin I/O	40 mA
Arus DC untuk pin 3.3V	50 mA
Memori Flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

2.2.2 Daya (*Power*)

Arduino Uno dapat disuplai melalui koneksi USB atau dengan sebuah power suplai eksternal. Suplai eksternal (non-USB) dapat diperoleh dari sebuah adaptor AC ke DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan sebuah *center-positive* plug yang panjangnya 2,1 mm ke power jack dari *board*. Kabel lead dari sebuah battery dapat dimasukkan dalam *header*/kepala pin *Ground* (Gnd) dan pin Vin dari konektor *POWER*.

Board Arduino Uno dapat beroperasi pada sebuah suplai eksternal 6 sampai 20 Volt. Jika disuplai dengan yang lebih kecil dari 7 V, kiranya pin 5 Volt mungkin mensuplai kecil dari 5 Volt dan *board* Arduino Uno bisa menjadi tidak stabil. Jika menggunakan suplai yang lebih dari besar 12 Volt, *voltage* regulator bisa kelebihan panas dan membahayakan *board* Arduino Uno. *Range* yang direkomendasikan adalah 7 sampai 12 Volt. (arduino.cc, 2013)

Pin-pin dayanya adalah sebagai berikut:

- a) VIN. Tegangan input ke Arduino *board* ketika *board* sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur). Kita dapat menyuplai tegangan melalui pin ini, atau jika penyuplaiannya tegangan melalui *power jack*, aksesnya melalui pin ini.
- b) 5V. Pin output ini merupakan tegangan 5 Volt yang diatur dari regulator pada *board*. *Board* dapat disuplai dengan salah satu suplai dari DC *power jack* (7-12V), USB *connector* (5V), atau pin VIN dari *board* (7-12). Penyuplaiannya tegangan melalui pin 5V atau 3,3V membypass regulator, dan dapat membahayakan *board*. Hal itu tidak dianjurkan.
- c) 3V3. Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada *board*. Arus maksimum yang dapat dilalui adalah 50 mA.
- d) GND. Pin *ground*.

2.2.3 Memori

ATmega328 mempunyai 32 KB yang bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. (dengan 0,5 KB digunakan untuk *bootloader*). ATmega 328 juga mempunyai 2 KB SRAM yang *volatile* (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program. dan 1 KB EEPROM (yang dapat dibaca dan ditulis (RW/read and written). (arduino.cc, 2013)

2.2.4 Input dan Output

Setiap 14 pin digital pada Arduino Uno dapat digunakan sebagai input dan output. Fungsi-fungsi tersebut beroperasi di tegangan 5 Volt. Setiap pin dapat memberikan atau menerima suatu arus maksimum 40 mA dan mempunyai sebuah

resistor pull-up (terputus secara default) 20-50 kOhm. Selain itu, beberapa pin mempunyai fungsi-fungsi sebagai berikut:

- a) **Serial: 0 (RX) dan 1 (TX).** Digunakan untuk menerima (RX) dan memancarkan (TX) serial data TTL (*Transistor-Transistor Logic*). Kedua pin ini dihubungkan ke pin-pin yang sesuai dari chip Serial Atmega8U2 USB-ke-TTL.
- b) **External Interrupts: 2 dan 3.** Pin-pin ini dapat dikonfigurasikan untuk dipicu sebuah interrupt (gangguan) pada suatu nilai rendah, suatu kenaikan atau penurunan yang besar, atau suatu perubahan nilai.
- c) **PWM: 3, 5, 6, 9, 10, dan 11.** Memberikan 8-bit PWM *output* dengan fungsi `analogWrite()`.
- d) **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** Pin-pin ini mensupport komunikasi SPI menggunakan `SPI library`.
- e) **LED: 13.** Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai *HIGH* LED menyala, ketika pin bernilai *LOW* LED mati.

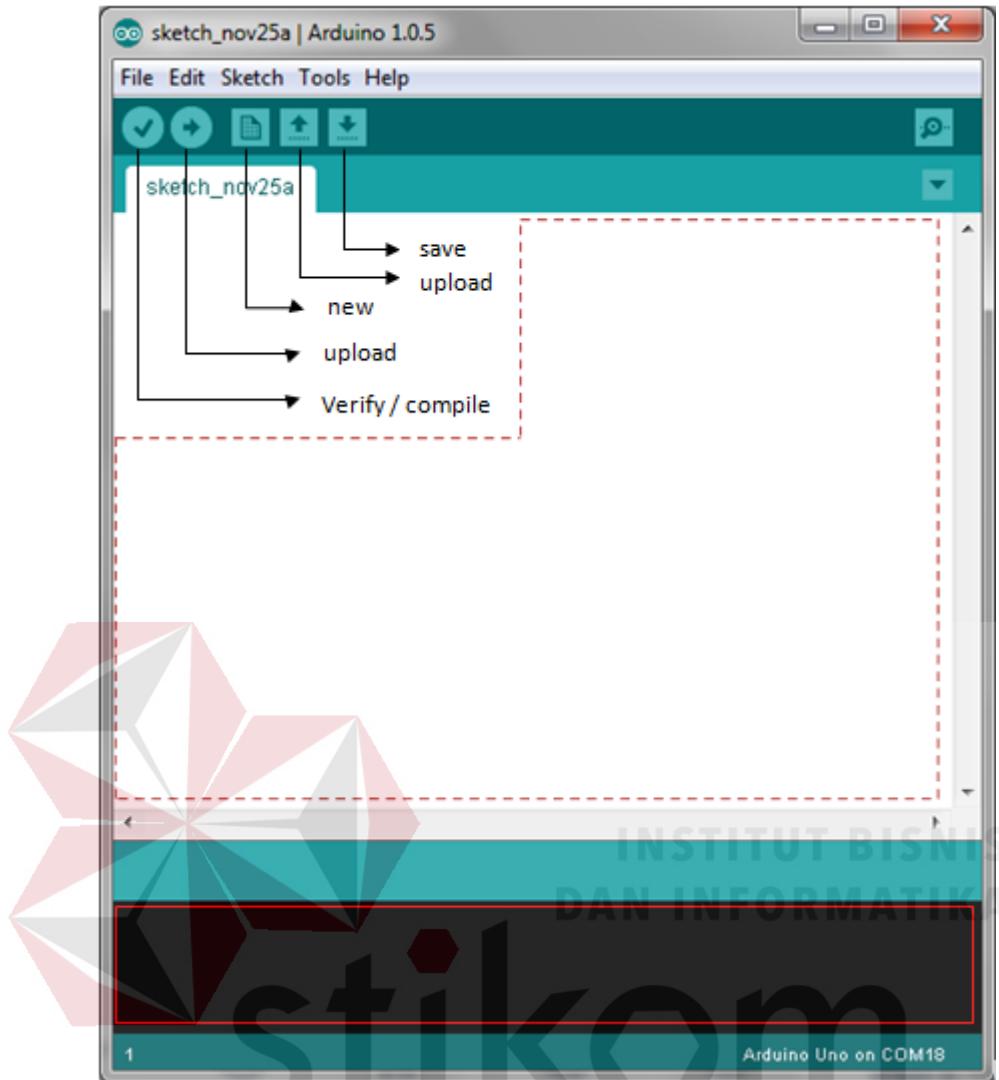
Arduino UNO mempunyai 6 input analog, diberi label A0 sampai A5, setiapnya memberikan resolusi 10 bit. Secara default, 6 input analog tersebut mengukur tegangan dari *ground* sampai tegangan 5 Volt, dengan itu memungkinkan untuk mengganti batas atas dari rangenya dengan menggunakan pin AREF dan fungsi `analogReference()`. Di sisi lainnya, beberapa pin mempunyai fungsi spesifik yaitu pin A4 atau SDA dan pin A5 atau SCL. Mendukung komunikasi TWI dengan menggunakan `Wire library`. Ada sepasang pin lainnya pada board yaitu AREF referensi tegangan untuk input analog.

Digunakan dengan *analogReference()*, dan reset untuk mereset mikrokontroler.
(arduino.cc, 2013)

2.3 Software Arduino IDE

Arduino IDE adalah *software* yang ditulis menggunakan java dan berdasarkan pengolahan seperti, avr-gcc, dan perangkat lunak *open source* lainnya (Djuandi, 2011). Arduino IDE terdiri dari:

1. *Editor program*, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa processing.
2. *Verify / Compiler*, sebuah modul yang mengubah kode program (bahasa processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroller tidak akan bisa memahami bahasa processing, yang dipahami oleh mikrokontroller adalah kode biner.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori mikrokontroller di dalam papan arduino.



Gambar 2.6 Tampilan *Software Arduino IDE*

Pada Gambar 2.6 terdapat *menu bar*, kemudian *toolbar* dibawahnya, dan sebuah area putih untuk *editing sketch*, area hitam dapat kita sebut sebagai *progress area*, dan paling bawah dapat kita sebut sebagai “*status bar*”.

2.4 Bahasa Pemograman Arduino

Arduino ini bisa dijalankan di komputer dengan berbagai macam *platform* karena didukung atau berbasis Java. *Source program* yang dibuat untuk

aplikasi mikrokontroler adalah bahasa C/C++ dan dapat digabungkan dengan assembly. (arduino.cc, 2013)

1. Struktur

Setiap program Arduino (biasa disebut *sketch*) mempunyai dua buah fungsi yang harus ada (arduino.cc, 2013). Antara lain:

a) **void setup() { }**

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

b) **void loop() { }**

Fungsi ini akan dijalankan setelah *setup* (fungsi *void setup*) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (*power*) dilepaskan.

2. Serial

Serial digunakan untuk komunikasi antara arduino *board*, komputer atau perangkat lainnya. Arduino *board* memiliki minimal satu *port* serial yang berkomunikasi melalui pin 0 (RX) dan 1 (TX) serta dengan komputer melalui USB. Jika menggunakan fungsi – fungsi ini, pin 0 dan 1 tidak dapat digunakan untuk *input* digital atau *output* digital (arduino.cc, 2013). Terdapat beberapa fungsi serial pada arduino, antara lain:

a) *Serial.begin ()*

Fungsi ini digunakan untuk transmisi data serial dan mengatur data *rate* dalam *bits per second (baud)*. Untuk berkomunikasi dengan komputer gunakan salah satu dari angka ini: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, atau 115200.

b) `Serial.available()`

Fungsi ini digunakan untuk mendapatkan jumlah data *byte (characters)* yang tersedia dan membacanya dari *port serial*. Data tersebut adalah data yang telah tiba dan disimpan dalam *buffer serial* yang menampung sampai 64 *bytes*.

c) `Serial.read()`

Fungsi digunakan untuk membaca data serial yang masuk.

d) `Serial.print()` dan `Serial.println()`

Fungsi ini digunakan untuk mencetak data ke *port serial* dalam format text ASCII. Sedangkan fungsi `Serial.println()` sama seperti fungsi `Serial.print()` hanya saja ketika menggunakan fungsi ini akan mencetak data dan kemudian diikuti dengan karakter *newline* atau *enter*.

3. *Syntax*

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan. (arduino.cc, 2013)

a) `//`(komentar satu baris)

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

b) `/* */`(komentar banyak baris)

Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

c) `{ }`(kurung kurawal)

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

- d) ;(titik koma)

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

4. Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya. (arduino.cc, 2013)

- a) **int** (integer)

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32,768 dan 32,767.

- b) **long** (long)

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.

- c) **boolean** (boolean)

Variabel sederhana yang digunakan untuk menyimpan nilai *TRUE* (benar) atau *FALSE* (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

- d) **float** (float)

Digunakan untuk angka desimal (*floating point*). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari -3.4028235E+38 dan 3.4028235E+38.

- e) **char** (character)

Menyimpan 1 karakter menggunakan kode ASCII (misalnya ‘A’ = 65). Hanya memakai 1 *byte* (8 bit) dari RAM.

5. Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana). (arduino.cc, 2013)

- a) = (sama dengan)

Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya: $x = 10 * 2$, x sekarang sama dengan 20).

- b) % (persen)

Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya: $12 \% 10$, ini akan menghasilkan angka 2).

- c) + (penjumlahan)
- d) - (pengurangan)
- e) * (perkalian)
- f) / (pembagian)

6. Operator Pembanding

Digunakan untuk membandingkan nilai logika.

- a) ==

Sama dengan (misalnya: $12 == 10$ adalah *FALSE* (salah) atau $12 == 12$ adalah *TRUE* (benar)).

- b) !=

Tidak sama dengan (misalnya: $12 != 10$ adalah *TRUE* (benar) atau $12 != 12$ adalah *FALSE* (salah)).

- c) <

Lebih kecil dari (misalnya: $12 < 10$ adalah *FALSE* (salah) atau $12 < 12$ adalah *FALSE* (salah) atau $12 < 14$ adalah *TRUE* (benar)).

d) $>$

Lebih besar dari (misalnya: $12 > 10$ adalah *TRUE* (benar) atau $12 > 12$ adalah *FALSE* (salah) atau $12 > 14$ adalah *FALSE* (salah)).

7. Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (banyak lagi yang lain dan bisa dicari di internet). (arduino.cc, 2013)

a) **If else**, dengan format seperti berikut ini:

```
if (kondisi) { }
```

```
else if (kondisi) { }
```

```
else { }
```

Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya *TRUE*, dan jika tidak (*FALSE*) maka akan diperiksa apakah kondisi pada *else if* dan jika kondisinya *FALSE* maka kode pada *else* yang akan dijalankan.

b) **While**, dengan format seperti berikut ini:

```
While(kondisi) {}
```

Dengan struktur ini, while akan melakukan pengulangan terus menurus dan tak terbatas sampai kondisi didalam kurung () menjadi false.

c) **for**, dengan format seperti berikut ini:

```
for (int i = 0; i < #pengulangan; i++) { }
```

Digunakan bila ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan *i++* atau ke bawah dengan *i--*.

8. Operator Boolean

Operator ini dapat digunakan dalam kondisi if, antara lain:

- a) **&&** (logika *and*), dengan format seperti berikut ini:

```
if (digitalRead(2) == HIGH && digitalRead(3) == HIGH) {}
```

Digunakan bila ingin mendapatkan nilai yang *true* hanya jika kedua input bernilai *HIGH*.

- b) **||** (logika *or*), dengan format seperti berikut ini:

```
if (x > 0 || y > 0) {}
```

Digunakan bila ingin mendapatkan nilai yang *true* hanya jika nilai x dan y lebih besar dari 0.

- c) **!** (*not*), dengan format seperti berikut ini:

```
if (!x) {}
```

Digunakan bila ingin mendapatkan nilai yang *true* hanya jika nilai tidak sama dengan x.

9. Digital

- a) **pinMode(pin, mode)**

Digunakan untuk menetapkan mode dari suatu pin, *pin* adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah *INPUT* atau *OUTPUT*.

- b) **digitalWrite(pin, value)**

Ketika sebuah pin ditetapkan sebagai *OUTPUT*, pin tersebut dapat dijadikan *HIGH* (5 volts) atau *LOW* (diturunkan menjadi *ground*).

c) **`digitalRead(pin)`**

Ketika sebuah pin ditetapkan sebagai *INPUT* maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah *HIGH* (5 volts) atau *LOW* (diturunkan menjadi *ground*).

10. Analog

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam analog. Berikut ini cara untuk menghadapi hal yang bukan digital.

a) **`analogWrite(pin, value)`**

Beberapa pin pada Arduino mendukung PWM (*pulse width modulation*) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (*on*) atau mati (*off*) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. *Value* (nilai) pada format kode tersebut adalah angka antara 0 (0% *duty cycle* ~ 0V) dan 255 (100% *duty cycle* ~ 5V).

b) **`analogRead(pin)`**

Ketika pin analog ditetapkan sebagai *INPUT* anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts).

2.5 Sensor Analog pH Meter Kit

Sensor pH adalah sensor yang dapat mendekripsi kadar pH air. Sensor ini sangat membantu mengingatkan tingkat kadar pH pada air atau untuk memantau kadar pH air untuk pencemarain air. Secara fisik, sensor ini terdiri dari LED

sebagai power indikator, konektor BNC, dan interface sensor PH2.0. Untuk menggunakan, cukup hubungkan sensor pH ini dengan Arduino menggunakan kabel analog yang disertakan dalam kit ini ke IO Expansion Shield atau bisa pula menggunakan kabel Jumper. *IO Expansion Shield* adalah *shield* untuk menghubungkan sensor dengan Arduino. (Moisture, 2013)



Gambar 2.7 Sensor Analog pH Meter Kit

Spesifikasi:

- a. Module Power : 5.00V
- b. Module Size : 43mmx32mm
- c. Measuring Range :0-14PH
- d. Measuring Temperature :0-60 derajat C
- e. Accuracy : $\pm 0.1\text{pH}$ (25 derajat C)
- f. Response Time : < 1min
- g. pH Sensor with BNC Connector
- h. pH2.0 Interface (3 foot patch)
- i. Gain Adjustment Potentiometer
- j. Power Indicator LED

2.6 *Digital Temperature DS18B20 (WATERPROOF)*

Produk ini merupakan sensor suhu DS18B20 dengan kemampuan tahan air (*waterproof*). Cocok digunakan untuk mengukur suhu pada tempat yang sulit, atau basah. Karena output data produk ini merupakan data digital, maka Anda tidak perlu khawatir terhadap degradasi data ketika menggunakan untuk jarak yang jauh. DS18B20 menyediakan 9 hingga 12-bit (yang dapat dikonfigurasi) data (dfrobot.com).

Karena setiap sensor DS18B20 memiliki silicon serial number yang unik, maka beberapa sensor DS18B20 dapat dipasang dalam 1 bus. Hal ini memungkinkan pembacaan suhu dari berbagai tempat. Meskipun secara datasheet sensor ini dapat membaca bagus hingga 125°C, namun dengan penutup kabel dari PVC disarankan untuk penggunaan tidak melebihi 100°C.



Gambar 2.8 Digital Temperature DS18B20

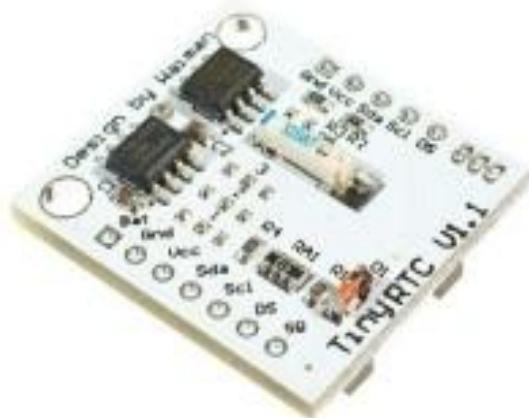
Spesifikasi :

- a. Usable with 3.0V to 5.5V power/data
- b. $\pm 0.5^\circ\text{C}$ Accuracy from -10°C to $+85^\circ\text{C}$

- c. Usable temperature range: -55 to 125°C (-67°F to +257°F)
- d. 9 to 12 bit selectable resolution
- e. Uses 1-Wire interface- requires only one digital pin for communication
- f. Unique 64 bit ID burned into chip
- g. Multiple sensors can share one pin
- h. Temperature-limit alarm system
- i. Query time is less than 750ms
- j. 3 wires interface:
 - 1. Red wire - VCC.
 - 2. Black wire – GND.
 - 3. Yellow wire – DATA.
- k. Stainless steel tube 6 mm diameter by 35 mm long.
- l. Cable diameter: 4 mm.
- m. Length: 90 cm.

2.7 Real Time Clock (RTC)

RTC adalah jenis pewaktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam kita. Agar dapat berfungsi, pewaktu ini membutuhkan dua parameter utama yang harus ditentukan, yaitu pada saat mulai (*start*) dan pada saat berhenti (*stop*). Real Time Clock Module 02 DS1307, merupakan produk RTC keluaran DFRobot yang di desain oleh waiman. Produk ini dijual beserta baterai kancing CR2032), yang dapat mengoperasikan module ini hingga 9 tahun (secara teoritis 17 tahun) tanpa bantuan external power 5 volt. RTC DS1307 ini diakses melalui protokol I2C (dfrobot.com).



Gambar 2.9 RTC

Spesifikasi:

- a. Two wire I2C interface
- b. Hour : Minutes : Seconds AM/PM
- c. Day Month, Date – Year
- d. Leap year compensation.
- e. Accurate calendar up to year 2100.
- f. Battery backup included.
- g. 1Hz output pin.
- h. 56 Bytes of Non-volatile memory available to user.

2.8 Xbee Series 2 Chip Antenna dan Xbee Pro Series 2 Wire Antenna

Xbee series 2 modul RF dirancang untuk beroperasi dalam protokol ZigBee dengan biaya yang murah dan jaringan sensor nirkabel menggunakan daya yang rendah. Modul ini membutuhkan daya yang rendah dan dapat melakukan pengiriman data yang handal antara perangkat dengan jarak yang jauh. Modul ini beroperasi pada frekuensi 2.4 GHz. (Inc, XBee Series 2 OEM RF Modules, 2007)

Xbee *series 2* ini mempunyai beberapa model antena, salah duanya adalah *chip antenna* dan *wire antenna*. *Chip antenna* merupakan suatu chip keramik yang terletak pada *board* modul Xbee, bentuknya lebih kecil. *Chip antenna* memiliki pola radiasi cardoid, yang artinya sinyal dilemahkan dalam berbagai arah dan sangat baik digunakan dalam area yang tidak terlalu besar atau kecil. Sedangkan *wire antenna* merupakan suatu antena kawat yang terletak pada board modul Xbee, *wire antenna* memiliki pola radiasi *omnidirectional* yang artinya jarak transmisi maksimum hampir sama pada semua arah ketika antena tersebut tegak lurus terhadap modul. Gambar 2.10 merupakan gambar dari modul Xbee *series 2 chip antenna* dan Gambar 2.11 merupakan gambar dari modul Xbee *series 2 wire antenna*. (Faludi, 2011)



Gambar 2.10 Xbee Series 2 *Chip Antenna*



Gambar 2.11 Xbee Series 2 *Wire Antenna*

Berikut adalah spesifikasi dari modul Xbee pro *series 2 chip antenna* (Inc, Xbee Series 2 OEM RF Modules, 2007):

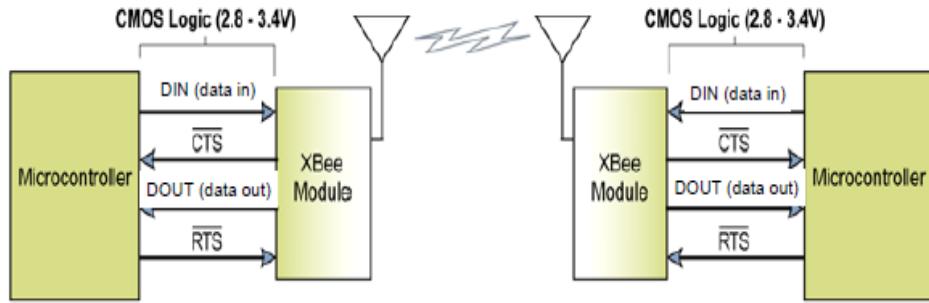
- a) Jarak jangkauan *indoor* 133 ft atau 40 meter.
- b) Jarak jangkauan *outdoor line of sight* 400 ft atau 120 meter.
- c) Transmit *power output* 2 mW (+ 3 dbm).
- d) Radio Frekuensi data *rate* 250 Kbps.
- e) Frekuensi 2.4 GHz.
- f) *Receiver sensitivity* -98 dbm (1 % *packet error rate*).
- g) Antena menggunakan *chip antenna*.

Berikut adalah spesifikasi dari modul Xbee pro *series 2 wire antenna* (Inc, 2012):

- a) Jarak jangkauan *indoor* 300 ft (90 meter) dan 200 ft (60 meter)
- b) Jarak jangkauan *outdoor line of sight* 2 miles atau 3200 meter dan 5000 ft atau 1500 meter (variant lainnya).
- c) Transmit *power output* 50 mW (+ 17 dbm).
- d) Radio Frekuensi data *rate* 250 Kbps.
- e) Frekuensi 2.4 GHz.
- f) *Receiver sensitivity* -102 dbm.
- g) Antena menggunakan *wire antenna*.

2.9 Komunikasi Serial Xbee Series 2

Xbee series 2 merupakan sebuah modul yang terdiri dari receiver dan transmitter melalui port serial. Melalui port serial ini Xbee dapat berkomunikasi secara UART (*Universal Asynchronous Reciever transmpter*). Gambar 2.12 menunjukkan diagram sistem aliran data secara UART. (Inc, 2007)



Gambar 2.12 Diagram Sistem Aliran Data UART pada Xbee

2.10 AT / Transparent Mode

Dalam mode transparent/AT, modul Xbee bertindak sebagai pengganti *serial line*. Semua data UART (*Universal Asynchronous Reciever transmiter*) diterima melalui pin *input* akan ditransmisikan. Ketika data tersebut diterima maka data akan dikirimkan keluar (Xbee lainnya) melalui pin *output*. Data atau paket yang diterima bisa ditujukan ke satu sasaran (*point to point*) atau ke beberapa sasaran (*star/broadcast*). (Inc, 2007)

2.11 Coordinator, Router, dan End Device

Pada jaringan ZigBee terdapat tiga jenis *node*, yaitu:

1. Coordinator

Sebuah *node* yang berfungsi untuk membentuk suatu jaringan. *Coordinator* bertanggung jawab untuk membangun *operating channel* dan PAN (*Personal Area Network*) ID untuk suatu jaringan. *Coordinator* dapat membentuk suatu jaringan dengan mengijinkan *router* dan *end device* untuk bergabung dalam jaringan tersebut. Setelah jaringan terbentuk, fungsi *coordinator* seperti *router* (dapat berpartisipasi dalam *routing* paket dan menjadi sumber atau tujuan untuk paket data). (Inc, 2007)

2. Router

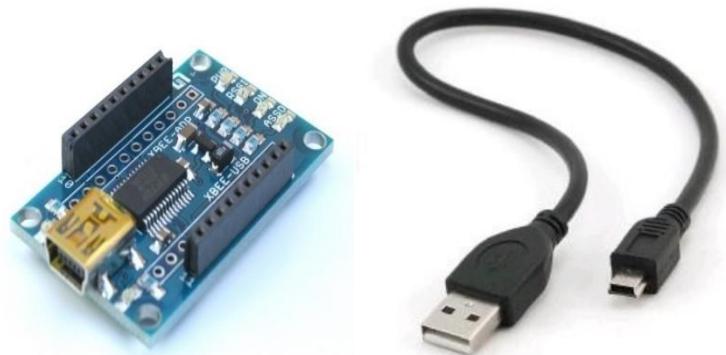
Sebuah *node* yang membuat / menjaga informasi jaringan dan menggunakan informasi ini untuk menentukan rute untuk paket data. Agar suatu *router* dapat mengijikan *router* lain dan *end device* untuk bergabung, sebelumnya *router* tersebut harus terlebih dahulu bergabung dengan jaringan yang ada. *Router* dapat berpartisipasi dalam *routing* paket dan menjadi sumber atau tujuan untuk paket data. (Inc, 2007)

3. End device

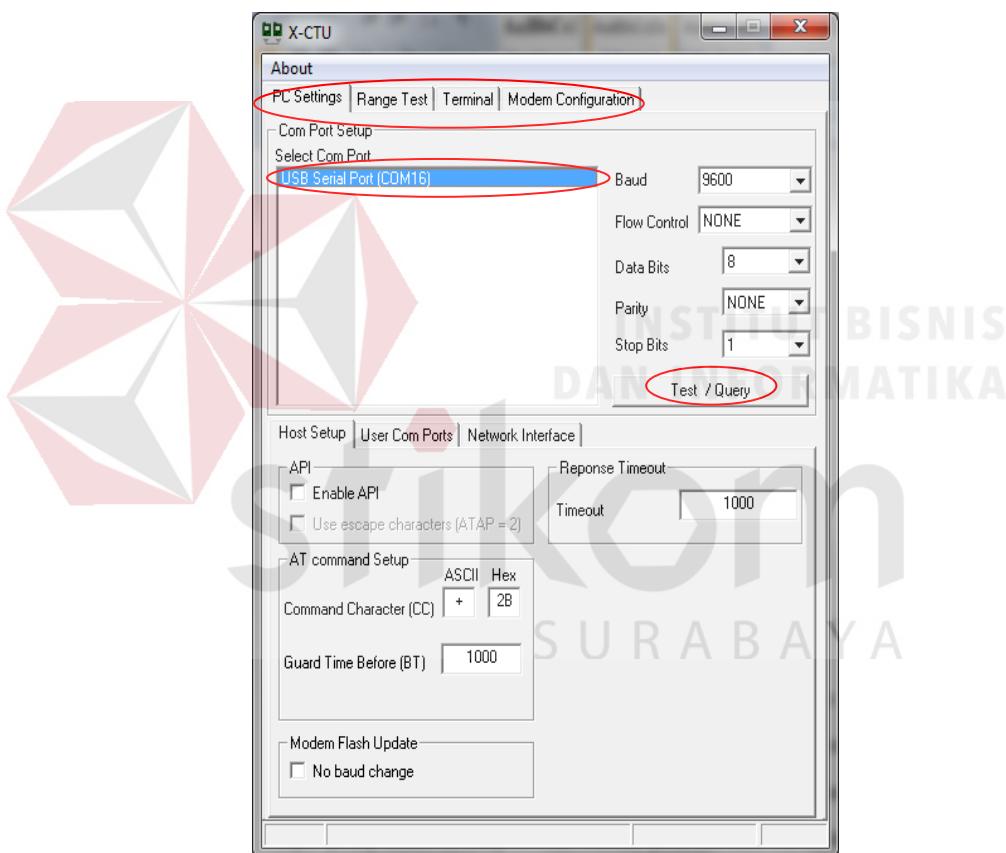
End device harus selalu berinteraksi atau terhubung dengan *coordinator* atau *router* untuk dapat menerima dan mengirimkan data. *End device* juga dapat menjadi sumber atau tujuan untuk paket data tetapi tidak bisa untuk menentukan rute paket data. (Inc, 2007)

2.12 Xbee Usb Adapter dan Software X-CTU

Xbee usb *adapter* (Gambar 2.13) merupakan alat untuk menghubungkan modul Xbee ke komputer dengan kabel mini usb (Gambar 2.13) dan selanjutnya dapat dikonfigurasi menggunakan *software* X-CTU (Gambar 2.14). *software* X-CTU merupakan *software* yang digunakan untuk mengkonfigurasi Xbee agar dapat berkomunikasi dengan Xbee lainnya. Parameter yang harus diatur adalah PAN ID (*Personal Area Network*) ID yaitu parameter yang mengatur radio mana saja yang dapat berkomunikasi, agar dapat berkomunikasi PAN ID dalam satu jaringan harus sama. Xbee dapat berkomunikasi point to point dan point to multipoint (broadcast).



Gambar 2.13 Xbee Usb Adapter dan Kabel Mini Usb

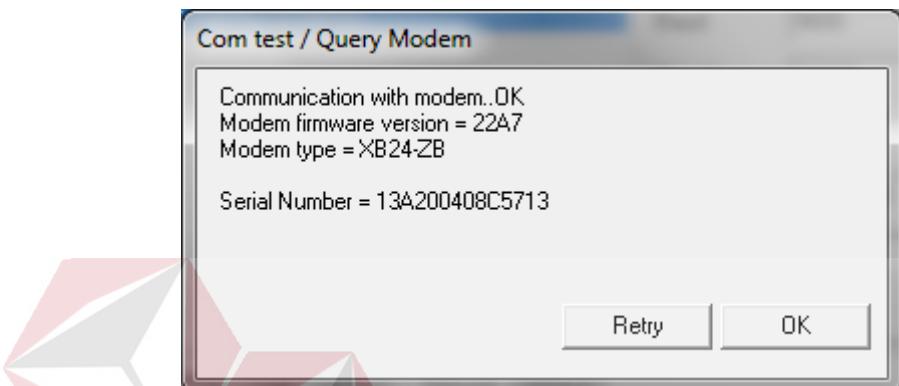


Gambar 2.14 Tampilan Software X-CTU

Pada Gambar 2.14 *software* X-CTU, terdapat empat tab di bagian atas program. Masing – masing tab mempunyai fungsi yang berbeda – beda (Inc, 2008). Berikut adalah ke empat tab tersebut:

1. *PC Settings*

Pada tab ini mengijinkan pengguna untuk memilih *COM port* yang diinginkan dan mengkonfigurasi *port* tersebut sesuai pengaturan Xbee yang diinginkan. Terdapat tombol *Test / Query* pada tab *PC Settings*, tombol ini digunakan untuk menguji *COM port* yang telah dipilih, jika pengaturan dan *COM port* benar, maka akan muncul kotak dialog respon seperti pada Gambar 2.15.



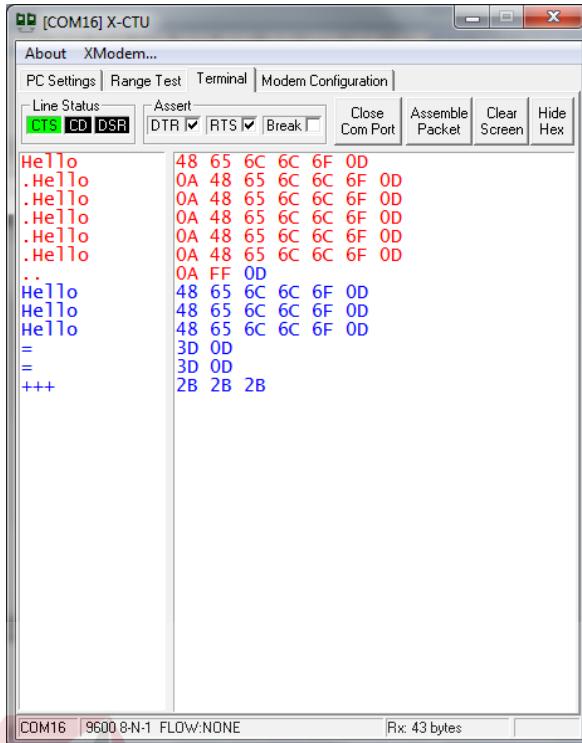
Gambar 2.15 Kotak Dialog Respon

2. Range Test

Pada tab *range test* ini, pengguna dapat melakukan pengujian *range test* antara dua Xbee dengan mengirimkan paket data yang ditentukan pengguna dan memverifikasi apakah paket data yang dikirim sama dengan yang diterima.

3. Terminal

Pada tab ini, pengguna memungkikan akses ke *COM port* komputer dengan program terminal *emulation*. Tab ini juga memungkinkan untuk mengakses *firmware* Xbee menggunakan *AT commands* dan dapat mengirim atau menerima data dalam format Hex dan ASCII dengan memilih *Assemble Packet* (Gambar.2.16).



Gambar 2.16 Tab Terminal



Gambar 2.17 Assemble Packet

Pada Gambar 2.16 area putih dalam terminal *tab* ini berisi data informasi komunikasi yang terjadi antara 2 XBee atau lebih. Teks yang berwarna biru merupakan teks yang telah diketik pengguna dan diarahkan ke *port* serial XBee sedangkan teks merah merupakan data yang masuk dari *port* serial XBee.

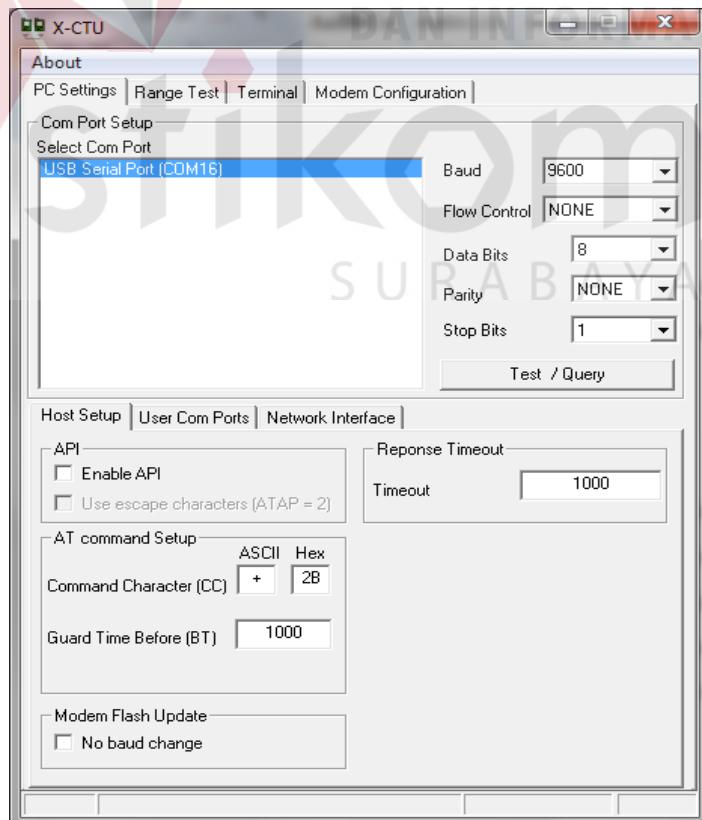
4. Modem Configuration Tab

Pada tab ini, pengguna dapat melakukan pemrograman pada pengaturan *firmware* Xbee dan merubah versi *firmwarenya* melalui *Graphical User Interface* (GUI). Terdapat 4 fungsi dasar dalam *modem configuration tab*, yaitu:

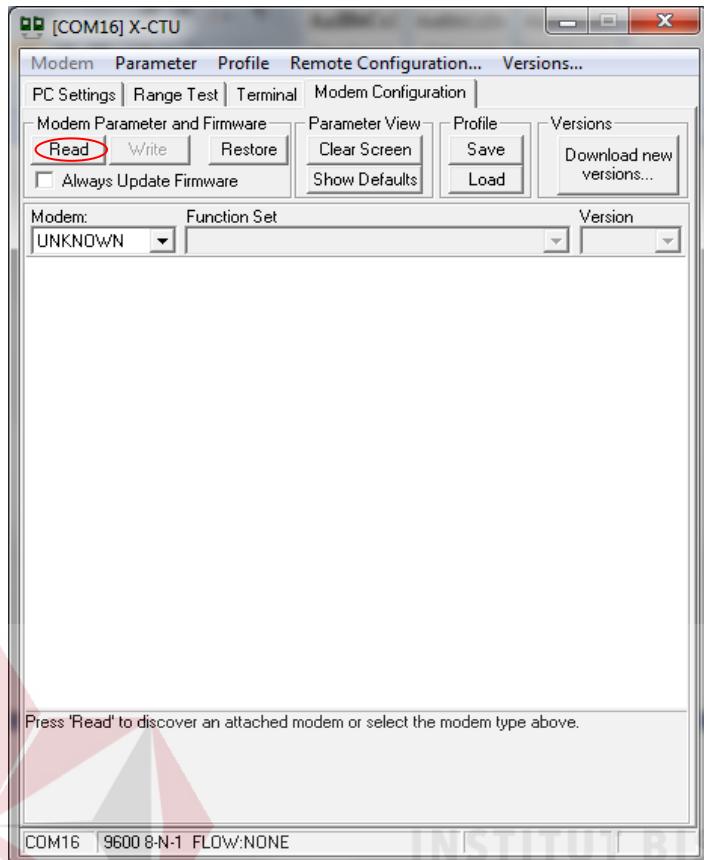
- Menyediakan fasilitas GUI untuk mengatur *firmware* pada Xbee.
- Read* dan *write* firmware ke mikrokontroller Xbee.

Untuk dapat membaca (*read*) firmware Xbee, pertama hubungkan Xbee dengan Xbee usb adapter yang telah terhubung dengan kabel mini usb, kemudian hubungkan kabel tersebut ke komputer melalui *interface* usb.

Jalankan aplikasi X-CTU pada komputer, selanjutnya atur *com port* pada tab *PC Settings* seperti Gambar 2.18. Pada *tab Modem Configuration*, klik “*Read*” pada bagian *Modem Parameter and Firmware* (Gambar 2.19).

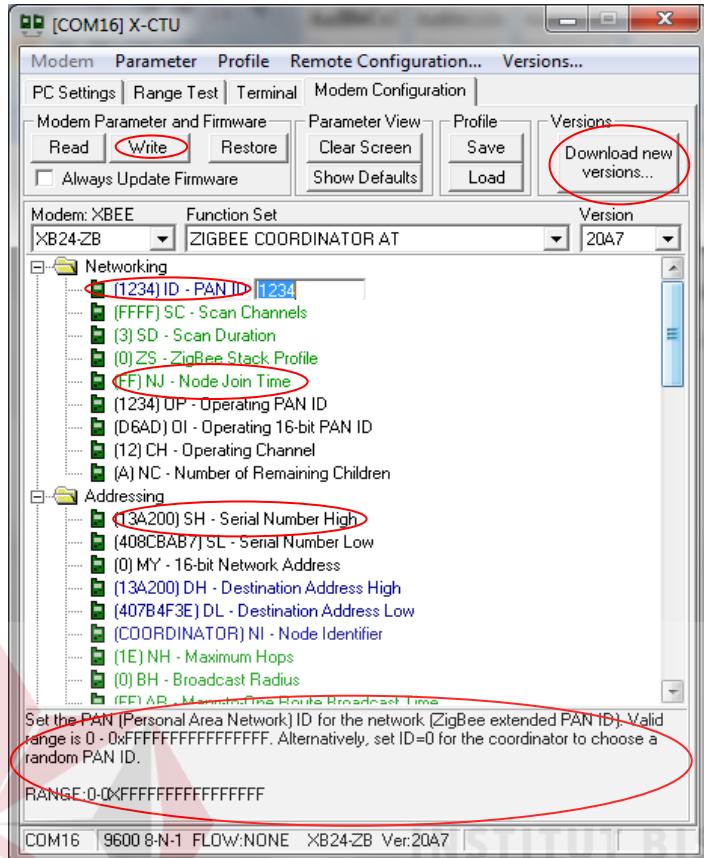


Gambar 2.18 Tab PC Settings



Gambar 2.19 Tab Modem Configuration

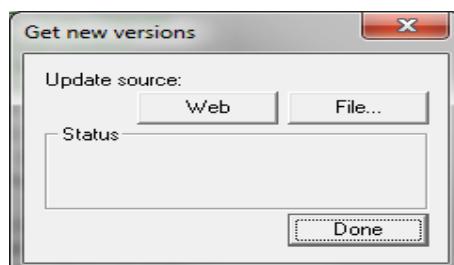
Setelah *firmware* Xbee telah terbaca, terdapat tiga warna dalam pengaturan konfigurasi (Gambar 2.20) yaitu hitam yang berarti *read-only* dan tidak bisa dirubah nilainya, kemudian hijau yang berarti nilai *default* Xbee dan biru yang berarti nilai yang pengguna tentukan sesuai keinginan. Untuk mengubah nilai parameter yang bisa dirubah, klik paramater tersebut kemudian ketik nilai yang baru sesuai keinginan pengguna. Untuk memudahkan pengisian nilai, terdapat deskripsi atau keterangan dalam pengisian nilai pada setiap parameter yang berada pada bagian bawah. Setelah semua nilai – nilai yang baru masuk, maka nilai tersebut dapat disimpan ke memori *non-volatile* pada Xbee. Klik tombol “Write” pada bagian *Modem Parameter and Firmware* untuk menyimpan konfigurasi parameter ke memori pada Xbee (Gambar 2.20).



Gambar 2.20 Firmware Xbee Telah Terbaca

c) Meng-update firmware pada Xbee

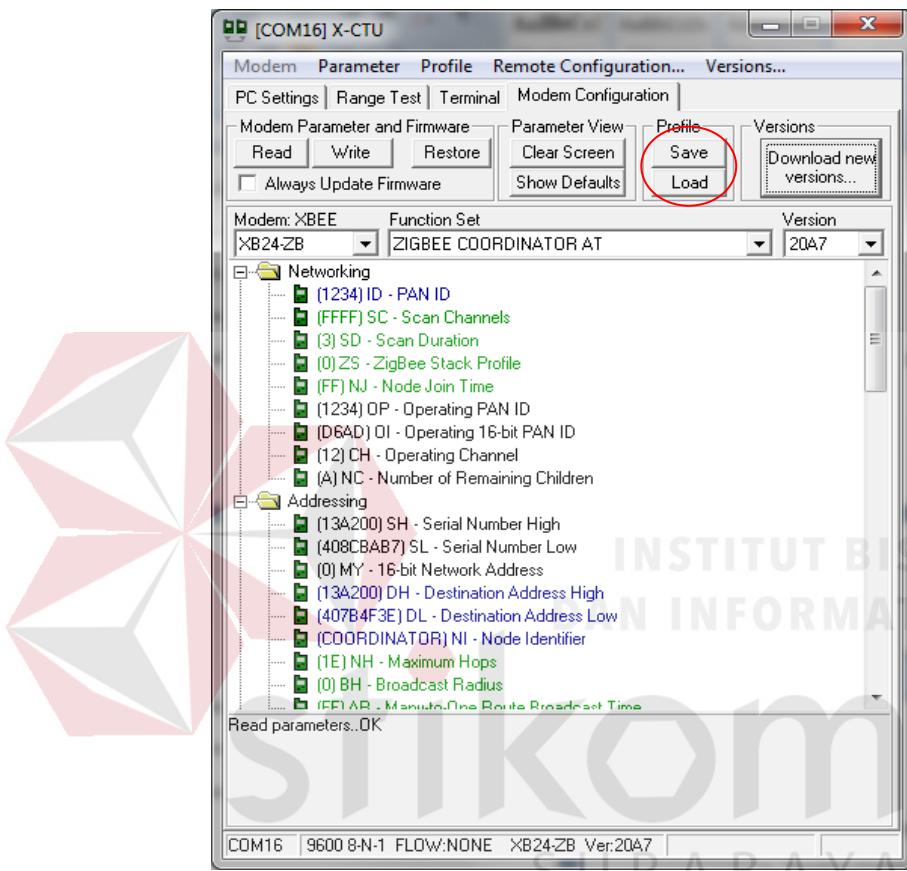
User atau pengguna dapat meng-update firmware pada Xbee baik melalui *web* atau menginstalnya dari *file.zip* atau *disk*. Klik “Download New Versions” pada bagian *Versions* (Gambar 2.20). Klik tombol *Web* jika ingin meng-update file firmware melalui *web* atau klik tombol *File* jika ingin meng-update melalui *file.zip* atau *disk* (Gambar 2.21).



Gambar 2.21 Kotak Dialog Get New Versions

d) Save atau *load modem profile*

X-CTU dapat menyimpan dan men-*load* profil modem atau konfigurasi yang telah disimpan, ini sangat berguna ketika parameter konfigurasi yang sama ditetapkan pada beberapa Xbee yang berbeda (Gambar 2.22).

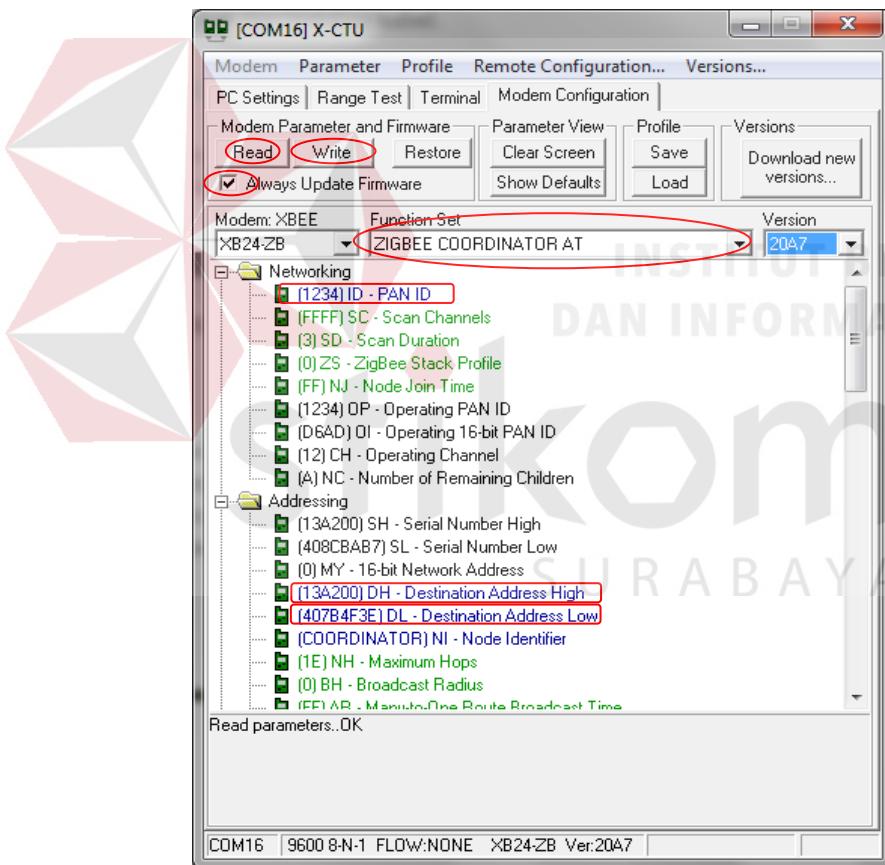


Gambar 2.22 Save dan Load Modem Profile

2.13 Topologi *Point to Point*

Topologi *point to point* adalah topologi yang membangun hubungan langsung antara dua *node* jaringan. Dalam jaringan sensor nirkabel atau WSN ini dapat menggunakan topologi *point to point*. Jenis *node* dan parameter yang harus dikonfigurasi agar dua Xbee dapat berkomunikasi secara *point to point* adalah salah satu *node* harus menjadi *coordinator* dan lainnya menjadi *router* atau *end device*. Klik “Read” dan “Always Update Firmware” pada tab *modem*

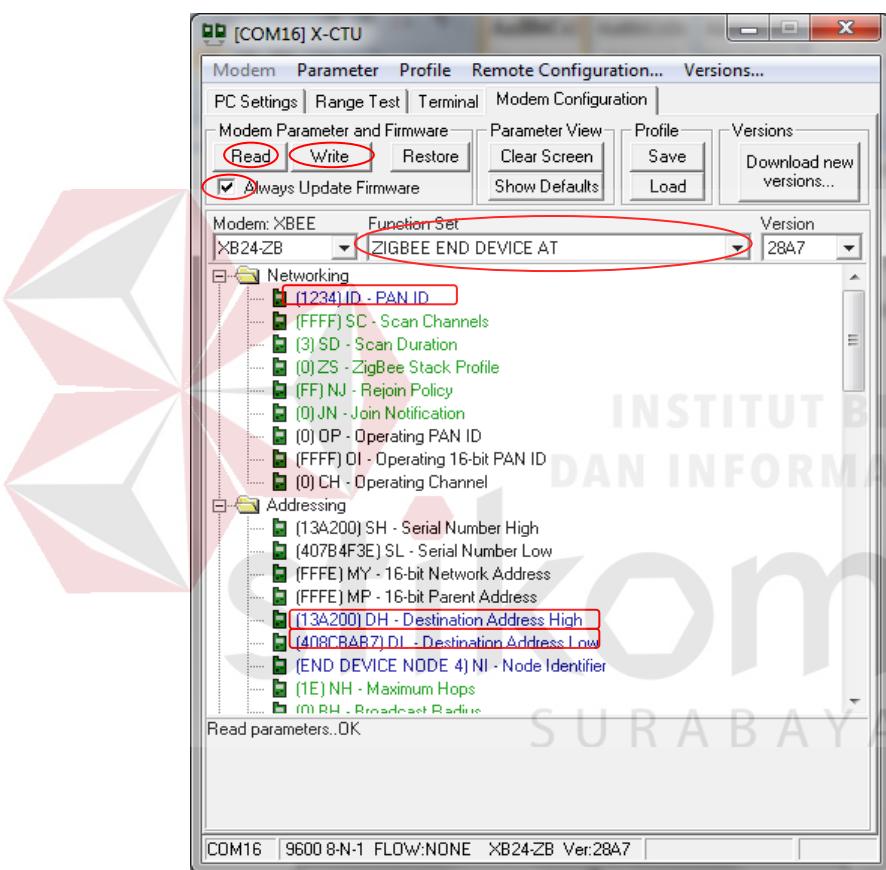
configuration dalam *software* X-CTU untuk dapat membaca modul Xbee, pada bagian *function set* diatur menjadi ZIGBEE COORDINATOR AT, selanjutnya parameter pada *node coordinator* yang harus diatur adalah parameter PAN ID, PAN ID dalam satu jaringan (antara dua Xbee) harus sama agar dapat berkomunikasi. Kemudian DH dan DL yang harus diatur sesuai dengan SH dan SL milik *router* atau *end device* yang merupakan *source address*. Jika jenis *node* dan parameter sudah diatur, setelah itu klik “*Write*” untuk menyimpan konfigurasi yang telah diatur ke dalam Xbee (Gambar 2.23).



Gambar 2.23 Konfigurasi Node Coordinator

Pada *node* yang akan dijadikan *router* atau *end device*, pada bagian *function set* diatur menjadi ZIGBEE ROUTER AT ATAU END DEVICE AT. Selanjutnya parameter pada *node router* atau *end device* yang harus diatur adalah

parameter PAN ID, PAN ID dalam satu jaringan (antara dua Xbee) harus sama agar dapat berkomunikasi dalam hal ini mengikuti PAN ID *coordinator*. Kemudian DH dan DL yang harus diatur sesuai dengan SH dan SL milik *coordinator* yang merupakan *source address*. Jika jenis *node* dan parameter sudah diatur, setelah itu klik “Write” untuk menyimpan konfigurasi yang telah diatur ke dalam Xbee (Gambar 2.24).

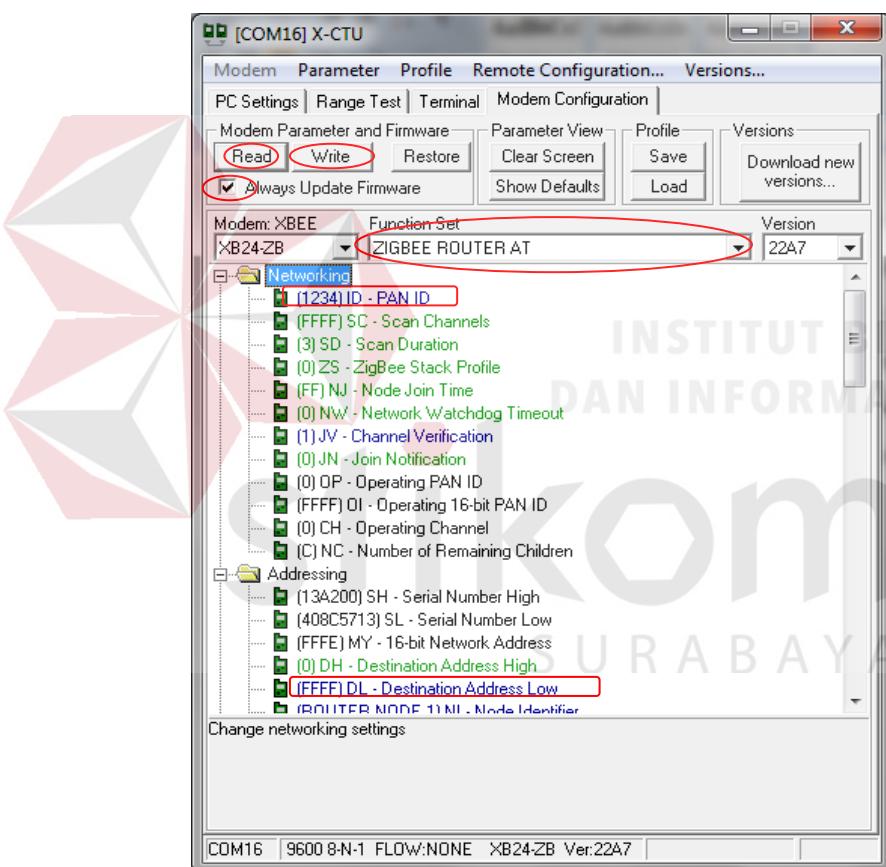


Gambar 2.24 Konfigurasi *Node End Device*

2.14 Topologi *Point to Multipoint*

Topologi *point to multipoint* adalah topologi yang membangun hubungan antara beberapa *node*, yang dimana suatu *node* dapat berkomunikasi dengan beberapa *node* secara *broadcast*. Jenis *node* dan parameter yang harus dikonfigurasi agar suatu Xbee dapat berkomunikasi secara *point to multipoint*

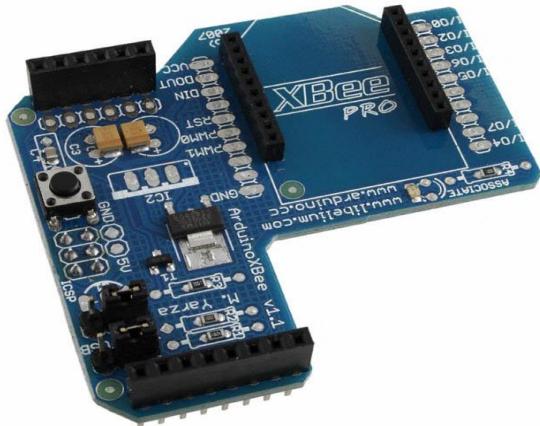
adalah salah satu *node* harus menjadi *coordinator* dan lainnya menjadi *router* atau *end device*. Parameter yang harus diatur hampir sama dengan parameter dalam komunikasi *point to point*, perbedaannya terletak pada DH dan DL *node* yang bersangkutan. Dalam *point to multipoint* ini, DH dan DL diatur nilainya menjadi DH = 0 dan DL = FFFF, FFFF mempunyai arti bahwa data akan dikirim secara *broadcast* sehingga beberapa *node* dalam PAN ID yang sama akan mendapat data tersebut (Gambar 2.25).



Gambar 2.25 Konfigurasi Point to Multipoint

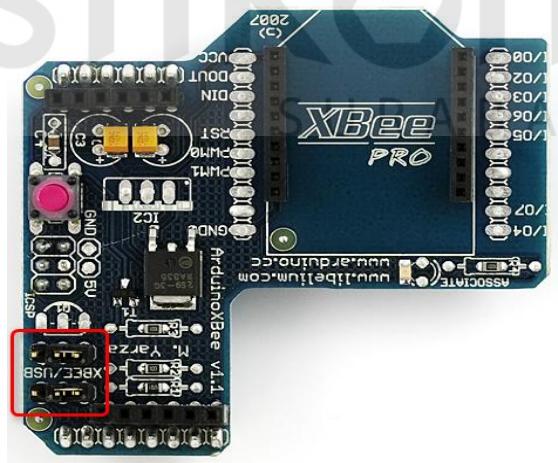
2.15 Xbee Shield

Xbee *shield* merupakan suatu *board* yang dapat menghubungkan *board* arduino untuk berkomunikasi secara nirkabel atau *wireless* menggunakan modul Xbee atau Zigbee (Gambar 2.26). (arduino.cc, 2013)



Gambar 2.26 Xbee Shield

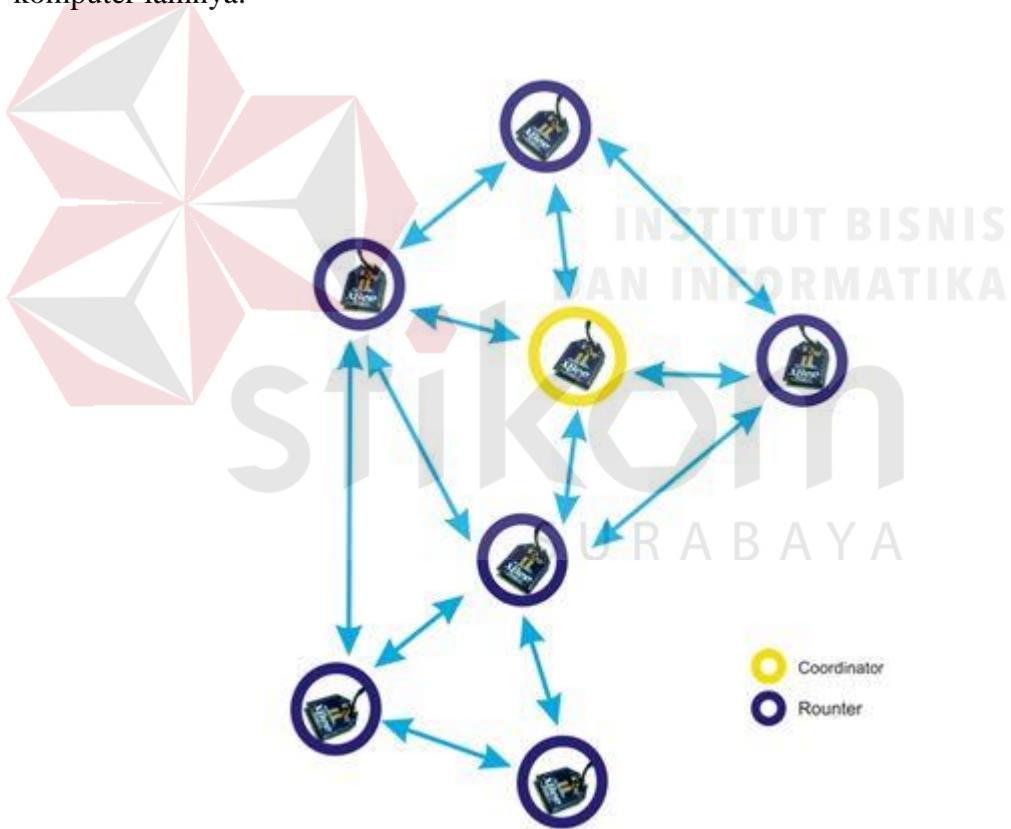
Xbee shield memiliki dua jumper terbuat dari plastik yang dapat di *removeable* dari tiga pin pada shied yang berlabel Xbee/USB (Gambar 2.27). Jumper ini menentukan komunikasi serial Xbee agar terhubung pada komunikasi serial antar mikrokontroller atau USB pada *board arduino*. (arduino.cc, 2013)



Gambar 2.27 Jumper pada Xbee Shield

2.16 Topology Mesh Network

Topologi mesh adalah suatu bentuk hubungan antar perangkat/node dimana setiap perangkat terhubung secara langsung ke perangkat lainnya. Tiap-tiap node dalam topologi mesh tidak hanya berfungsi sebagai penerima data untuk dirinya sendiri namun juga sebagai penyedia data untuk perangkat/ node yang lainnya. Memiliki sifat Robust, yaitu Apabila terjadi gangguan pada koneksi komputer A dengan komputer B karena rusaknya kabel koneksi (links) antara A dan B, maka gangguan tersebut tidak akan memengaruhi koneksi komputer A dengan komputer lainnya.

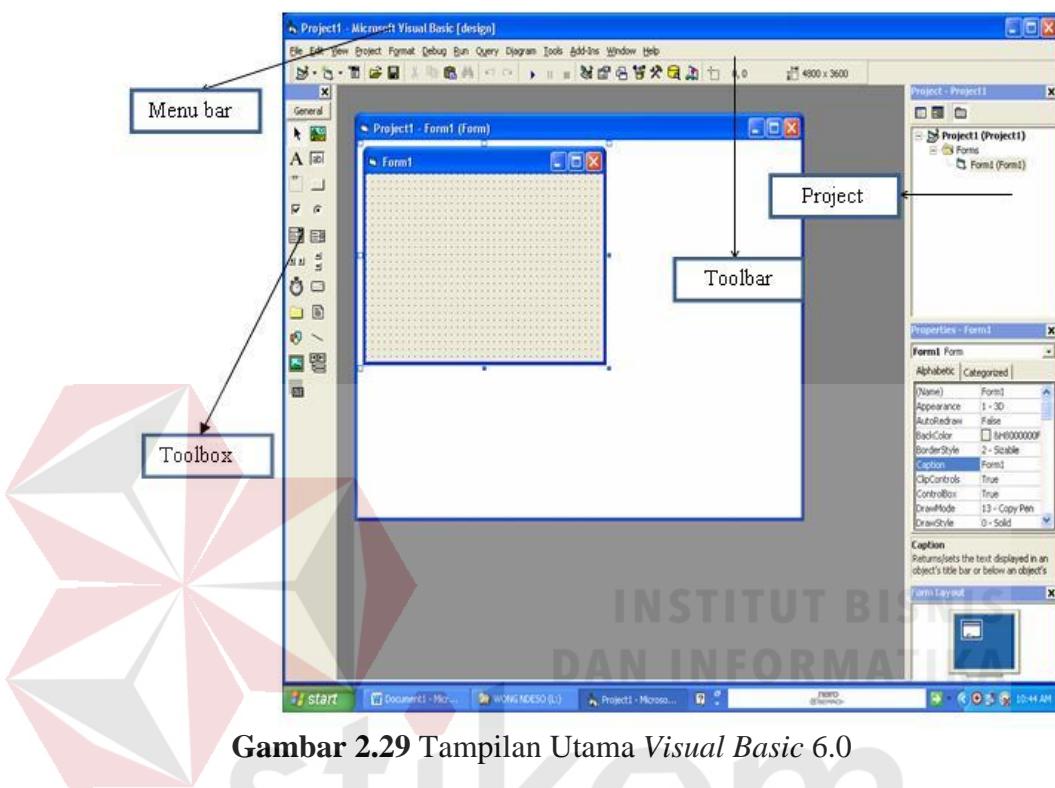


Gambar 2.28 Topologi *Mesh Network*

2.17 Visual Basic

Visual Basic adalah salah suatu *development tools* untuk membangun aplikasi dalam lingkungan *Windows*. Dalam pengembangan aplikasi, *Visual Basic*

menggunakan pendekatan *Visual* untuk merancang *user interface* dalam bentuk *form*. Tampilan *Visual Basic* terdapat pada *Integrated Development Environment* (IDE) seperti pada gambar di bawah ini:



Gambar 2.29 Tampilan Utama *Visual Basic* 6.0

Adapun penjelasan jendela-jendela adalah sebagai berikut :

- Menu Bar, digunakan untuk memilih tugas-tugas tertentu seperti menyimpan project, membuka project, dll
- Main Toolbar, digunakan untuk melakukan tugas-tugas tertentu dengan cepat.
- Jendela Project, jendela berisi gambaran dari semua modul yang terdapat dalam aplikasi.
- Jendela Form Designer, jendela merupakan tempat anda untuk merancang *user interface* dari aplikasi.

- e. Jendela *Toolbox*, jendela berisi komponen-komponen yang dapat anda gunakan untuk mengembangkan *user interface*.
- f. Jendela *Code*, merupakan tempat bagi anda untuk menulis coding. Anda dapat menampilkan jendela dengan menggunakan kombinasi Shift-F7.
- g. Jendela *Properties*, merupakan daftar properti-properti *object* yang sedang terpilih. Sebagai contohnya anda dapat mengubah warna tulisan (foreground) dan warna latarbelakang (*background*). Anda dapat menggunakan F4 untuk menampilkan jendela properti.



BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

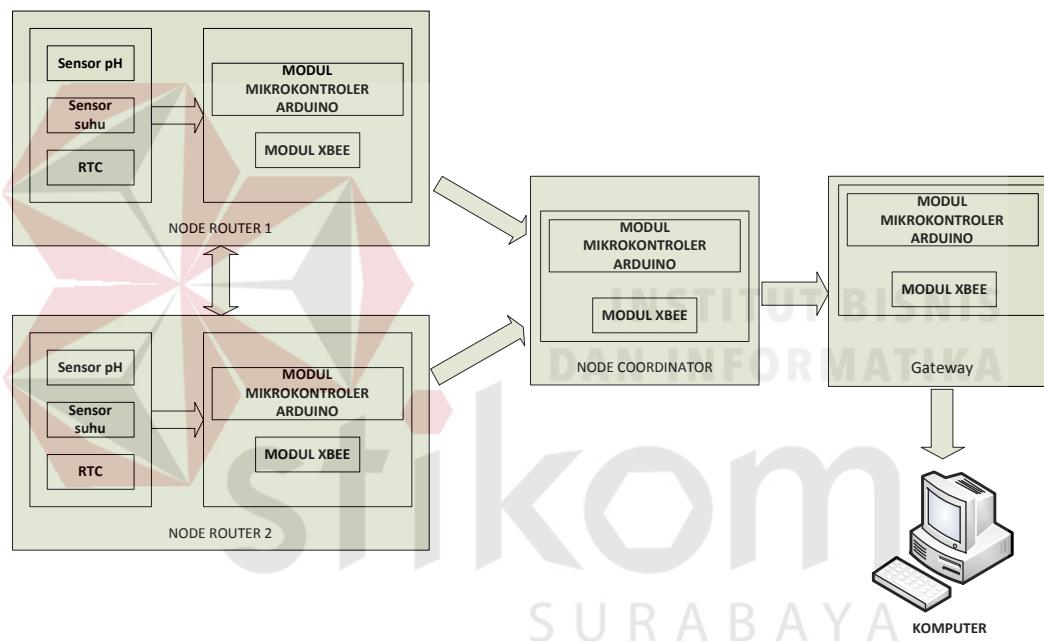
Metode penelitian yang digunakan pada tugas akhir ini melalui beberapa tahapan penelitian dan mencari informasi tentang data yang dibutuhkan dalam melakukan tugas akhir ini. Peneltiaan pertama adalah pengembangan konsep penelitian berdasarkan daftar pustaka. Selanjutnya perencanaan penelitian meliputi perancangan sistem perangkat keras dan perangkat lunak. Dengan cara ini penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dari buku, bahan-bahan kuliah dan internet yang berkaitan dengan permasalahan. Informasi data-data meliputi sensor Analog pH Meter Kit, sensor suhu DS18B20 dan modul RTC sebagai input data kadar pH dan suhu air pada prototipe secara *real time*, kemudian diproses dalam modul mikrokontroler arduino uno. Modul Xbee series 2 sebagai *receiver* dan *transmitter* data secara nirkabel dari dan ke *node* lainnya serta informasi dalam penerimaan dan pengirimin data.

Setelah didapatkan informasi mengenai hal-hal yang dibutuhkan maka langkah selanjutnya adalah membuat skrip perancangan sistem WSN menggunakan *software* arduino IDE untuk menghasilkan informasi data yang nantinya akan digunakan dalam pengujian pengiriman dan penerimaan data pH air dan suhu air secara *real time*.

Pada bab ini dibahas mengenai masalah yang timbul dalam perencanaan dan pembuatan perangkat keras (*hardware*) maupun perangkat lunak (*software*).

Dari kedua bagian tersebut akan dipadukan / diintegrasikan agar dapat bekerja sama menjalankan sistem dengan baik.

Gambar 3.1 merupakan gambar blok diagram sistem yang merupakan penjelasan singkat dari perancangan sistem yang dibuat pada judul tugas akhir “Rancang Bangun Wireless Sensor Network (WSN) Untuk Monitoring Pencemaran Air Sungai Menggunakan Topologi Mesh Network”.



Gambar 3.1 Blok Diagram Sistem

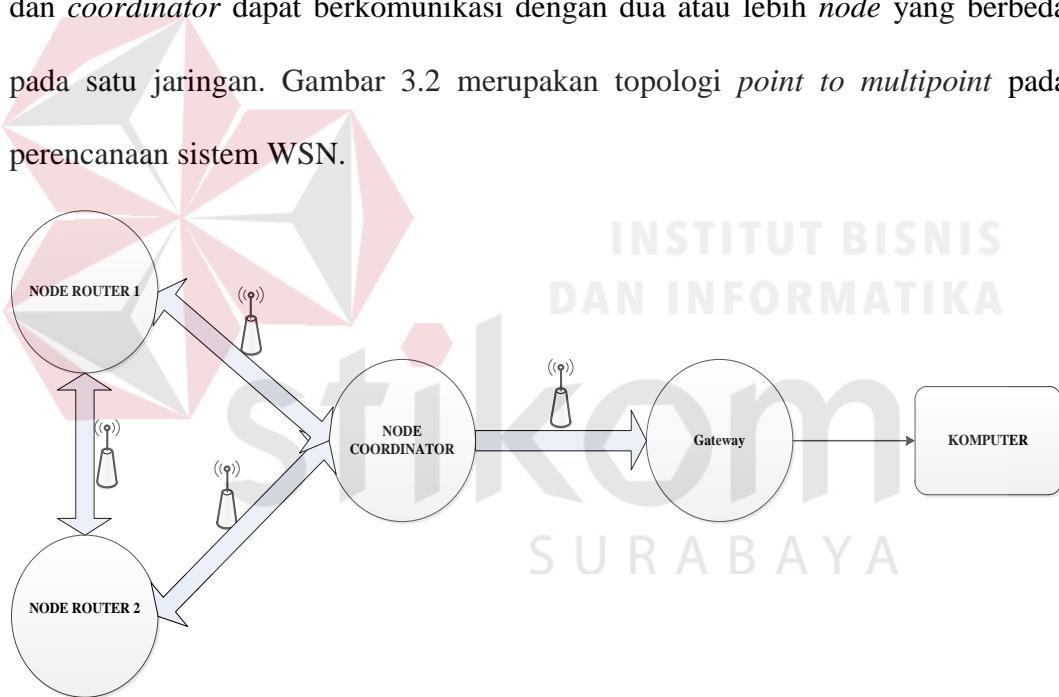
Dari blok diagram sistem Gambar 3.1, terbagi menjadi 4 kelompok bagian, yaitu bagian *Router 1*, *Router 2*, *Coordinator* dan *End User*.

- 1 *Router 1* yang terdiri dari sensor pH, sensor suhu dan RTC yang terhubung dengan kabel serta arduino dan xbee yang terhubung dengan *shield xbee*.
- 2 *Router 2* yang terdiri dari sensor pH, sensor suhu dan RTC yang terhubung dengan kabel serta arduino dan xbee yang terhubung dengan *shield xbee*.

- 3 *Coordinator* yang terdiri dari arduino dan xbee yang terhubung dengan *shield* xbee.
- 4 *End User* yang terdiri dari arduino dan xbee yang terhubung dengan *shield* xbee dan arduino yang terhubung dengan komputer melalui kabel.

3.2 Desain Topologi

Pada perancangan sistem ini menggunakan model topologi *point to multipoint*. Model topologi *point to multipoint* ini digunakan agar *node router* 1, 2 dan *coordinator* dapat berkomunikasi dengan dua atau lebih *node* yang berbeda pada satu jaringan. Gambar 3.2 merupakan topologi *point to multipoint* pada perencanaan sistem WSN.



Gambar 3.2 Topologi *Point to Multipoint*

Pada perencanaan sistem WSN menggunakan topologi *point to multipoint* ini, diharapkan proses pengiriman masing-masing *node* dapat berjalan lancar.

1. Jumlah *node* yang digunakan berjumlah 4, yaitu *node router* 1, 2, *coordinator* dan *end device*.

2. *Node router 1* akan mengirimkan data pH air, suhu air dan waktu secara *broadcast* dan nirkabel ke *node router 2* dan *node coordinator*.
3. *Node router 2*, akan mengirimkan data pH air, suhu air dan waktu secara *broadcast* dan nirkabel ke *node router 1* dan *node coordinator*.
4. *Node coordinator* menerima data dari *node router 1* dan *node router 2* baik itu melalui *node router* itu sendiri atau dari *node router 1* yang menuju *coordinator* melewati *node router 2* maupun dari *node router 2* yang menuju *coordinator* melewati *node router 1*, maka data-data tersebut akan dirata-rata. Kemudian hasilnya akan dikirimkan ke *node end device*.
5. *Node end device* menerima data dari *node coordinator*.
6. Komputer akan mengambil data yang terdapat di *node end devices*, selanjutnya akan menampilkannya melalui program yang telah dibuat.

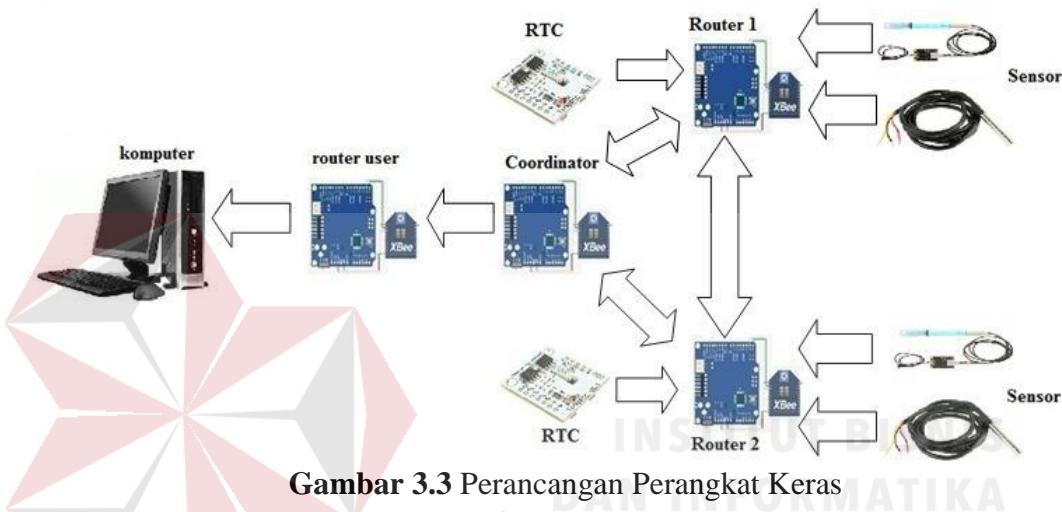
Catatan :

Node End Device dapat mengirimkan data kepada node coordinator apabila node coordinator lebih dari satu, karena apabila node coordinator lebih dari 1 dan node coordinator mengirimkan datanya secara bersamaan maka node end device tidak dapat menerima semua data yang dikirim dari node coordinator secara bersamaan, oleh karena itu nantinya node end device akan meminta data kepada node coordinator terlebih dahulu.

3.3 Perancangan Perangkat Keras

Gambar 3.3 merupakan perangkat keras yang dibutuhkan dalam tugas akhir ini. Pada *node router 1* dan *2* perangkat keras yang dibutuhkan yaitu sensor Analog pH Meter Kit untuk mendapatkan berapa kadar pH yang terkandung didalam air, sensor suhu DS18B20 untuk mendapatkan suhu air, modul RTC

untuk mendapatkan tanggal dan waktu, modul mikrokontroler arduino uno untuk *processing*, Xbee series 2 untuk mengirim atau menerima data secara nirkabel, dan Xbee shield sebagai penghubung antara modul arduino uno dengan Xbee series 2. Sedangkan pada *node coordinator* perangkat keras yang dibutuhkan terdiri dari modul mikrokontroler arduino uno, Xbee series 2 dan Xbee shield.



Gambar 3.3 Perancangan Perangkat Keras

Berdasarkan desain topologi sebelumnya *node router* 1, 2, dan *coordinator* dapat berkomunikasi secara *point to multipoint*. Agar dapat berkomunikasi *point to multipoint*, terdapat parameter-parameter yang harus diatur didalam Xbee series 2 menggunakan *software X-CTU*. Parameter-parameter tersebut adalah PAN ID dalam satu jaringan harus sama, kemudian DH dan DL yang diatur nilainya menjadi DH = 0 dan DL = FFFF, FFFF mempunyai arti bahwa data akan dikirim secara *broadcast* sehingga beberapa *node* dalam PAN ID yang sama akan mendapat data tersebut (Gambar 2.25).

Sistem mulai berjalan ketika “ok” pada node *router* 1 bernilai “0” maka program akan memberi delay selama 5 detik, setelah 5 detik “ok” akan bernilai “1” apabila “ok” telah bernilai “1” maka selanjutnya *router* 1 akan

mengirimkan data *router* 1 yang mempunyai ID “C1” dan berisi data pH, data suhu, data tanggal, dan data jam ke node *coordinator* dan *node router* 2. Setelah data sampai di *node router* 2 selanjutnya *node router* 2 akan merubah ID data yang diterima dari *node router* 1 yang awalnya mempunyai ID “R” akan dirubah menjadi “Q”. Setelah ID berubah selanjutnya *node router* 2 akan meneruskan data yang diterima dari *node router* 1 ke *node coordinator*. Setelah melakukan pengiriman data, “ok” akan dirubah nilainya menjadi “0”. Setelah nilai “ok” menjadi “0” maka selanjutnya sama seperti pada *node router* 1 akan didelay selama 5 detik, setelah 5 detik “ok” akan bernilai “1” apabila “ok” telah bernilai “1” maka selanjutnya *router* 2 akan mengirimkan data *router* 2 yang mempunyai ID “C2” dan berisi data pH, data suhu, data tanggal, dan data jam ke node *coordinator* dan *node router* 1. Setelah data sampai di *node router* 2 selanjutnya *node router* 1 akan merubah ID data yang diterima dari *node router* 2 yang awalnya mempunyai ID “R” akan dirubah menjadi “Q”. Setelah ID berubah selanjutnya *node router* 1 akan meneruskan data yang diterima dari *node router* 2 ke *node coordinator*. Setelah melakukan pengiriman data, “ok” akan dirubah nilainya menjadi “0”. Dengan melakukan proses pengiriman data seperti diatas maka sistem pengiriman data seperti diatas dapat disebut sebagai proses pengiriman data menggunakan topologi *mesh network*. Setelah *node coordinator* menerima 5 data dari *node router* 1 dan *node router* 2 maka selanjutnya *coordinator* akan melakukan rata-rata dari data yang telah diterima. *Node coordinator* akan melakukan rata-rata data pH, dan data suhu. Setelah hasil rata-rata didapatkan maka selanjutnya *coordinator*

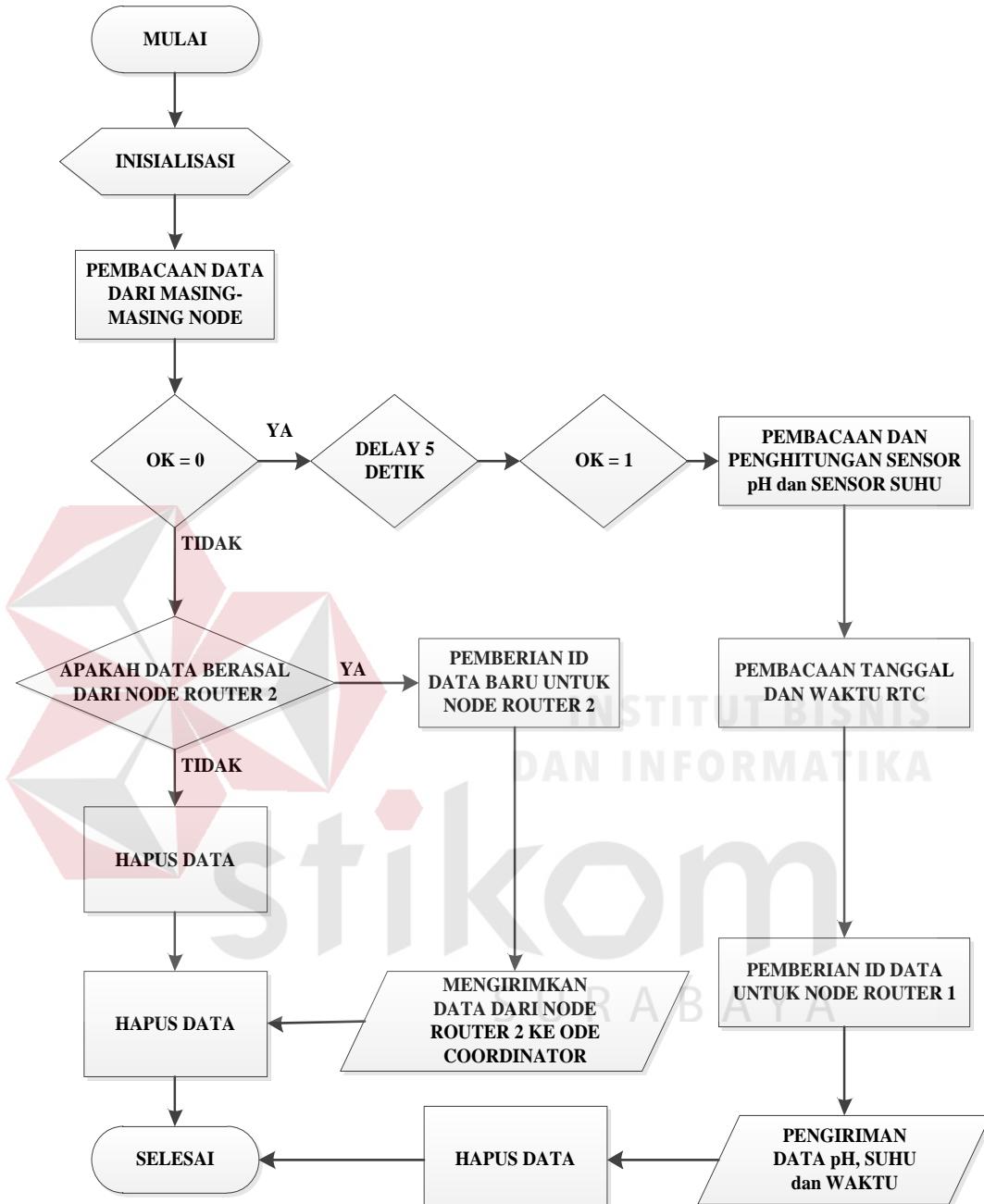
akan mengirimkan hasil rata-rata ke *node end device* untuk ditampilkan pada program *visual basic*.

3.4 Pemrograman mikrokontroler Arduino Uno pada *Software ArduinoIDE*

Untuk melakukan perancangan sistem ini terlebih dahulu harus membuat skrip atau melakukan pemograman mikrokontorler arduino uno pada *software arduino IDE* agar mendapat hasil sesuai dengan yang dibutuhkan.



3.4.1 Membuat Skrip untuk *Node Router 1*



Gambar 3.4 Diagram Alir Pembuatan Skrip pada *Node router 1*

Berikut ini penjabaran dari Gambar 3.4 proses pembuatan skrip pada *node router 1*.

1. Inisialisai

Pembuatan skrip ini dimulai dengan menuliskan skrip berikut ini. skrip ini merupakan proses inisialisasi dan harus ada dalam setiap program yang dibuat pada *node router* seperti pada gambar *flowchart 3.4*.

```
// library yang digunakan
#include <Wire.h>
#include "RTClib.h"
#include <OneWire.h>
#include <String.h>
// inisialisasi variabel
#define SensorPin 0 //pH meter Analog output to Arduino Analog
Input 0
int DS18S20_Pin = 3; //DS18S20 Signal pin on digital 3 Temperature
chip i/o
OneWire ds(DS18S20_Pin); // on digital pin 3
String cek,id,dta1,dta2;
int x=0;
unsigned long int avgValue; //Store the average value of the
sensor feedback
int buf[10],temp,a=1;
// inisialisasi RTC
RTC_DS1307 RTC;

void setup()
{
    // inisialisasi Wire (komunikasi I2C)
    Wire.begin(); // Start the Wire (I2C communications)
    // start RTC
    RTC.begin();
    // hanya sekali digunakan untuk mengatur tanggal dan waktu
    // pada modul RTC
    //RTC.adjust(DateTime(__DATE__, __TIME__));
    // inisialisasi komunikasi serial
    Serial.begin(9600);
}
//semua proses yang dilakukan secara berulang-ulang akan
dimasukkan ke dalam fungsi loop ini
void loop() {}
```

2. Pembacaan Data *Node*

Proses pembacaan data dilakukan secara berulang-ulang oleh masing-masing *node* ketika “ok” bernilai “0” maka akan diberi jeda selama 5 detik, setelah dijeda selama 5 detik maka “ok” akan bernilai 1, apabila “ok” bernilai 1 maka *node router* 1 akan mengirimkan data *node router* 1. Ketika ada data yang dikirimkan dari *node* lain dan kemudian data tersebut diterima (data masuk

diakhiri dengan karakter `\n` atau *enter*), gambar *flowchart* 3.4. Contoh skrip pembacaan data dari suatu *node* dalam jaringan.

```
void loop() {
if (ok==0)
{
delay(5000);
ok=1;
dta1=ambldata();
Serial.print(dta1); //meneruskan atau mengirim data router 1 ke
node coordinator
}
while(Serial.available()>0){
    char data_masuk=(char)Serial.read();
    id += data_masuk;
    if(data_masuk =='\n'){}
}
}
```

3. Menyeleksi Data dan Memberikan id baru untuk *Node Router 2* serta

Meneruskan Data *Node Router 2*

Setelah data yang diakhiri karakter `\n` diterima, dan data tersebut berasal dari *node router 2* yang melakukan pengiriman data (mengirim data *router 2*), maka *node router 1* akan melakukan penyeleksian dan pemberian ID baru untuk *node router 2*. Proses pemberian ID data baru ini dilakukan untuk mempermudah pengguna mengetahui ID data *node router 2* yang melalui *node router 1* untuk selanjutnya akan diteruskan ke *node coordinator*. ID data *node router 2* akan berubah dari “R2” menjadi “Q2” jika melalui *node router 1*. Setelah itu “ok” akan diriset kembali menjadi “0”. Berikut skrip penyeleksian dan pemberian ID baru data yang berasal dari *node router 2* serta meneruskan data ke *coordinator*, gambar 3.4.

```
void loop(){
if(data_masuk =='\n')
{
//menerima data dari R2 dan skrip penyeleksian
if((id[0]=='$') && (id[1]=='R') && (id[2]=='2'))
{
    id[1]='Q'; // perubahan ID "R" menjadi "Q"
    delay(5000);
    //meneruskan atau mengirim data ke node coordinator
    Serial.print(id);
}
}
```

```
    id="";
    // nilai dalam variabel "id" dikosongkan
    ok=0;
}
```

4. Pembacaan sensor pH

Node router akan melakukan proses pembacaan dan perhitungan nilai kadar pH air yang dilakukan sensor analog pH meter kit, gambar 3.4. Contoh skrip sebagai berikut.

```
float ph()
{
    pinMode(SensorPin,OUTPUT);
    //mendapatkan 10 nilai sampel dari sensor
    for(int i=0;i<10;i++)
    {
        buf[i]=analogRead(SensorPin);
        delay(10);
    }
    for(int i=0;i<9;i++) //mengurutkan dari kecil ke besar
    {
        for(int j=i+1;j<10;j++)
        {
            if(buf[i]>buf[j])
            {
                temp=buf[i];
                buf[i]=buf[j];
                buf[j]=temp;
            }
        }
    }
    avgValue=0;
    //mengambil nilai rata-rata dari 6 sampel
    for(int i=2;i<8;i++)
    avgValue+=buf[i];
    //konversi dari analog ke millivolt
    float phValue=(float)avgValue*5.0/1024/6;
    //konversi dari millivolt kedalam nilai pH
    phValue=3.5*phValue;
    digitalWrite(SensorPin, HIGH);
    delay(800);
    digitalWrite(SensorPin, LOW);
    return(phValue);
}
```

Sensor ph akan mengambil 10 nilai sampel dari sensor, lalu akan ditampung pada buf, kemudian akan mengurutkan dari nilai terkecil kenilai terbesar. Setelah itu merata-rata 6 sampel yang telah diambil tadi lalu dimasukan kedalam “avgValue”. Selanjutnya mengkonversi dari analog kedalam millivolt dengan cara “float phValue=(float)avgValue*5.0/1024/6” setelah

didapatkan nilai phValue sekarang mengkonversi dari millivolt kedalam nilai pH dengan cara “`phValue=3.5*phValue`” setelah itu dilakukan baru didapatkan berapa nilai pH tersebut.

5. Pembacaan sensor suhu

```

float getTempSuhu() {
    //returns the temperature from one DS18S20 in DEG Celsius

    byte data[12];
    byte addr[8];

    if ( !ds.search(addr) ) {
        //no more sensors on chain, reset search
        ds.reset_search();
        return 34;
    }

    if ( OneWire::crc8( addr, 7) != addr[7] ) {
        Serial.println("CRC is not valid!");
        return 35;
    }

    if ( (addr[0] != 0x10 && addr[0] != 0x28) {
        Serial.print("Device is not recognized");
        return 36;
    }

    ds.reset();
    ds.select(addr);
    ds.write(0x44,1); // start conversion, with parasite power
                      // on at the end

    byte present = ds.reset();
    ds.select(addr);
    ds.write(0xBE); // Read Scratchpad

    for (int i = 0; i < 9; i++) { // we need 9 bytes
        data[i] = ds.read();
    }

    ds.reset_search();

    byte MSB = data[1];
    byte LSB = data[0];

    float tempRead = ((MSB << 8) | LSB); //using two's
    compliment
    float TemperatureSum = tempRead / 16;
    return TemperatureSum;
}

```

DS18S20 yang merupakan sensor suhu dengan *output* data berupa digital maka dibutuhkan library DS18S20 yang dapat diunduh di <http://playground.arduino.cc/Learning/OneWire>. Setelah selesai diunduh ekstrak file tersebut ke dalam folder *libraries* pada *directory* folder yang telah diunduh pada saat menginstal *driver* arduino. *Library* ini berfungsi untuk mengenali sensor DS18S20 supaya bisa berkomunikasi dengan arduino. Selanjutnya buka IDE arduino dan isikan *code* program diatas untuk mengukur suhu menggunakan sensor DS18S20.

6. Pembacaan Tanggal dan Waktu RTC

Proses pembacaan tanggal dan waktu digunakan untuk mengetahui waktu pada saat pengambilan data kadar pH dan suhu air oleh sensor pH dan sensor suhu. Dengan cara ini dapat mengetahui perubahan pH dan suhu air secara *realtime*, gambar 3.4.

```
String datetime ()
{
    //date
    String date1;
    DateTIme now = RTC.now(); //mengambil tanggal dan waktu
    //format tanggal dan waktu
    date1=String(now.day())+"/"+String(now.month())+"/"+_
    String(now.year())+"_"+String(now.hour())+":"+
    String(now.minute()) + ":"+String(now.second());
    return (date1);
}
```

7. Konversi Data Integer ke String

Proses mengkonversi data dari float ke sting dibutuhkan karena ada beberapa fungsi yang mempunyai tipe data float, gamabr 3.4. Berikut ini skrip untuk mengkonversi data dari float ke string.

```
String konvrsi( float data) //mengkonversi integer ke string
{
    String out;
    int puluhan,satuan,koma,temp,temp2;
    temp2=data;
```

```

puluhan=temp2/10;
satuan=temp2%10;
temp=(puluhan*10)+satuan;
koma=(data*1000)-(temp*1000);
out=String(temp)+"."+String(koma);
return(out);
}

```

8. Pemberian ID Data dan Pengiriman Data

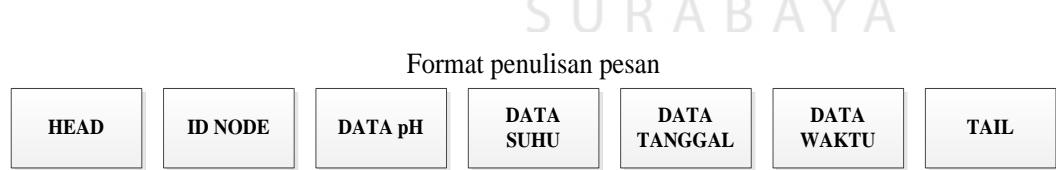
Pemberian ID data ini dilakukan untuk mempermudah pengguna melihat data pH air, suhu air dan waktu yang berasal dari beberapa *node router*. *Node router* 1 memiliki simbol “\$” dan ID data berawalan “R1” dan *node router* 2 memiliki ID data berawalan “R2” yang diikuti dengan data pH air, suhu air air dan waktu, gambar *flowchart* 3.4. Kemudian data tersebut dikirim ke *coordinator* dan *node router* lainnya.

```

String ambldata()
{
    String tmp;
    // format pengiriman data dari router 1 ke router 2 atau ke
    // coordinator
    tmp="$R1_"+konvrsi(ph())+"_"+konvrsi(getTempSuhu())+"_"+datetime()
    +"\%\n";
    return(tmp);
}

```

Format pengiriman atau penulisan pesan yang terjadi pada setiap *node* diatur seperti Gambar 3.7.



Gambar 3.5 Format Penulisan Pesan

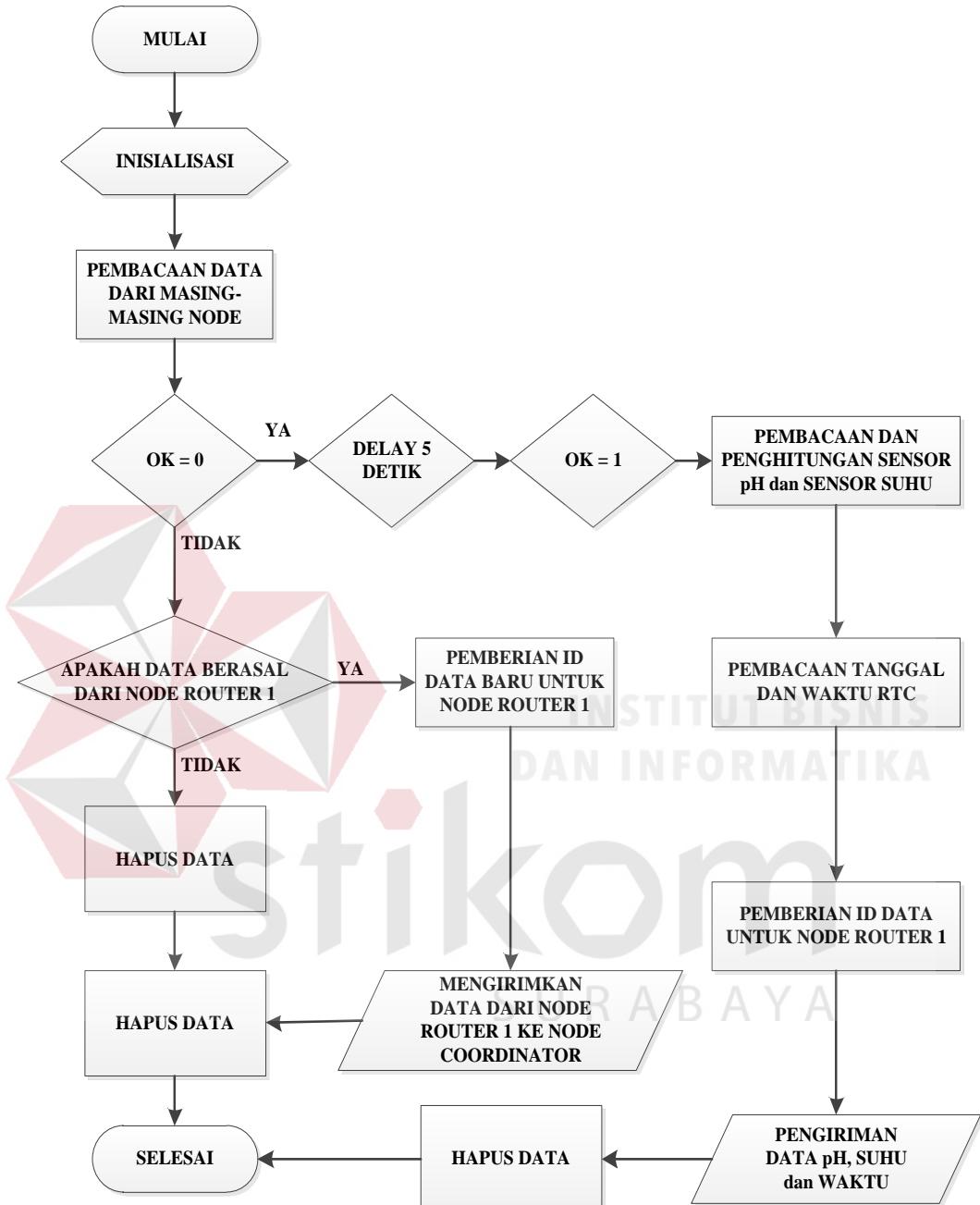
Setiap *node* memiliki *head* dan id *node* yang harus dikenali. Semua penulisan tersebut ditulis semua dan dipisahkan oleh tanda garis bawah. Dalam tugas akhir ini penulis menggunakan simbol \$ untuk digunakan sebagai *head*. Maka setiap *node* akan mengetahui apabila data yang diterima harus mempunyai awalan \$. Simbol *head* disini akan mengantisipasi sebuah *node* ilegal yang tiba-

tiba mengirimkan data ke *node*. Apabila data yang diterima tidak memiliki *head*, maka data tersebut akan dianggap ilegal dan *node* akan membuang data tersebut.

Dari penjelasan di atas, *node router* 1 harus menyeleksi data yang berhak untuk diterima olehnya. Karena *node router* 1 hanya diijinkan untuk menerima data dari *node router* 2, maka data tersebut harus memiliki *head* dan id R. Apabila data yang diterima tidak memiliki simbol tersebut maka data akan dibuang karena dianggap ilegal.



3.4.2 Membuat Skrip untuk *Node Router 2*



Gambar 3.6 Diagram Alir Pembuatan Skrip pada *Node router 2*

Pada Gambar 3.6 proses diagram alir pembuatan skrip pada *node router 2* hampir sama dengan diagram alir pada *node router 1*. Perbedaannya hanya terdapat pada pembuatan skrip untuk pemberian ID data *node router 2* itu sendiri dan pemberian ID data baru untuk *node router 1* yang melalui *node router 2*. ID

data *node router* 2 diberi awalan “R2”. Sedangkan ID data *node router* 1 yang sebelumnya berawalan “R1” berubah menjadi “Q1” karena data tersebut melalui *node router* 2 untuk diteruskan ke *node coordinator*. Berikut ini merupakan perbedaan skrip yang terdapat pada *node router* 2.

```
String ambldata()
{
    String tmp;
    // format pengiriman data dari router 2 ke router 1 atau ke
    // coordinator
    tmp="$R2_"+konvrsi(ph())+"_"+konvrsi(getTempSuhu())+"_"+datetime()
    +"%\n";
    return(tmp);
}
```

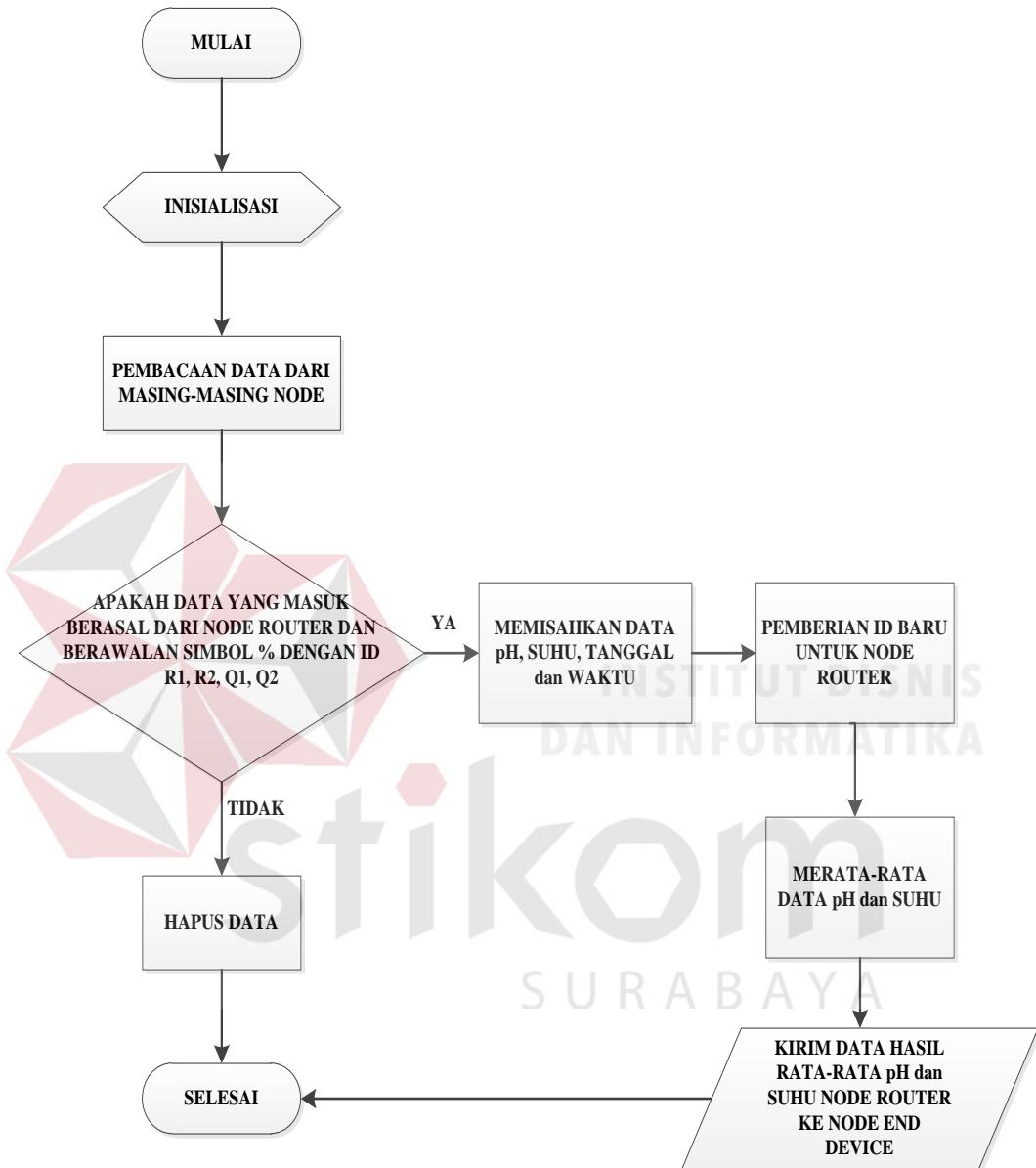
Kemudian skrip perubahan ID data *node router* 1.

```
if(data_masuk =='\n')
{
    //menerima data dari R1 dan skrip penseleksian
    if((id[0]=='$') && (id[1]=='R') && (id[2]=='1'))
    {
        id[1]='Q'; // perubahan ID "R" menjadi "Q"
        delay(5000);
        Serial.print(id); //meneruskan atau mengirim data ke node
        //coordinator
        id=""; // nilai dalam variabel "id" dikosongkan
        ok=0; }
```

3.4.3 Membuat Skrip untuk *Node Coordinator*

Pada perencanaan sistem ini, *node coordinator* akan menerima data dari *node router* 1 dan *node router* 2. *Node coordinator* hanya menerima data dari *node router* 1 dan *node router* 2 dengan format pesan data berawalan “\$” dan memiliki ID “R1”, “R2”, “Q1” dan “Q2”. Ketika keempat data ini diterima, maka seluruh ID akan dirubah menjadi “C”. Data yang dikirimkan oleh *router* 1 atau *router* 2 akan disimpan datanya sampai berjumlah 5 data. Dari 5 data data tersebut nantinya akan dirata-rata terlebih dahulu, data yang dirata-rata adalah data pH dan

data suhu. Dan hasil rata-rata data ini akan dikirimkan ke *node end device* yang kemudian akan ditampilkan di komputer menggunakan program *visual basic*.



Gambar 3.7 Diagram Alir Proses *Node Coordinator*

Berdasarkan Gambar 3.7, berikut adalah contoh pemrograman modul mikrokontroler arduino uno pada *node coordinator* yang diprogram pada *software Arduino IDE* :

1. Inisialisasi

Pembuatan skrip ini dimulai dengan menuliskan skrip berikut ini. Skrip ini merupakan proses inisialisasi dan harus ada dalam program yang dibuat pada *node coordinator*, gambar *flowchart 3.7*.

```
// inisialisasi variabel
String indata,simpan1,simpan2,simpan3,simpan4;
float ph,suhu,ph2,suhu2,ph3,suhu3,ph4,suhu4;
char tmpx[100];
String
tanggal,tanggal2,tanggal3,tanggal4,kirim,kirim2,kirim3,kirim4,jam,
jam2,jam3,jam4;
int
counph=0,counshu=0,counph2=0,counshu2=0,counph3=0,counshu3=0,counp
h4=0,counshu4=0;
int x=0;
int datax1,datax2=1,datax3=1,datax4=1,d=0,d2=0,d3=0,d4=0 ;
boolean cek=false, cek2=false, cek3=false, cek4=false;
void setup()
{
    // inisialisasi komunikasi serial
    Serial.begin(9600);
}
```

2. Menerima Data, Penseleksian Data dan Memisahkan Data pH, Suhu, Tanggal dan Waktu

Skrip ini merupakan proses penerimaan data yang masuk dari *node router 1* dan *node router 2*. Setelah data diterima oleh *node coordinator* maka data akan diseleksi apakah data ini berasal dari *node router 1*, *node router 2*, *node router 1* yang melalui *node router 2* atau *node router 2* yang melalui *node router*

1. Setelah itu data pH, suhu, tanggal dan waktu akan dipisahkan, karena data pH dan suhu akan dirata-rata, gambar 3.7.

```
void loop()
{
    while(Serial.available()>0)
    {
        char data_masuk=(char)Serial.read();
        indata += data_masuk;
        //menerima data dan penseleksian data dari R1
        if(data_masuk=='_ && indata[1]=='R' && indata[2]=='1')
        {
            // memisahkan data router 1
            if(datax==1){ x=0; datax++; }
            else if(datax==2)
```

```

        {ph=ph+atof(tmpx); x=0; datax++; counph++;}
    else if(datax==3)
        {suhu=suhu+atof(tmpx); x=0; datax++; counshu++;}
    else if(datax==4)
        {tanggal=tmpx; x=0; datax++; hps();
        }
    else if(datax==5)
        {jam=tmpx; x=0;
        datax=1; hps();
        }
}
//menerima data dan pen seleksian data dari R2
else if(data_masuk=='_' && indata[1]=='R' && indata[2]=='2')
{
    // memisahkan data router 2
    if(datax2==1){ x=0; datax2++; }
    else if(datax2==2)
        {ph2=ph2+atof(tmpx); x=0; datax2++; counph2++;}
    else if(datax2==3)
        {suhu2=suhu2+atof(tmpx); x=0; datax2++; counshu2++;}
    else if(datax2==4)
        {tanggal2=tmpx; x=0; datax2++; hps();
        }
    else if(datax2==5)
        {jam2=tmpx; x=0;
        datax2=1; hps();
        }
}
//menerima data dan pen seleksian data dari R1 yang melalui R2
else if(data_masuk=='_' && indata[1]=='Q' && indata[2]=='1')
{
    //memisahkan data dari R1 yang melalui R2
    if(datax3==1){ x=0; datax3++; }
    else if(datax3==2)
        {ph3=ph3+atof(tmpx); x=0; datax3++; counph3++;}
    else if(datax3==3)
        {suhu3=suhu3+atof(tmpx); x=0; datax3++; counshu3++;}
    else if(datax3==4)
        {tanggal3=tmpx; x=0; datax3++; hps();
        }
    else if(datax3==5)
        {jam3=tmpx; x=0;
        datax3=1; hps();
        }
}
//menerima data dan pen seleksian data dari R2 yang melalui R1
else if(data_masuk=='_' && indata[1]=='Q' && indata[2]=='2')
{
    //memisahkan data dari R2 yang melalui R1
    if(datax4==1){ x=0; datax4++; }
    else if(datax4==2)
        {ph4=ph4+atof(tmpx); x=0; datax4++; counph4++;}
    else if(datax4==3)
        {suhu4=suhu4+atof(tmpx); x=0; datax4++; counshu4++;}
    else if(datax4==4)
        {tanggal4=tmpx; x=0; datax4++; hps();
        }
    else if(datax4==5)
        {jam4=tmpx; x=0;
        datax4=1; hps();
        }
}
}

```

3. Menyeleksi data dan pemberian ID baru

Void baca merupakan fungsi yang berisi skrip untuk menyeleksi data yang masuk dan pemberian ID baru “C”. Karena node ini hanya menerima data dari *node router 1* dan *node router 2* yang masing-masing mempunyai format pesan dengan simbol awal “\$” dan ID “R1”, “R2”, “Q1” dan “Q2”, maka skrip penyeleksian data dan pemberian ID data yang baru dalam hal ini ID “C” sebagai berikut, gambar 3.7.

```
void baca()
{
    int i=0;
    if ((indata[0]=='$') && (indata[1]=='R' ) )
    {
        //pemberian ID baru C
        indata[1]='C';
        if (indata[2]=='1')
        {
            indata=""; // mengkosongkan data
            Serial.print("R1");
            Serial.print(simpan1);
        }
        if (indata[2]=='2')
        {
            indata=""; // mengkosongkan data
            Serial.print("R2");
            Serial.print(simpan3);
        }
    }
    else if ((indata[0]=='$') && (indata[1]=='Q' ))
    {
        // pemberian ID baru C
        indata[1]='C';
        if (indata[2]=='1')
        {
            indata=""; //mengkosongkan data
            Serial.print("R1_melalui_R2");
            Serial.print(simpan2);
        }
        if (indata[2]=='2')
        {
            indata=""; //mengkosongkan data
            Serial.print("R2_melalui_R1");
            Serial.print(simpan4);
        }
    }
}
else
{
    indata="";
}
```

```

    }
}

```

4. Merata-rata Data pH dan suhu serta Pengiriman Data

Proses skrip ini untuk merata-rata pH dan suhu air serta ketentuan untuk mengirimkan data, gambar 3.7.

```

if(data_masuk=='\n')
{
    delay(100);
    baca();
    indata="";
    if(counph==5 && counshu==5)
    {
        // proses rata-rata data dari router 1
        suhu=suhu/counshu; ph=ph/counph;
        kirim="$C1_"+konvrsi(ph)+"_"+konvrsi(suhu)+"_"+
        tanggal+"_"+jam+"\%\n";
        counph=0;
        counshu=0;
        ph=0;
        suhu=0;
        Serial.print(kirim);
        cek=true;
    }
    else if(counph2==5 && counshu2==5)
    {
        // proses rata-rata data dari router 2
        suhu2=suhu2/counshu2; ph2=ph2/counph2;
        kirim2="$C2_"+konvrsi(ph2)+"_"+konvrsi(suhu2)+"_"+
        _+tanggal2+"_"+jam2+"\%\n";
        counph2=0;
        counshu2=0;
        ph2=0;
        suhu2=0;
        Serial.print(kirim2);
        cek2=true;
    }
    else if(counph3==5 && counshu3==5)
    {
        // proses rata-rata data dari router 1 yang melalui router 2
        suhu3=suhu3/counshu3; ph3=ph3/counph3;
        kirim3="$C1_"+konvrsi(ph3)+"_"+konvrsi(suhu3)+"_"+
        _+tanggal3+"_"+jam3+"\%\n";
        counph3=0;
        counshu3=0;
        ph3=0;
        suhu3=0;
        if(cek==false)Serial.println(kirim3);
        else cek=false;
    }
    else if(counph4==5 && counshu4==5)
    {
        // proses rata-rata data dari router 2 yang melalui router 1
        suhu4=suhu4/counshu4; ph4=ph4/counph4;
    }
}

```

```

        kirim4="$C2_" +konvrsi(ph4)+"_"+konvrsi(suhu4)+""
        +"_+tanggal4+"_"+jam4+"%\n";
        counph4=0;
        counshu4=0;
        ph4=0;
        suhu4=0;
        if(cek2==false)Serial.println(kirim4);
        else cek2=false;
    }
}

void hps()
{
    int c;
    for(c=0;c<=40;c++) {tmpx[c]=0; delay(5);}
}

```

Setelah menerima 5 data dari *router* 1 maupun menerima 5 data dari *router* 2, maka selanjutnya *node coordinator* akan melakukan rata-rata data pH dan data suhu dari *router* 1 maupun dari *router* 2. Setelah hasil rata-rata didapatkan maka selanjutnya *node coordinator* akan mengirimkan data ke *end device*.

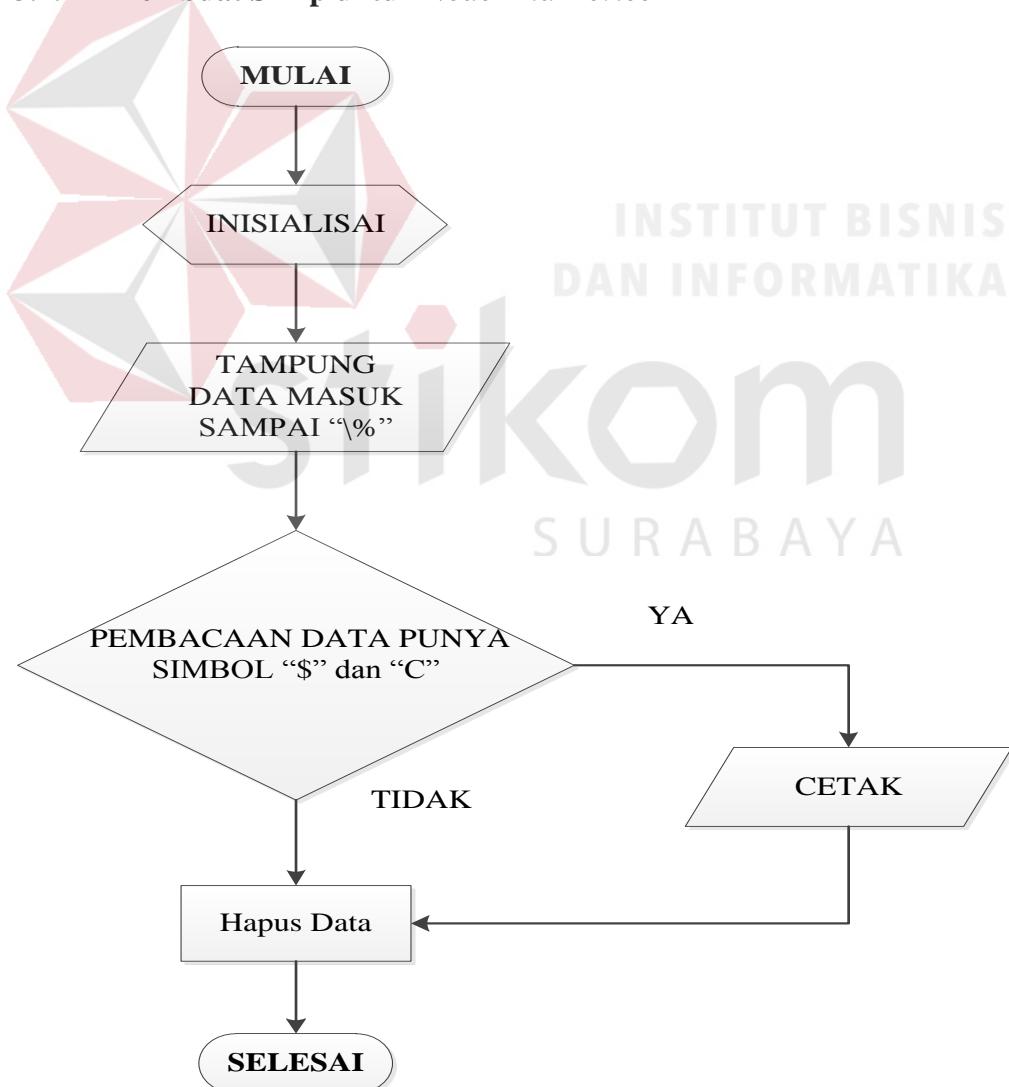
Ketentuan pengiriman data dari *node coordinator* ke *node end device* dibagi menjadi 4 kondisi, sebagai berikut:

1. Setelah *node coordinator* menerima 5 data dari *node router* 1 selanjutnya *coordinator* akan melakukan rata-rata kepada data pH dan data suhu. Setelah nilai rata-rata didapatkan selanjutnya *coordinator* akan mengirimkan hasil rata-rata tersebut ke *node end device*.
2. Setelah *node coordinator* menerima 5 data dari *node router* 2 selanjutnya *coordinator* akan melakukan rata-rata kepada data pH dan data suhu. Setelah nilai rata-rata didapatkan selanjutnya *coordinator* akan mengirimkan hasil rata-rata tersebut ke *node end device*.
3. Setelah *node coordinator* menerima 5 data dari *node router* 1 yang melalui *node router* 2 selanjutnya *coordinator* akan melakukan rata-rata kepada

data pH dan data suhu. Setelah nilai rata-rata didapatkan selanjutnya *coordinator* akan mengirimkan hasil rata-rata tersebut ke *node end device*, apabila kondisi pertama tidak terpenuhi.

4. Setelah *node coordinator* menerima 5 data dari *node router* 2 yang melalui *node router* 1 selanjutnya *coordinator* akan melakukan rata-rata kepada data pH dan data suhu. Setelah nilai rata-rata didapatkan selanjutnya *coordinator* akan mengirimkan hasil rata-rata tersebut ke *node end device*, apabila kondisi kedua tidak terpenuhi.

3.4.4 Membuat Skrip untuk *Node End Device*



Gambar 3.8 Diagram Alir Proses *Node End Device*

Pada alur proses Gambar 3.8 *node end device* akan mengambil data yang memiliki simbol “\$” dari *node coordinator*. Apabila *node end device* menerima data selain itu, maka data akan dihapus atau dibuang.

Pemrograman pada setiap *node* dilakukan pada modul mikrokontrolernya. Pemrograman ini bertujuan untuk memasukkan perintah yang harus dilakukan mikrokontroler itu.

Pada tugas akhir ini, penulis menggunakan arduino uno sebagai mikrokontrolernya. *Software* yang digunakan untuk memprogram arduino tersebut ialah *software* Arduino IDE.

Berikut ini penjabaran dari Gambar 3.7 proses pembuatan skrip pada *node end device*.

1. Inisialisasi

Dalam pembuatan variabel ini penulis menggunakan variabel tipe string yang bernama “indata” untuk menampung data. Pembuatan variabel ini diletakkan diluar fungsi void agar variabel ini dapat digunakan secara global, gambar 3.8.

Berikut sebagai contoh :

```
String indata;
```

2. fungsi void setup

Dalam fungsi void setup perintah akan dibaca 1 kali setelah program berjalan. Dalam tugas akhir ini penulis mengisikan *baudrate* dan variabel indata dalam kondisi kosong, gambar 3.8. Berikut sebagai contoh :

```
void setup()
{
    Serial.begin(9600);
    indata="";
}
```

3. fungsi void loop

Dalam void loop perintah akan dibaca berulang kali. Dalam tugas akhir ini penulis mengisi perintah menunggu data yang masuk (*listening*), dan pengosongan data kembali, gambar 3.8. Berikut contohnya :

```
void loop()
{
    while(Serial.available()>0)
    {
        char data_masuk=(char)Serial.read();
        indata+=data_masuk;

        if(data_masuk == '%')
        {

            if((indata[0] == '$') && (indata[1]=='C'))
            {
                delay(100);
                Serial.println(indata);
            }
            indata="";
        }
        delay(100);
    }
}
```

Dalam perintah di atas arduino diperintah untuk membaca data sampai bertemu tanda “%” (persen). Apabila data telah bertemu tanda seperti itu maka dia akan melakukan penseleksian data. *End device* hanya akan menerima data yang berawalan id “C” selain itu data akan diabaikan. Setelah data diterima maka data akan langsung diteruskan kekomputer. Apabila data tersebut tidak berisikan % dan C, maka data akan dibuang. Setelah semua perintah sudah berjalan perintah terakhir adalah mengosongkan variabel lagi untuk proses pembacaan data selanjutnya.

3.4.5 Pemrograman *visual basic*

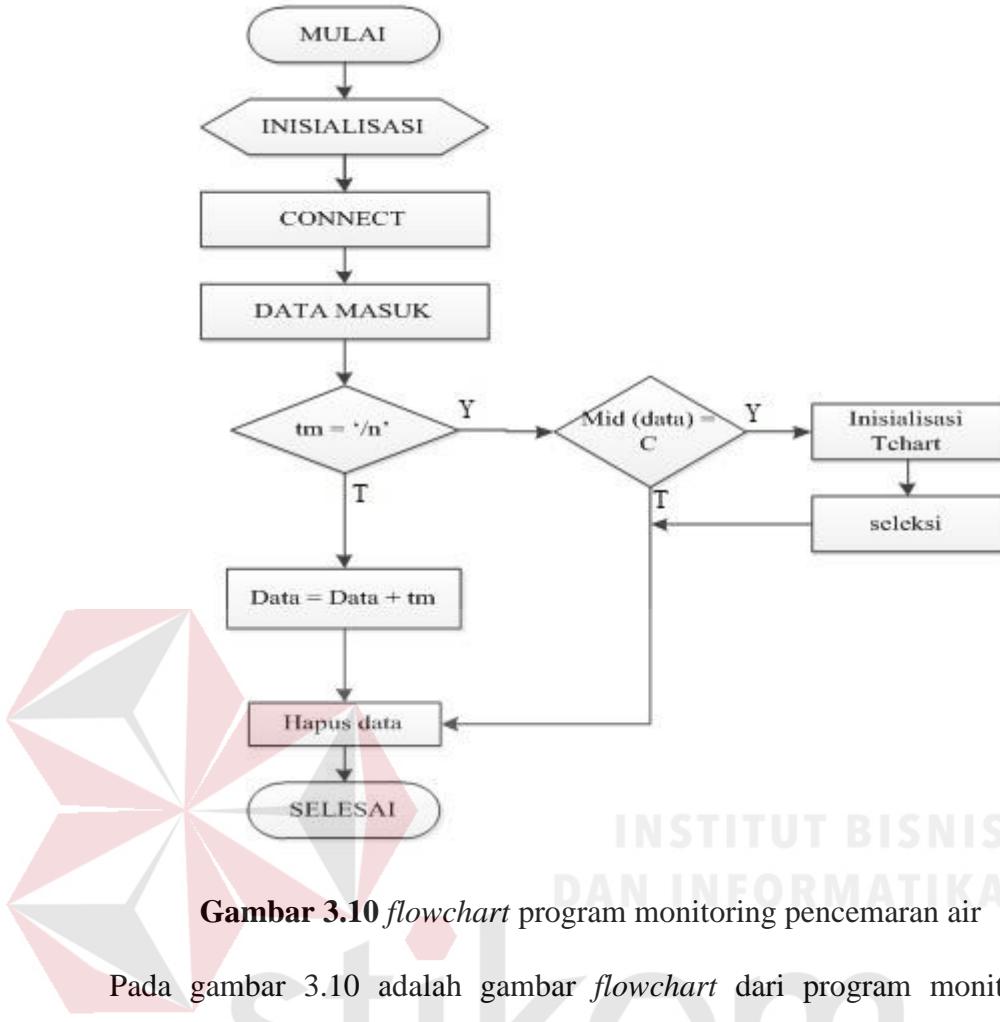
Pada pemrograman ini, komputer akan mengolah data yang dikirimkan oleh *node end device*. Data yang diterima akan diambil isi dari data tersebut yaitu nilai pH, suhu, tanggal dan waktu untuk ditampilkan di layar monitor.



Gambar 3.9 Tampilan *software* dari visual basic

Dalam aplikasi ini memberikan informasi data yang masuk ke dalam komputer, data yang masuk kedalam computer berupa data pH, suhu dan waktu. Langkah awal untuk menjalankan aplikasi ini adalah dengan melakukan pemilihan port serial (COM) yang terdeteksi sebagai port *node end device* sebelum menekan tombol “Connect”. Ketika menekan tombol “Connect” namun port serial belum dipilih maka akan muncul kotak pengingat bahwa port belum dipilih.

Setelah langkah awal telah terpenuhi maka *software* akan mulai berjalan dan mencari data yang masuk. Berikut alur program yang dibuat penulis.



Pada gambar 3.10 adalah gambar *flowchart* dari program monitoring pencemaran air sungai, keterangannya sebagai berikut.

a. Penseleksi data

Secara *detail* data yang diterima oleh komputer adalah data yang diteruskan oleh *node end device* yang diterima oleh *node coordinator*. Jadi data yang didalamnya masih terkandung isi yang tidak dibutuhkan seperti *head*, tanda garis bawah, id *node* dan *tail* akan dihapus.

Contoh pesan yang diterima dari *node end device* tersebut seperti ini (\$C1_7_35_.13/06/2014_02:30:50.%). Maka data yang diterima perlu diambil yaitu nilai pH (nilainya : 7), suhu (nilainya : 35), tanggal (13/06/2014) dan waktu

(02:30:50). Untuk id *node* itu nantinya untuk pemisah dari mana asal data itu berasal. Bila memiliki C1 artinya pesan yang diteruskan *node coordinator* dari *node* sensor 1 akan ditampilkan pada layar *router* 1 dan apabila 2 artinya pesan yang diteruskan *node coordinator* dari *node* sensor 2 akan ditampilkan pada layar *router* 2.

b. Tampilan grafik

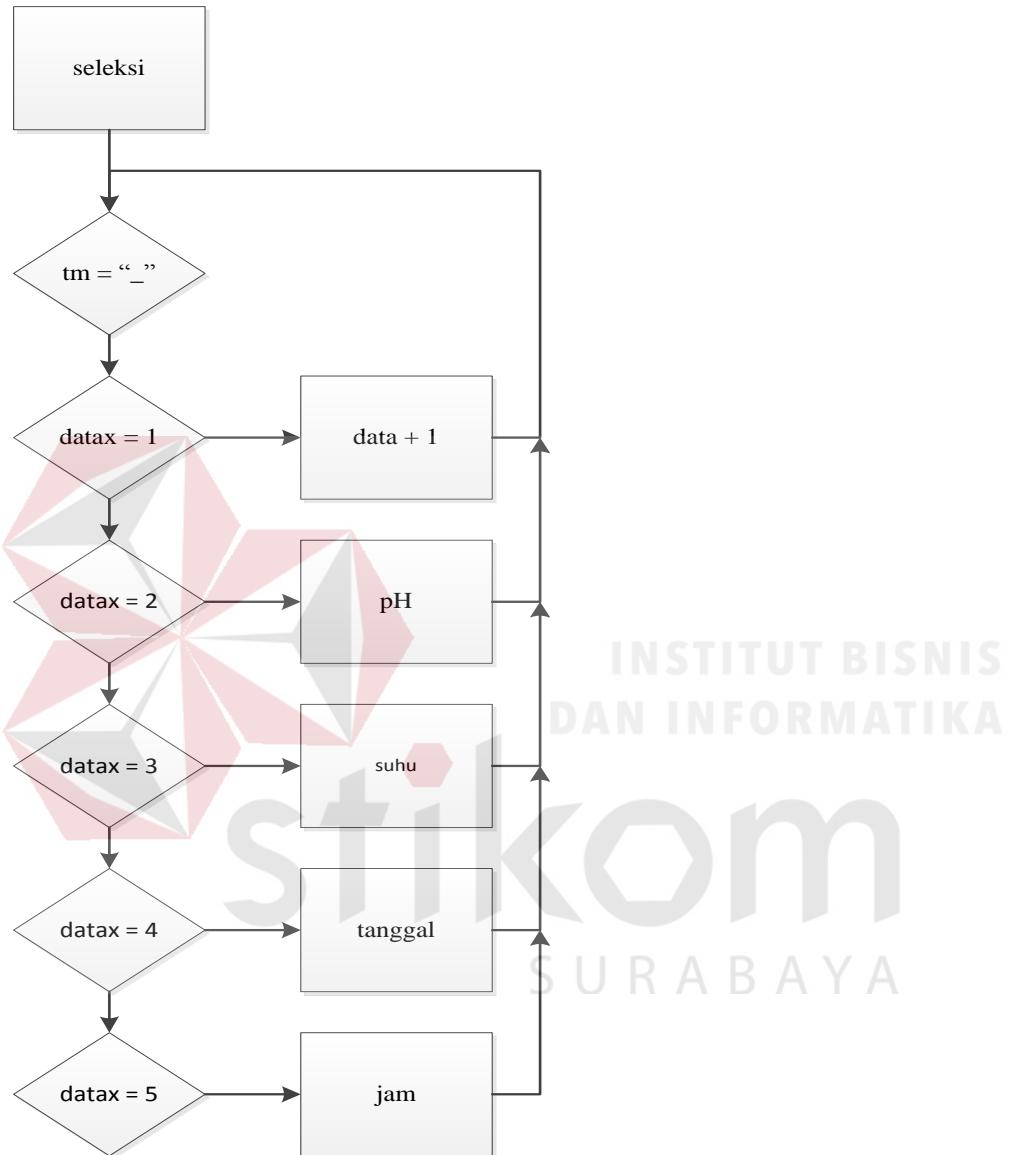
Tampilan grafik ini digunakan untuk mempresentasikan data pH, suhu dan waktu yang diperoleh dari data yang dikirimkan *node coordinator* ke *node end device*. Sumbu x dalam grafik ini mempresentasikan data ketinggian dan sumbu y mepresentasikan waktu.

c. Langkah terakhir

Dalam langkah terakhir ini aplikasi akan menghapus semua data yang ditampung. Karena aplikasi ini melakukan proses berulang-ulang, penghapusan data tampung disini bertujuan agar data yang berhasil ditampung tidak mengganggu proses pembacaan data sesudahnya.

Didalam proses seleksi terdapat 2 proses,

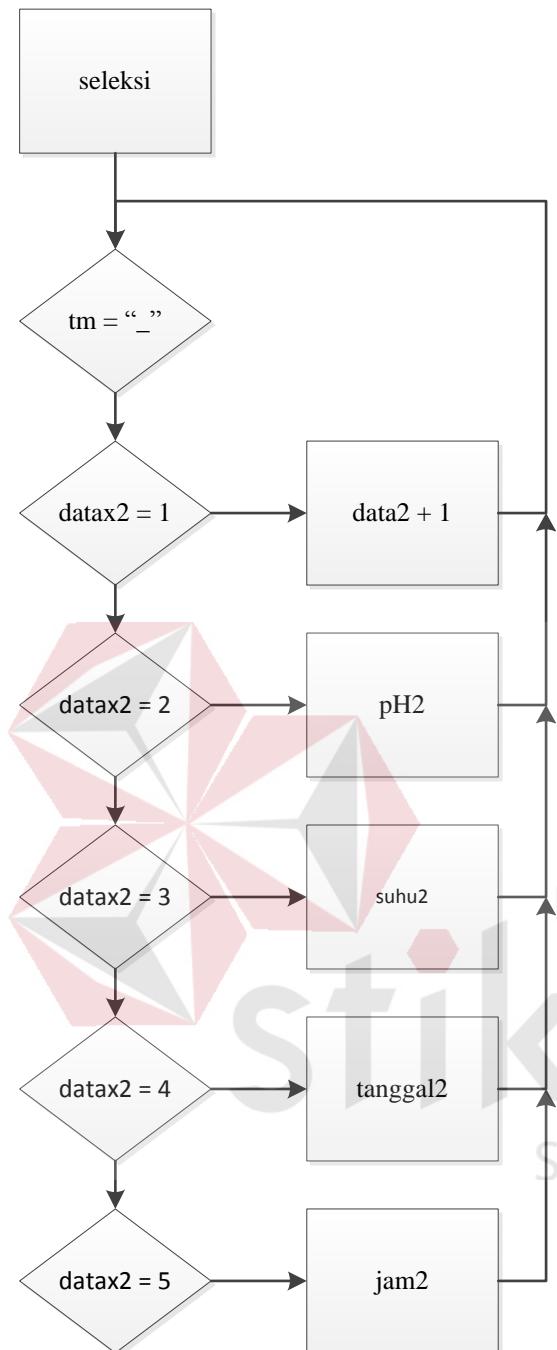
1. Proses penerimaan dari data yang ber ID C1



Gambar 3.11 *flowchart* penerimaan dari data yang ber ID C1

Apabila data yang ber ID berawalan C1 masuk ke seleksi maka datax = 1 bernilai data + 1, datax = 2 bernilai pH, datax = 3 bernilai suhu, datax = 4 bernilai tanggal, datax = 5 bernilai jam.

2. Proses penerimaan dari data yang ber ID C2



Gambar 3.12 *flowchart* penerimaan dari data yang ber ID C2

Apabila data yang ber ID berawalan C1 masuk ke seleksi maka datax2 = 1 bernilai data2 + 1, datax2 = 2 bernilai pH2, datax2 = 3 bernilai suhu2, datax2 = 4 bernilai tanggal2, datax2 = 5 bernilai jam2.

BAB IV

HASIL PENGUJIAN DAN PENGAMATAN

Pengujian sistem yang dilakukan penulis merupakan pengujian terhadap perangkat keras dan perangkat lunak secara keseluruhan dan digunakan untuk mengetahui apakah sistem dapat berjalan sesuai yang diharapkan.

4.1 Kebutuhan *Hardware* dan *Software*

Untuk dapat menguji sistem ini, diperlukan perangkat keras dan perangkat lunak.

4.1.1 Kebutuhan Perangkat Keras (*Hardware*)

Penulis membutuhkan perangkat keras sebagai berikut:

1. Komputer atau laptop
2. Arduino uno SMD R3
3. Sensor Analog pH meter kit
4. Sensor suhu DS18B20
5. *Real time Clock* (RTC) modul
6. Xbee series 2 dengan Xbee shield

4.1.2 Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pengujian sistem sebagai berikut:

1. *Software arduino IDE*
2. *Software Visual Basic*

4.2 Pengujian Sistem

Pengujian sistem yang dilakukan penulis merupakan pengujian terhadap perangkat keras dan perangkat lunak secara keseluruhan dan digunakan untuk mengetahui apakah sistem dapat berjalan sesuai yang diharapkan.

4.3 Pengujian Sensor pH, sensor Suhu, Modul RTC dan Visual Basic

Pengujian sensor pH, sensor suhu dan modul RTC dilakukan untuk mengetahui informasi kadar pH dan suhu air beserta waktu disetiap *node* pada prototipe. Terdapat dua *node* sensor yang diletakan pada tempat atau kondisi yang berbeda pada saat pengukuran. Yang dimaksud kondisi yang berbeda adalah kondisi dari kadar pH yang akan diuji, pada saat pengujian penulis menggunakan 3 kadar pH yang berbeda nilainya, yaitu netral, asam dan basa. Data yang dihasilkan oleh sensor dan RTC diolah oleh arduino supaya dapat menghasilkan nilai pH, suhu dan waktu pada saat data pH dan air itu diambil. Pengujian visual basic untuk mengetahui apakah aplikasi visual basic dapat mengolah data masuk dan menampilkan grafik.

4.3.1 Peralatan

Peralatan yang dibutuhkan pada setiap *node* dalam pengujian ini adalah sebagai berikut:

1. Arduino uno
2. Sensor Analog pH meter kit
3. Sensor suhu DS18B20
4. Modul RTC

5. Komputer atau laptop
6. Kabel USB
7. *Software arduino IDE*
8. *Software Visual Basic*
9. pH meter
10. *Thermometer*

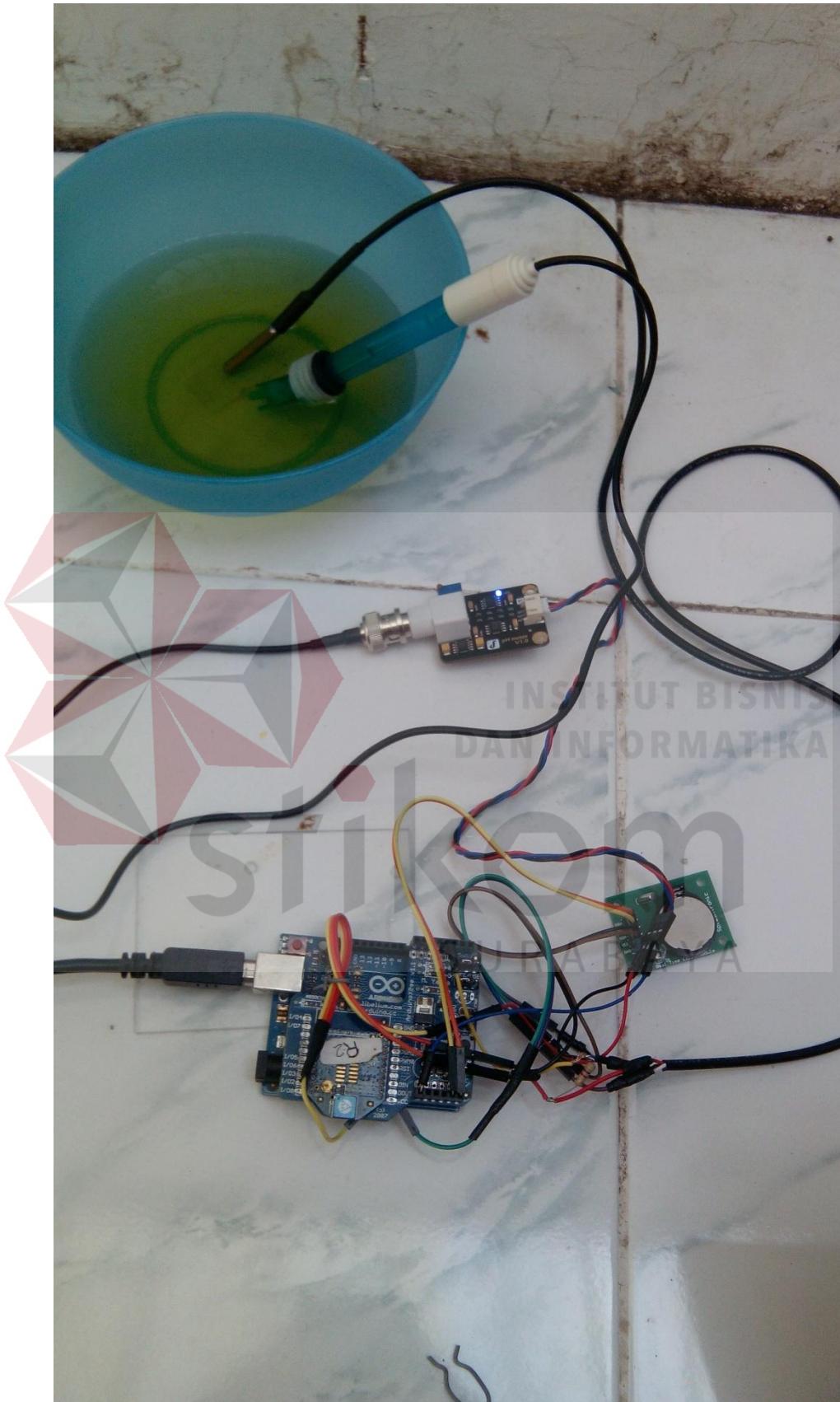
4.3.2 Prosedur Pengujian

Langkah – langkah dalam prosedur pengujian ini adalah sebagai berikut:

1. Hubungkan sensor analog pH meter kit, sensor suhu DS18B20 dan modul RTC ke arduino uno menggunakan kabel pada pin yang telah ditentukan.
2. Hubungkan arduino uno (*node router*), (*node coordinator*), (*node end device*) pada komputer menggunakan kabel USB.
3. Aktifkan komputer atau laptop dan jalankan *software arduino IDE*.
4. *Upload* skrip yang telah dibuat ke dalam arduino uno (*node router*) yang digunakan untuk membaca *output* sensor dan modul RTC. *Upload* juga skrip yang telah dibuat ke dalam (*node coordinator*) yang digunakan untuk menerima data yang dikirim oleh *node router*. Serta *Upload* juga skrip yang telah dibuat ke dalam (*node end device*) yang digunakan untuk menerima data yang dikirim oleh *node coordinator*.
5. Buka serial monitor pada *software arduino IDE* untuk melihat hasil *output* dari sensor dan RTC.
6. Buka aplikasi monitoring dari visual basic
7. Amati dan bandingkan *output* sensor di visual basic dengan pengukuran manual menggunakan pH meter kit dan *Thermometer*.

4.3.3 Hasil Pengujian

Pada pengujian ini sensor pH dan sensor suhu digunakan untuk mendapatkan kadar pH dan suhu air pada prototipe. Selain itu juga melakukan pengukuran pH dan suhu air secara manual menggunakan pH meter dan *Thermometer*. Proses dimulai dari masing-masing sensor yang mengirimkan *output* menuju *router 1* ataupun menuju *router 2*, kemudian *router* menambahkan simbol yang berfungsi sebagai *id(data)* di masing-masing *router* dan menggabungkan keempat data tersebut yaitu data pH, suhu, data tanggal dan jam. Setelah proses pemberian id dan penggabungan data maka *router* mengirimkan data tersebut ke *coordinator* dan *router* lainnya. Setelah *coordinator* mendapatkan 5 data masing-masing dari *router 1* dan *router 2* selanjutnya *coordinator* akan melakukan rata-rata dari data yang diterima dari *router 1* dan *router 2*. Setelah hasil rata-rata didapat selanjutnya *coordinator* akan meneruskan data ke *node end device*. Setelah data diterima oleh *end device*, selanjutnya *end device* akan mengirimkan datanya pada software visual basic. Data yang diterima software visual basic akan ditampilkan agar *user* dapat mengetahui informasi yang dikirim oleh *node router 1* maupun *node router 2*. Pengujian dapat dilakukan seperti Gambar 4.1.



Gambar 4.1 Pengujian Sensor ph dan Sensor Suhu

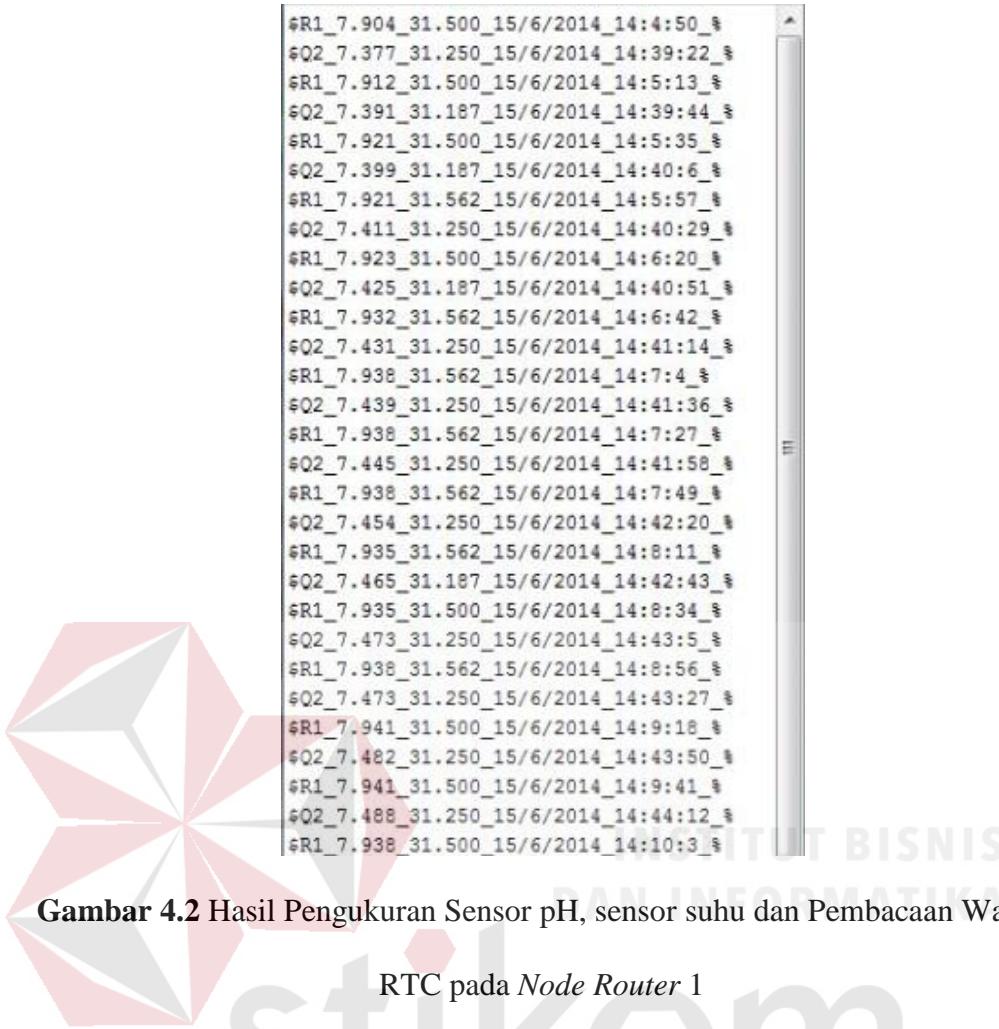
Untuk mendapatkan hasil pengukuran dari sensor digunakan skrip berikut ini pada setiap *node router*:

1. *Node router 1*

```
void loop()
{
    if (ok==0)
    {
        delay(5000);
        ok=1;
        dtal1=amb1data();
        Serial.print(dtal1);
    }
    while(Serial.available()>0)
    {
        char data_masuk=(char)Serial.read();
        id += data_masuk;
        if(data_masuk =='\n')
        {
            if((id[0]=='$') && (id[1]=='R') && (id[2]=='2'))
            {
                id[1]='Q'; // perubahan ID "R" menjadi "Q"
                delay(5000);
                Serial.print(id);
                id="";
                ok=0;
            }
            else
            {
                id="";
            }
        }
    }
}
```

Skrip atau teks yang tercetak tebal merupakan program yang dilakukan secara berulang – ulang. Skrip tersebut akan membaca data yang masuk melalui komunikasi nirkabel, jika *node router 1* kondisi “ok” = 0 maka akan didelay selama 5 detik setelah itu status “ok” akan menjadi 1, apabila ok=1 maka *router 1* akan mengirimkan data ke *coordinator* dan *router 2*. Setelah itu apabila *router 1* menerima pesan “\$R2” yang dimana pesan tersebut merupakan kiriman data dari *router 2* ID “R” akan dirubah menjadi “Q”, maka setelah delay 5000 ms atau 5 detik, *node router 1* akan meneruskan data yang dikirim *router 2* ke *node coordinator*. Membaca kadar pH air, suhu air dan

waktu pada prototipe dan mengirimkan data tersebut secara *broadcast* ke *coordinator* dan *node router* 2. Fungsi ph () dan fungsi getTempSuhu () merupakan skrip pembacaan kadar pH air dan suhu air yang dilakukan oleh sensor dan pembacaan waktu yang dilakukan oleh modul RTC yang kemudian data tersebut dikirimkan secara *broadcast* oleh xbee *series* 2. Data pH air, suhu air dan waktu pada *node router* 1 memiliki ID “R1”. Jika *node router* 1 menerima pesan “%R” yang mana pesan data tersebut dari *node router* 2, maka dengan delay 5000 ms atau 5 detik, *node router* 1 membaca data dari *node router* 2 yang kemudian memforwardnya ke *node coordinator*. Pembacaan data dari *node router* 2 dilakukan pada skrip void loop (). Didalam void loop apabila jika ada data atau pesan yang diterima mempunyai awalan “%R2”, maka karakter “R” akan diubah menjadi “Q”. Gambar 4.2 menunjukkan hasil pengukuran pH air oleh sensor analog pH meter kit, suhu air oleh sensor DS18S20 dan waktu oleh RTC pada *node router* 1.



```

$R1_7.904_31.500_15/6/2014_14:4:50_%
$Q2_7.377_31.250_15/6/2014_14:39:22_%
$R1_7.912_31.500_15/6/2014_14:5:13_%
$Q2_7.391_31.187_15/6/2014_14:39:44_%
$R1_7.921_31.500_15/6/2014_14:5:35_%
$Q2_7.399_31.187_15/6/2014_14:40:6_%
$R1_7.921_31.562_15/6/2014_14:5:57_%
$Q2_7.411_31.250_15/6/2014_14:40:29_%
$R1_7.923_31.500_15/6/2014_14:6:20_%
$Q2_7.425_31.187_15/6/2014_14:40:51_%
$R1_7.932_31.562_15/6/2014_14:6:42_%
$Q2_7.431_31.250_15/6/2014_14:41:14_%
$R1_7.938_31.562_15/6/2014_14:7:4_%
$Q2_7.439_31.250_15/6/2014_14:41:36_%
$R1_7.938_31.562_15/6/2014_14:7:27_%
$Q2_7.445_31.250_15/6/2014_14:41:58_%
$R1_7.938_31.562_15/6/2014_14:7:49_%
$Q2_7.454_31.250_15/6/2014_14:42:20_%
$R1_7.935_31.562_15/6/2014_14:8:11_%
$Q2_7.465_31.187_15/6/2014_14:42:43_%
$R1_7.935_31.500_15/6/2014_14:8:34_%
$Q2_7.473_31.250_15/6/2014_14:43:5_%
$R1_7.938_31.562_15/6/2014_14:8:56_%
$Q2_7.473_31.250_15/6/2014_14:43:27_%
$R1_7.941_31.500_15/6/2014_14:9:18_%
$Q2_7.482_31.250_15/6/2014_14:43:50_%
$R1_7.941_31.500_15/6/2014_14:9:41_%
$Q2_7.488_31.250_15/6/2014_14:44:12_%
$R1_7.938_31.500_15/6/2014_14:10:3_%

```

Gambar 4.2 Hasil Pengukuran Sensor pH, sensor suhu dan Pembacaan Waktu

RTC pada *Node Router 1*

Format data pada *node router 1* sebagai berikut

\$R1_7.904_31.500_15/6/2014_14:4:50_% dimana \$ adalah *header* data, R1 adalah ID untuk *node router 1*, 7.904 adalah kadar pH air, 31.500 adalah suhu air, 15/6/2014 adalah tanggal, 14:4:50 adalah waktu, dan % adalah *tail* dari data.

2. *Node Router 2*

Skrip yang digunakan pada *node router 2* hampir sama dengan *node router 1*, perbedaannya hanya terletak ID . ID yang diberikan untuk *node router 2* adalah “R2” dan perubahan ID *node router 1* yang melewati *node router 2* adalah “Q1” yang akan dikirim ke *node coordinator*.

```
void loop()
{
    if ( ok==0)
    {
        delay(5000);
        ok=1;
        dtal1=ambldata();
        Serial.print(dtal1);
    }
    while(Serial.available()>0)
    {
        char data_masuk=(char)Serial.read();
        id += data_masuk;
        if(data_masuk =='\n')
        {
            if((id[0]=='$') && (id[1]=='R') && (id[2]=='1'))
            {
                id[1]='Q';
                delay(5000);
                Serial.print(id);
                id="";
                ok=0;
            }
            else
            {
                id="";
            }
        }
    }
}
```

Gambar 4.3 menunjukkan hasil pengukuran pH air oleh sensor analog pH meter kit, suhu air oleh sensor DS18S20 dan waktu oleh RTC pada *node router* 2.



```

$Q1_7.904_31.500_15/6/2014_14:4:50_%
$R2_7.377_31.250_15/6/2014_14:39:22_%
$Q1_7.912_31.500_15/6/2014_14:5:13_%
$R2_7.391_31.187_15/6/2014_14:39:44_%
$Q1_7.921_31.500_15/6/2014_14:5:35_%
$R2_7.399_31.187_15/6/2014_14:40:6_%
$Q1_7.921_31.562_15/6/2014_14:5:57_%
$R2_7.411_31.250_15/6/2014_14:40:29_%
$Q1_7.923_31.500_15/6/2014_14:6:20_%
$R2_7.425_31.187_15/6/2014_14:40:51_%
$Q1_7.932_31.562_15/6/2014_14:6:42_%
$R2_7.431_31.250_15/6/2014_14:41:14_%
$Q1_7.938_31.562_15/6/2014_14:7:4_%
$R2_7.439_31.250_15/6/2014_14:41:36_%
$Q1_7.938_31.562_15/6/2014_14:7:27_%
$R2_7.445_31.250_15/6/2014_14:41:58_%
$Q1_7.938_31.562_15/6/2014_14:7:49_%
$R2_7.454_31.250_15/6/2014_14:42:20_%
$Q1_7.935_31.562_15/6/2014_14:8:11_%
$R2_7.465_31.187_15/6/2014_14:42:43_%
$Q1_7.935_31.500_15/6/2014_14:8:34_%
$R2_7.473_31.250_15/6/2014_14:43:5_%
$Q1_7.938_31.562_15/6/2014_14:8:56_%
$R2_7.473_31.250_15/6/2014_14:43:27_%
$Q1_7.941_31.500_15/6/2014_14:9:18_%
$R2_7.482_31.250_15/6/2014_14:43:50_%
$Q1_7.941_31.500_15/6/2014_14:9:41_%
$R2_7.488_31.250_15/6/2014_14:44:12_%
$Q1_7.938_31.500_15/6/2014_14:10:3_%

```

Gambar 4.3 Hasil Pengukuran Sensor pH, sensor suhu dan Pembacaan Waktu

RTC pada *Node Router 2*

Format data pada *node router 2* sebagai berikut

\$R2_7.377_31.250_15/6/2014_14:39:22_% dimana \$ adalah *header* data, R2 adalah ID untuk *node router 2*, 7.377 adalah kadar pH air, 31.250 adalah suhu air, 15/6/2014 adalah tanggal, 14:39:22 adalah waktu, dan % adalah *tail* dari data.

Dari hasil pengujian sensor pH meter kit dan sensor DS18S20 pada *node router 1* dan *2* dapat diketahui bahwa pengukuran pH air menggunakan sensor pH meter kit dan pengukuran suhu air menggunakan sensor DS18S20 terdapat sedikit perbedaan dengan pengukuran secara manual menggunakan pH meter dan *Thermometer*. Hasil persentase kesalahan dapat dihitung dengan rumus :

$$error(\%) = \frac{(PembacaanSensor - AlatUkurSebenarnya) \times 100}{PembacaanSensor}$$

Tabel 4.1 merupakan selisih hasil pengujian sensor pH meter kit dan sensor DS18S20 pada *node router* 1 dan Tabel 4.2 merupakan selisih hasil pengujian sensor pH meter kit dan sensor DS18S20 pada *node router* 2 dengan pengujian menggunakan ph meter dan *Thermometer*.

(Kondisi Netral)

Tabel 4.1 Hasil Pengujian Sensor pH, sensor suhu dan Alat Ukur pada *router* 1

No	Sensor pH	Alat Ukur	Selisih	Sensor suhu	Alat Ukur	Selisih
1	7,28	7,27	0,01	31,04 °C	30 °C	1,04
2	7,12	7,10	0,02	31,10 °C	31 °C	0,10
3	7,23	7,20	0,03	30,01 °C	30 °C	0,01
4	7,10	7,05	0,05	30,65 °C	31 °C	0,35
5	7,55	7,50	0,05	30,87 °C	31 °C	0,13
6	7,31	7,29	0,02	30,43 °C	30 °C	0,43
7	7,10	7,12	0,02	31,24 °C	31 °C	0,24
8	7,01	7,11	0,10	31,56 °C	31 °C	0,56
9	7,66	7,50	0,16	31,86 °C	32 °C	0,14
10	7,45	7,40	0,05	30,12 °C	30 °C	0,12

(Kondisi Asam)

Tabel 4.2 Hasil Pengujian Sensor pH, sensor suhu dan Alat Ukur pada *router* 2

No	Sensor pH	Alat Ukur	Selisih	Sensor suhu	Alat Ukur	Selisih
1	2,34	2,21	0,13	31,12 °C	30 °C	1,12
2	1,53	1,25	0,28	32,31 °C	32 °C	0,31
3	1,34	1,32	0,02	30,05 °C	30 °C	0,05
4	1,67	1,89	0,22	31,51 °C	32 °C	0,49
5	2,21	2,43	0,22	30,12 °C	30 °C	0,12
6	2,21	2,11	0,1	31,12 °C	31 °C	0,12
7	1,78	1,98	0,2	31,10 °C	30 °C	1,1
8	2,56	2,50	0,06	30,12 °C	30 °C	0,12
9	2,11	2,33	0,22	30,81 °C	31 °C	0,81
10	2,14	2,21	0,07	31,02 °C	31 °C	0,02

(Kondisi Basa)

Tabel 4.3 Hasil Pengujian Sensor pH, sensor suhu dan Alat Ukur pada *router* 1

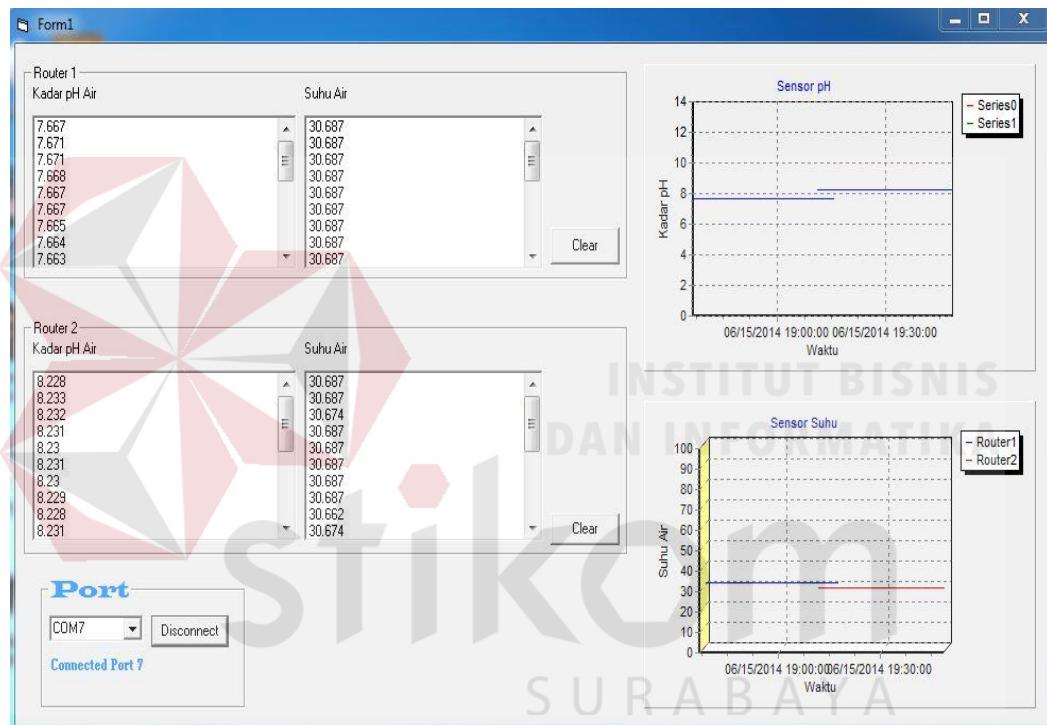
No	Sensor pH	Alat Ukur	Selisih	Sensor suhu	Alat Ukur	Selisih
1	9,69	9,65	0,04	30,18 °C	30 °C	0,18
2	9,91	9,87	0,04	29,98 °C	29 °C	0,98
3	9,74	9,77	0,03	30,12 °C	30 °C	0,12
4	9,57	9,60	0,03	31,01 °C	30 °C	1,01
5	9,85	9,79	0,06	30,42 °C	30 °C	0,42

Berdasarkan hasil pengujian pada Tabel 4.1 dapat diketahui bahwa pengukuran pH air menggunakan sensor pH meter kit pada prototipe ini mempunyai rata-rata persentase kesalahan 0,37% dan pengukuran suhu air menggunakan sensor DS18S20 pada prototipe ini mempunyai rata-rata persentasi kesalahan 0,608% pada *node router* 1 dan pada Tabel 4.2 dapat diketahui bahwa pengukuran pH air menggunakan sensor pH meter kit pada prototipe ini mempunyai rata-rata persentase kesalahan 1,86% dan pengukuran suhu air menggunakan sensor DS18S20 pada prototipe ini mempunyai rata-rata persentasi kesalahan 0,73% pada *node router* 2. Dalam penentuan nilai persentase kesalahan pembacaan sensor pH meter kit dan DS18S20 dengan rumus dianggap mutlak bernilai positif walaupun dalam hasil perhitungan bernilai negatif karena dalam perhitungan ini penulis hanya ingin mengetahui selisih nilai pembacaan sensor ph meter kit dan DS18S20 dengan pengukuran secara manual menggunakan pH meter dan *Thermometer*. Persentase kesalahan masih dapat ditoleransi untuk sebuah sistem yang bekerja secara cepat dan *real time*, yang berarti bahwa hasil pengukuran baik menggunakan sensor ataupun alat ukur sebenarnya tidak terdapat perbedaan yang terlalu jauh, sehingga sistem yang bekerja secara cepat dan *real*

time ini masih bisa mentoleransi persentasi kesalahan dari sensor dan alat ukur sebenarnya.

3. Visual Basic

Dalam aplikasi ini *user* akan mengetahui informasi data yang masuk ke dalam komputer, data yang masuk kedalam komputer berupa data pH, suhu dan waktu.



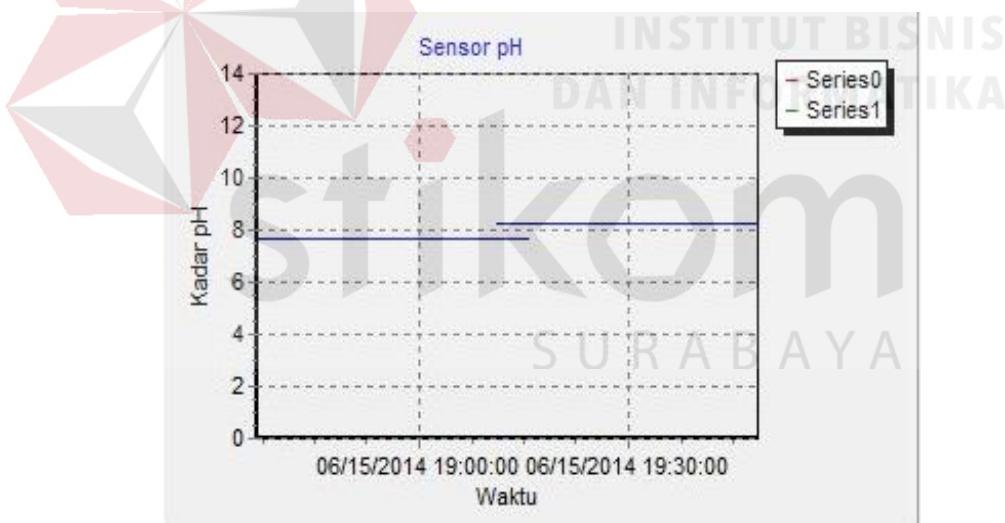
Gambar 4.4 Hasil tampilan software *visual basic*

Data yang masuk juga akan ditampilkan dalam grafik. Berikut grafik yang ditunjukkan dari data yang masuk pada Gambar 4.5

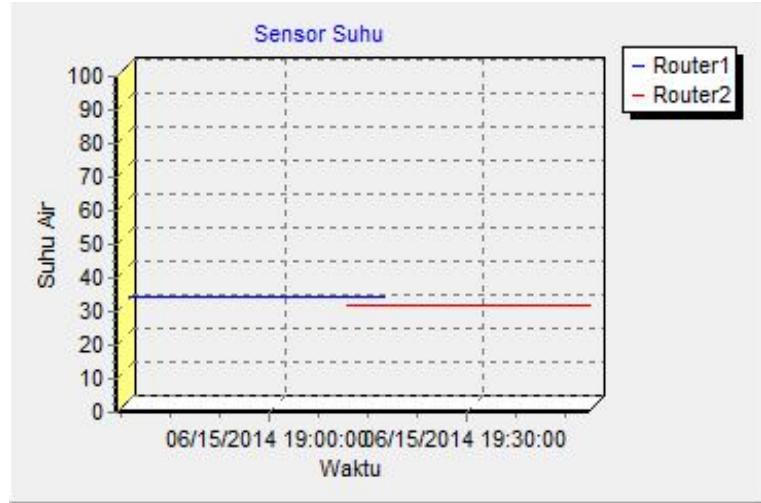
Router 1	
Kadar pH Air	Suhu Air
7.667	30.687
7.671	30.687
7.671	30.687
7.668	30.687
7.667	30.687
7.667	30.687
7.665	30.687
7.664	30.687
7.663	30.687

Router 2	
Kadar pH Air	Suhu Air
8.228	30.687
8.233	30.687
8.232	30.674
8.231	30.687
8.23	30.687
8.231	30.687
8.23	30.687
8.229	30.687
8.228	30.662
8.231	30.674

Gambar 4.5 Data masuk yang diterima



Gambar 4.6 Data pH dari node router 1 dan node router 2



Gambar 4.7 Data suhu dari *node router 1* dan *node router 2*

4.4 Pengujian Protokol Komunikasi

Pengujian protokol komunikasi ini bertujuan untuk mengetahui apakah pengiriman data dari setiap *node* ke *node* lainnya yang dituju benar dan diterima dengan baik oleh sisi penerima.

4.4.1 Peralatan

Peralatan yang digunakan pada setiap *node* dalam pengujian ini adalah sebagai berikut:

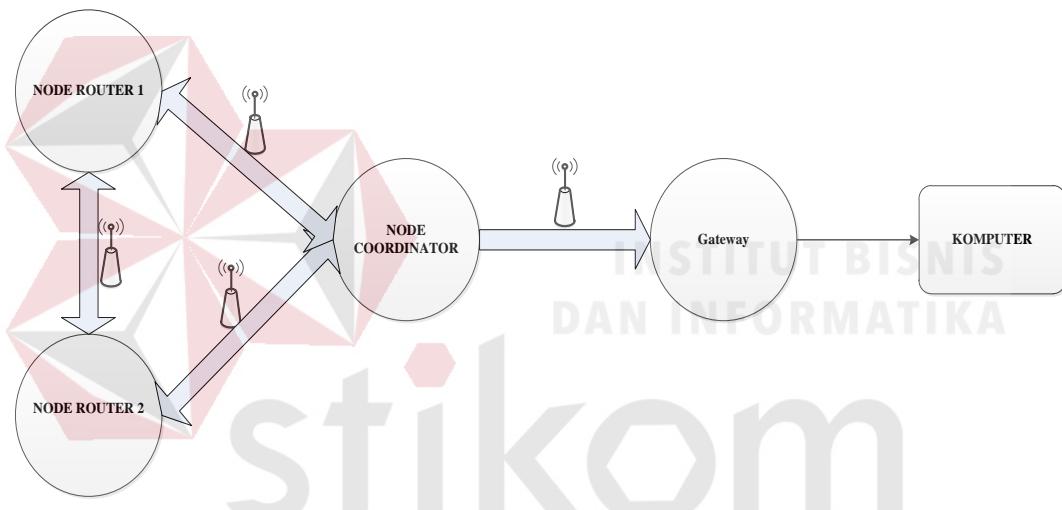
1. Arduino uno dan Xbee *series 2* beserta shield
2. Sensor pH meter kit, sensor DS18S20 dan modul RTC (hanya digunakan pada *node router 1* dan *router 2*)
3. Komputer atau laptop
4. Kabel USB
5. *Software arduino IDE*

4.4.2 Prosedur Pengujian

Langkah – langkah dalam prosedur pengujian ini adalah sebagai berikut:

1. Hubungkan setiap *node* pada komputer menggunakan kabel USB
2. Aktifkan komputer dan jalankan *software arduino IDE*
3. *Upload* masing-masing skrip yang digunakan untuk setiap *nodenya*
4. Buka serial monitor pada *software arduino IDE* untuk melihat data yang dikirim dan diterima di setiap *node*
5. Amati apakah ada data yang *crash* atau bertabrakan sehingga tidak diterima atau sampai pada *node* yang dituju.

4.4.3 Hasil Pengujian



Gambar 4.8 Komunikasi dalam Sistem WSN Monitoring Pencemaran Air Sungai

Gambar 4.8 merupakan komunikasi yang terjadi dalam sistem WSN untuk monitoring pencemaran air sungai. Terdapat beberapa tahapan proses komunikasi yang terjadi pada setiap *node*.

Proses pertama adalah *node coordinator* akan menerima dan menseleksi data yang diterima dari *node router 1* dan *2*. Berikut cuplikan skrip pada *node coordinator*:

```
void loop()
{
    while(Serial.available() > 0)
```

```

{
    char data_masuk=(char)Serial.read();
    indata += data_masuk;
if(data_masuk=='_' && ((indata[1]=='R' && indata[2]=='1') ||
(indata[0]=='R' && indata[1]=='1') || (indata[2]=='R' &&
indata[3]=='1')))

    {   // memisahkan data router 1
        if(datax==1){ x=0; datax++; }
        else if(datax==2)
            {ph=ph+atof(tmpx); x=0; datax++; counph++; }
        else if(datax==3)
            {suhu=suhu+atof(tmpx); x=0; datax++; counshu++; }
        else if(datax==4)
            {tanggal=tmpx; x=0; datax++; hps();
            }
        else if(datax==5)
            { jam=tmpx; x=0;
            datax=1; hps();
            }
    }

//menerima data dan penseleksian data dari R2
else if(data_masuk=='_' && indata[1]=='R' && indata[2]=='2')
{
    // memisahkan data router 2
    if(datax2==1){ x=0; datax2++; }
    else if(datax2==2)
        {ph2=ph2+atof(tmpx); x=0; datax2++; counph2++; }
    else if(datax2==3)
        {suhu2=suhu2+atof(tmpx); x=0; datax2++; counshu2++; }
    else if(datax2==4)
        {tanggal2=tmpx; x=0; datax2++; hps(); }
    else if(datax2==5)
        { jam2=tmpx; x=0;
        datax2=1; hps();
        }
}

//menerima data dan penseleksian data dari R1 yang melalui R2
else if(data_masuk=='_' && indata[1]=='Q' && indata[2]=='1')
{
    //memisahkan data dari R1 yang melalui R2
    if(datax3==1){ x=0; datax3++; }
    else if(datax3==2)
        {ph3=ph3+atof(tmpx); x=0; datax3++; counph3++; }
    else if(datax3==3)
        {suhu3=suhu3+atof(tmpx); x=0; datax3++; counshu3++; }
    else if(datax3==4)
        {tanggal3=tmpx; x=0; datax3++; hps(); }
    else if(datax3==5)
        { jam3=tmpx; x=0;
        datax3=1; hps();
        }
}

//menerima data dan penseleksian data dari R2 yang melalui R1
else if(data_masuk=='_' && indata[1]=='Q' && indata[2]=='2')
{
    //memisahkan data dari R2 yang melalui R1
    if(datax4==1){ x=0; datax4++; }
    else if(datax4==2)
        {ph4=ph4+atof(tmpx); x=0; datax4++; counph4++; }
    else if(datax4==3)
        {suhu4=suhu4+atof(tmpx); x=0; datax4++; counshu4++; }
}

```

```

        else if(datax4==4)
            {tanggall4=tmpx; x=0; datax4++; hps();
        }
        else if(datax4==5)
            { jam4=tmpx; x=0;
            datax4=1; hps();
        }
    }
}

```

Proses kedua, jika semua data yang dikirimkan oleh *node router* 1 dan 2 telah diterima pada *node coordinator*, maka ID semua data tersebut akan diubah menjadi “C1” dan “C2”. Selanjutnya *node coordinator* akan melakukan proses rata-rata kepada data pH dan data suhu yang telah diterima dari *router* 1 atau *router* 2 langsung, maupun dari *router* 1 yang melalui *router* 2 atau dari *router* 2 yang melalui *router* 1. Setelah hasil rata-rata didapatkan maka selanjutnya *node coordinator* akan mengirimkannya keapda *node end device*. Berikut cuplikan skrip atau program untuk mengubah ID semua data dan setelah itu merata-rata data pH dan suhu lalu mengirimkannya kepada *node end device*:

```

void baca()
{
    int i=0;
    if ((indata[0]=='$') && (indata[1]=='R'))
    {
        //pemberian ID baru C
        indata[1]='C';
        if (indata[2]=='1')
        {
            simpan1=indata;
            indata="";
            // mengkosongkan data
            Serial.print("R1");
            Serial.print(simpan1);
        }
        if (indata[2]=='2')
        {
            simpan3=indata;
            indata="";
            // mengkosongkan data
            Serial.print("R2");
            Serial.print(simpan3);
        }
    }
    else if ((indata[0]=='$') && (indata[1]=='Q'))
    {
        // pemberian ID baru C
        indata[1]='C';
        if (indata[2]=='1')
        {
            simpan2=indata;
        }
    }
}

```

```

        indata="";
        Serial.print("R1_melalui_R2");
        Serial.print(simpan2);
    }
    if (indata[2]=='2')
    {
        simpan4=indata;
        indata="";
        Serial.print("R2_melalui_R1");
        Serial.print(simpan4);
    }
}

if(data_masuk=='\n')
{
    delay(100);
    baca();
    indata="";
    if(counph==5 && counshu==5)

    {
        // proses rata-rata data dari router 1
        suhu=suhu/counshu; ph=ph/counph;
        kirim="$C1_"+konvrsi(ph)+"_"+konvrsi(suhu)+"_"+tanggal+"_"+
        jam+"_%\n";
        counph=0;
        counshu=0;
        ph=0;
        suhu=0;
        Serial.print(kirim);
        cek=true;
    }
    else if(counph2==5 && counshu2==5)
    {
        // proses rata-rata data dari router 2
        suhu2=suhu2/counshu2; ph2=ph2/counph2;
        kirim2="$C2_"+konvrsi(ph2)+"_"+konvrsi(suhu2)+"_"+tanggal2+
        _"jam2+"_%\n";
        counph2=0;
        counshu2=0;
        ph2=0;
        suhu2=0;
        Serial.print(kirim2);
        cek2=true;
    }
    else if(counph3==5 && counshu3==5)
    {
        // proses rata-rata data dari router 1 yang melalui router 2
        suhu3=suhu3/counshu3; ph3=ph3/counph3;
        kirim3="$C1_"+konvrsi(ph3)+"_"+konvrsi(suhu3)+"_"+tanggal3+
        _"jam3+"_%\n";
        counph3=0;
        counshu3=0;
        ph3=0;
        suhu3=0;
        if(cek==false)Serial.println(kirim3);
        else cek=false;
    }
    else if(counph4==5 && counshu4==5)
}

```

```

{
// proses rata-rata data dari router 2 yang melalui router 1
suhu4=suhu4/counshu4; ph4=ph4/counph4;
kirim4="$C2_" +konvrsi(ph4)+"_"+konvrsi(suhu4)+"_"+tanggal4+
_+jam4+_%\n";
counph4=0;
counshu4=0;
ph4=0;
suhu4=0;
if(cek2==false)Serial.println(kirim4);
else cek2=false;
}
}

```

Dari skrip tersebut maka dapat dijelaskan seperti berikut:

1. Jika *node coordinator* telah menerima 5 data pH dan 5 data suhu dari *node router* 1 yang langsung ke *coordinator* maka selanjutnya *node coordinator* akan langsung merata-rata data pH dan data suhu, apabila hasil rata-rata dari data pH dan data suhu telah didapatkan maka selanjutnya *node coordinator* akan mengirimkan data tersebut ke *node end device*.
2. Jika *node coordinator* telah menerima 5 data pH dan 5 data suhu dari *node router* 2 yang langsung ke *coordinator* maka selanjutnya *node coordinator* akan langsung merata-rata data pH dan data suhu, apabila hasil rata-rata dari data pH dan data suhu telah didapatkan maka selanjutnya *node coordinator* akan mengirimkan data tersebut ke *node end device*.
3. Jika *node coordinator* telah menerima 5 data pH dan 5 data suhu dari *node router* 1 yang melalui *node router* 2 ke *coordinator* maka selanjutnya *node coordinator* akan langsung merata-rata data pH dan data suhu, apabila hasil rata-rata dari data pH dan data suhu telah didapatkan maka selanjutnya *node coordinator* akan mengirimkan data tersebut ke *node end device*, apabila *node coordinator* tidak menerima 5 data ph dan 5 data suhu dari *node router* 1 yang langsung ke *node coordinator*.

4. Jika *node coordinator* telah menerima 5 data pH dan 5 data suhu dari *node router 2* yang melalui *node router 1* ke *coordinator* maka selanjutnya *node coordinator* akan langsung merata-rata data pH dan data suhu, apabila hasil rata-rata dari data pH dan data suhu telah didapatkan maka selanjutnya *node coordinator* akan mengirimkan data tersebut ke *node end device*, apabila *node coordinator* tidak menerima 5 data ph dan 5 data suhu dari *node router 2* yang langsung ke *node coordinator*.

Gambar 4.9 menunjukkan hasil pengiriman dan penerimaan data antar *node router* yang ditampilkan di serial monitor pada setiap *node*.

```

$R1_7.904_31.500_15/6/2014_14:4:50_%
$Q2_7.377_31.250_15/6/2014_14:39:22_%
$R1_7.912_31.500_15/6/2014_14:5:13_%
$Q2_7.391_31.187_15/6/2014_14:39:44_%
$R1_7.921_31.500_15/6/2014_14:5:35_%
$Q2_7.399_31.187_15/6/2014_14:40:6_%
$R1_7.921_31.562_15/6/2014_14:5:57_%
$Q2_7.411_31.250_15/6/2014_14:40:29_%
$R1_7.923_31.500_15/6/2014_14:6:20_%
$Q2_7.425_31.187_15/6/2014_14:40:51_%
$R1_7.932_31.562_15/6/2014_14:6:42_%
$Q2_7.431_31.250_15/6/2014_14:41:14_%
$R1_7.938_31.562_15/6/2014_14:7:4_%
$Q2_7.439_31.250_15/6/2014_14:41:36_%
$R1_7.938_31.562_15/6/2014_14:7:27_%
$Q2_7.445_31.250_15/6/2014_14:41:58_%
$R1_7.938_31.562_15/6/2014_14:7:49_%
$Q2_7.454_31.250_15/6/2014_14:42:20_%
$R1_7.935_31.562_15/6/2014_14:8:11_%
$Q2_7.465_31.187_15/6/2014_14:42:43_%
$R1_7.935_31.500_15/6/2014_14:8:34_%
$Q2_7.473_31.250_15/6/2014_14:43:5_%
$R1_7.938_31.562_15/6/2014_14:8:56_%
$Q2_7.473_31.250_15/6/2014_14:43:27_%
$R1_7.941_31.500_15/6/2014_14:9:18_%
$Q2_7.482_31.250_15/6/2014_14:43:50_%
$R1_7.941_31.500_15/6/2014_14:9:41_%
$Q2_7.488_31.250_15/6/2014_14:44:12_%
$R1_7.938_31.500_15/6/2014_14:10:3_%
$Q1_7.904_31.500_15/6/2014_14:4:50_%
$R2_7.377_31.250_15/6/2014_14:39:22_%
$Q1_7.912_31.500_15/6/2014_14:5:13_%
$R2_7.391_31.187_15/6/2014_14:39:44_%
$Q1_7.921_31.500_15/6/2014_14:5:35_%
$R2_7.399_31.187_15/6/2014_14:40:6_%
$Q1_7.921_31.562_15/6/2014_14:5:57_%
$R2_7.411_31.250_15/6/2014_14:40:29_%
$Q1_7.923_31.500_15/6/2014_14:6:20_%
$R2_7.425_31.187_15/6/2014_14:40:51_%
$Q1_7.932_31.562_15/6/2014_14:6:42_%
$R2_7.431_31.250_15/6/2014_14:41:14_%
$Q1_7.938_31.562_15/6/2014_14:7:4_%
$R2_7.439_31.250_15/6/2014_14:41:36_%
$Q1_7.938_31.562_15/6/2014_14:7:27_%
$R2_7.445_31.250_15/6/2014_14:41:58_%
$Q1_7.938_31.562_15/6/2014_14:7:49_%
$R2_7.454_31.250_15/6/2014_14:42:20_%
$Q1_7.935_31.562_15/6/2014_14:8:11_%
$R2_7.465_31.187_15/6/2014_14:42:43_%
$Q1_7.935_31.500_15/6/2014_14:8:34_%
$R2_7.473_31.250_15/6/2014_14:43:5_%
$Q1_7.938_31.562_15/6/2014_14:8:56_%
$R2_7.473_31.250_15/6/2014_14:43:27_%
$Q1_7.941_31.500_15/6/2014_14:9:18_%
$R2_7.482_31.250_15/6/2014_14:43:50_%
$Q1_7.941_31.500_15/6/2014_14:9:41_%
$R2_7.488_31.250_15/6/2014_14:44:12_%
$Q1_7.938_31.500_15/6/2014_14:10:3_%

```

Gambar 4.9 Pengiriman dan Penerimaan Data antar *Node Router*

Berdasarkan Gambar 4.9 dapat dilihat bahwa data pH air, suhu air dan waktu yang berasal dari *node router 1* memiliki ID “R1” dan *node router 2*

memiliki ID “R2”. Kemudian data dengan ID “Q2” pada *node router* 1 merupakan data yang berasal dari *node router* 2 yang diterima oleh *node router* 1, perhatikan nilai pH air, suhu air dan waktu dengan ID “R2” pada *node router* 2 dan ID “Q2” pada *node router* 1 akan bernilai sama. Begitu juga data dengan ID “Q1” pada *node router* 2 merupakan data yang berasal dari *node router* 1 yang diterima oleh *node router* 2, perhatikan nilai pH air, suhu air dan waktu dengan ID “R1” pada *node router* 1 dan ID “Q1” pada *node router* 2 akan bernilai sama.

Data tersebut selanjutnya akan diteruskan atau dikirimkan ke *node coordinator*.

Untuk hasil pengiriman dan penerimaan data antara *node router* dan *node coordinator* serta hasil rata-rata data diperlihatkan pada Gambar 4.10

R1#C1_7.904_31.500_15/6/2014_14:4:50 \$	R2_melalui_R1#C2_7.425_31.187_15/6/2014_14:40:51 \$	R2_melalui_R1#C2_7.513_31.187_15/6/2014_14:46:26 \$
R1_melalui_R2#C1_7.904_31.500_15/6/2014_14:4:50 \$	R1#C1_7.932_31.562_15/6/2014_14:6:42 \$	R1#C1_7.949_31.500_15/6/2014_14:12:17 \$
R2#C2_7.377_31.250_15/6/2014_14:39:22 \$	R1_melalui_R2#C1_7.932_31.562_15/6/2014_14:6:42 \$	R1_melalui_R2#C1_7.949_31.500_15/6/2014_14:12:17 \$
R2_melalui_R1#C2_7.377_31.250_15/6/2014_14:39:22 \$	R2#C2_7.431_31.250_15/6/2014_14:41:14 \$	R2#C2_7.522_31.125_15/6/2014_14:46:46 \$
R1#C1_7.912_31.500_15/6/2014_14:5:13 \$	R2_melalui_R1#C2_7.431_31.250_15/6/2014_14:41:14 \$	R2_melalui_R1#C2_7.522_31.125_15/6/2014_14:46:46 \$
R1_melalui_R2#C1_7.912_31.500_15/6/2014_14:5:13 \$	R1#C1_7.930_31.562_15/6/2014_14:7:4 \$	R1#C1_7.952_31.500_15/6/2014_14:12:39 \$
R2#C2_7.391_31.187_15/6/2014_14:39:44 \$	R1_melalui_R2#C1_7.938_31.562_15/6/2014_14:7:4 \$	R1_melalui_R2#C1_7.952_31.500_15/6/2014_14:12:39 \$
R2_melalui_R1#C2_7.391_31.187_15/6/2014_14:39:44 \$	R2#C2_7.439_31.250_15/6/2014_14:41:36 \$	R2#C2_7.525_31.187_15/6/2014_14:47:11 \$
R1#C1_7.921_31.500_15/6/2014_14:5:35 \$	R2_melalui_R1#C2_7.439_31.250_15/6/2014_14:41:36 \$	R2_melalui_R1#C2_7.525_31.187_15/6/2014_14:47:11 \$
R1_melalui_R2#C1_7.921_31.500_15/6/2014_14:5:35 \$	R1#C1_7.930_31.562_15/6/2014_14:7:27 \$	R1#C1_7.985_31.437_15/6/2014_14:13:1 \$
R2#C2_7.399_31.187_15/6/2014_14:40:6 \$	R1_melalui_R2#C1_7.938_31.562_15/6/2014_14:7:27 \$	R1_melalui_R2#C1_7.955_31.437_15/6/2014_14:13:1 \$
R2_melalui_R1#C2_7.399_31.187_15/6/2014_14:40:6 \$	R2#C2_7.445_31.250_15/6/2014_14:41:58 \$	R2#C2_7.533_31.187_15/6/2014_14:47:33 \$
R1#C1_7.921_31.562_15/6/2014_14:5:57 \$	R2_melalui_R1#C2_7.445_31.250_15/6/2014_14:41:58 \$	R2_melalui_R1#C2_7.533_31.187_15/6/2014_14:47:33 \$
R1_melalui_R2#C1_7.921_31.562_15/6/2014_14:5:57 \$	R1#C1_7.930_31.562_15/6/2014_14:7:49 \$	R1#C1_7.952_31.437_15/6/2014_14:13:24 \$
R2#C2_7.411_31.250_15/6/2014_14:40:29 \$	R1_melalui_R2#C1_7.938_31.562_15/6/2014_14:7:49 \$	R1_melalui_R2#C1_7.952_31.437_15/6/2014_14:13:24 \$
R2_melalui_R1#C2_7.411_31.250_15/6/2014_14:40:29 \$	R2#C2_7.454_31.250_15/6/2014_14:42:20 \$	R2#C2_7.536_31.187_15/6/2014_14:47:55 \$
R1#C1_7.923_31.500_15/6/2014_14:6:20 \$	R2_melalui_R1#C2_7.454_31.250_15/6/2014_14:42:20 \$	R2_melalui_R1#C2_7.536_31.187_15/6/2014_14:47:55 \$
#C1_7.916_31.512_15/6/2014_14:6:20 \$	R1#C1_7.930_31.562_15/6/2014_14:8:11 \$	R1#C1_7.952_31.375_15/6/2014_14:13:46 \$
R1_melalui_R2#C1_7.923_31.500_15/6/2014_14:6:20 \$	#C1_7.936_31.562_15/6/2014_14:8:11 \$	#C1_7.951_31.449_15/6/2014_14:13:46 \$
R2#C2_7.425_31.187_15/6/2014_14:40:51 \$	R1_melalui_R2#C1_7.935_31.562_15/6/2014_14:8:11 \$	R1_melalui_R2#C1_7.952_31.375_15/6/2014_14:13:46 \$
#C2_7.400_31.212_15/6/2014_14:40:51 \$	R2#C2_7.465_31.187_15/6/2014_14:42:43 \$	R2#C2_7.539_31.125_15/6/2014_14:48:18 \$
		#C2_7.531_31.162_15/6/2014_14:48:18 \$

Gambar 4.10 Hasil Pengiriman/Penerimaan/Rata-rata Data pada *Node Coordinator*

Gambar 4.10 merupakan hasil pengiriman / penerimaan dan perbandingan data pada *node coordinator*. Pada *node coordinator* terdapat beberapa data yang diterima dan dikirimkan antara lain:

1. Data R1\$C1_7.904_31.500_15/6/2014_14:4:50_% merupakan data yang diterima berasal dari *node router* 1.
2. Data R1_melalui_R2\$C1_7.904_31.500_15/6/2014_14:4:50_% merupakan data yang diterima berasal dari *node router* 1 melalui *node router* 2.
3. Data R2\$C2_7.377_31.250_15/6/2014_14:39:22_% merupakan data yang diterima berasal dari *node router* 2.
4. Data R2_melalui_R1\$C2_7.377_31.250_15/6/2014_14:39:22_% merupakan data yang diterima berasal dari *node router* 2 melalui *node router* 1.
5. Data \$C1_7,916_31.512_15/6/2014_14:6:20_% dan \$C2_7,400_31.212_15/6/2014_14:40:51_% merupakan hasil rata-rata data yang dikirimkan ke *node router end device*.

Berdasarkan Gambar 4.9 dan Gambar 4.10 dapat diketahui apakah terjadi data *crash* atau tabrakan data pada saat pengiriman data antar *node router* 1, 2, dan *node coordinator* sehingga mempengaruhi penerimaan data pada *node* yang dituju. Hasil dalam pengujian protokol komunikasi ini tidak ada data yang *crash* atau bertabrakan karena skrip atau program disetiap *node* mengirimkan datanya selalu bergantian atau tidak secara bersamaan, sehingga tidak terjadi *crash* atau tabrakan saat pengiriman datanya.

4.5 Pengujian Jarak Kemampuan Pengiriman dan Penerimaan Xbee

Pada pengujian ini bertujuan untuk mengetahui kemampuan jangkauan atau jarak pengiriman dan penerimaan data pada WSN menggunakan Xbee S2.

4.5.1 Peralatan

Peralatan yang digunakan pada setiap *node* dalam pengujian ini adalah sebagai berikut:

1. Arduino uno
2. Xbee pro S2 dan Xbee S2 beserta *shield*
3. Sensor ultrasonik PING dan modul RTC pada *node router*
4. Kabel USB
5. *Node coordinator* terhubung pada komputer
6. Baterai 9v (pada *node router*)

4.5.2 Prosedur Pengujian

Langkah-langkah dalam pengujian ini sebagai berikut:

1. Hubungkan *node coordinator* pada komputer dengan kabel USB.
2. aktifkan komputer dan jalankan program arduino IDE.
3. *Upload* skrip atau program sebagai *coordinator* yang digunakan pada *node coordinator*.
4. Setelah *upload* program selesai, buka serial monitor.
5. *Upload* skrip atau program untuk *router 1* dan *router 2* seperti pada saat *upload* program *coordinator*.
6. Hubungkan *node router 1* dan *router 2* dengan catu daya atau baterai pada masing-masing *node*.
7. Ukur jarak antar Xbee, dan carilah jangkauan atau jarak maksimal pengiriman dan penerimaan data.

4.5.3 Hasil Pengujian

Dari langkah – langkah pengujian komunikasi data pada Xbee yang telah dilakukan didapatkan hasil jarak atau jangkauan pengiriman dan penerimaan data sebagai berikut:

A. Pengujian Selisih pH dan suhu

Tabel 4.4 Pengiriman data dari *Router 1* dan *Router 2* pada jarak 20 meter

No	Router 1		Router 2	
	pH	Suhu °C	pH	Suhu °C
1	7,58	30,12	3,97	30,04
2	7,57	30,10	3,97	30,10
3	7,57	29,98	3,98	30,01
4	7,56	30,01	3,89	30,10
5	7,58	29,99	3,89	29,99

Tabel 4.5 Pengiriman data dari *Router 1* dan *Router 2* pada jarak 40 meter

No	Router 1		Router 2	
	pH	Suhu °C	pH	Suhu °C
1	7,56	30,10	3,89	30,05
2	7,56	30,10	3,97	30,10
3	7,67	30,11	3,99	29,98
4	7,53	29,98	3,98	30,10
5	7,49	29,97	3,98	29,99

Tabel 4.6 Pengiriman data dari *Router 1* dan *Router 2* pada jarak 60 meter

No	Router 1		Router 2	
	pH	Suhu °C	pH	Suhu °C
1	7,51	30,10	3,89	30,10
2	7,60	30,12	3,98	30,09
3	7,46	29,97	3,98	30,11
4	7,53	29,99	3,87	30,06
5	7,57	30,09	3,89	29,98

Tabel 4.7 Pengiriman data dari *Router 1* dan *Router 2* pada jarak 80 meter

No	Router 1		Router 2	
	pH	Suhu °C	pH	Suhu °C
1	7,50	30,10	3,87	30,05
2	7,53	30,05	3,98	30,10
3	7,61	30,01	3,89	30,12

4	7,49	30,09	3,99	30,01
5	7,55	30,12	3,99	29,99

Tabel 4.8 Pengiriman data dari Router 1 dan Router 2 pada jarak 100 meter

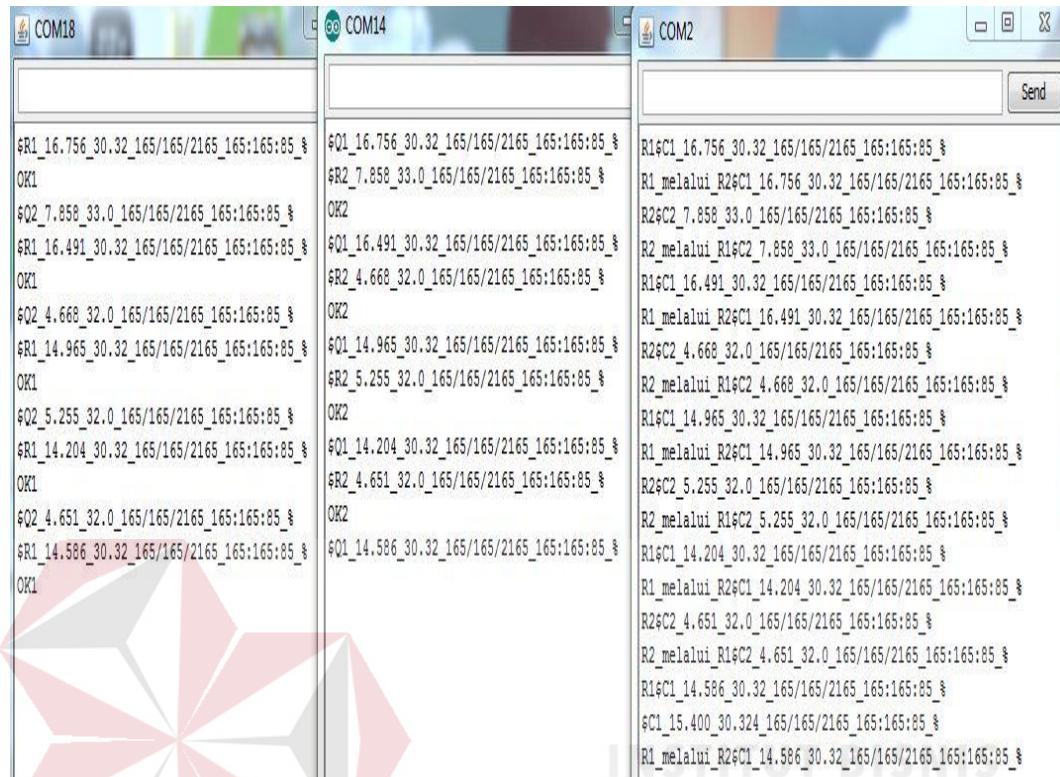
No	Router 1		Router 2	
	pH	Suhu °C	pH	Suhu °C
1	7,53	30,10	3,87	30,09
2	7,48	30,12	3,88	30,13
3	7,61	29,07	3,99	30,10
4	7,46	29,97	3,89	30,12
5	7,55	30,10	3,97	30,10

Tabel 4.9 Selisih pengiriman data antara jarak 20 meter dan 100 meter dengan rumus $|Data\ 100\ meter - Data\ 20\ meter|$

No	Router 1		Router 2	
	pH	Suhu °C	pH	Suhu °C
1	0,05	0,02	0,01	0,05
2	0,09	0,02	0,09	0,13
3	0,04	0,91	0,01	0,09
4	0,01	0,04	0	0,12
5	0,03	0,02	0,08	0,11
Rata-rata	0,044	0,202	0,038	0,1

Tabel 4.9 menunjukkan selisih pengiriman data antara jarak 20 meter dan 100 meter. Dari tabel 4.9 terdapat rata-rata selisih pengiriman data pH dari *router 1* sebesar 0,044 dan dari *router 2* sebesar 0,038. Rata-rata selisih pengiriman data suhu dari *router 1* sebesar 0,202 °C dan dari *router 2* sebesar 0,1 °C. Yang berarti seslisih Ph dan suhu untuk berbagai variasi tidak terlalu besar (kecil).

B. Pengujian Error



Gambar 4.11 Error Checking

Gambar 4.10 menunjukkan proses *error checking* yang ada pada *router 1*, *router 2*, *coordinator*. Setelah *router 1* mengirimkan data ke *coordinator* dan *coordinator* menerima data tersebut, selanjutnya *coordinator* akan mengirimkan data tersebut kembali ke *router 1*, apabila data yang dikirim oleh *coordinator* ke *router 1* sama dengan data yang dikirim dari *router 1* ke *coordinator*, maka *router 1* akan memberikan pesan “OK1. Demikian juga dengan *router 2*, bedanya *router 2* akan memberikan pesan “OK2”.

4.5.4 Analisis Hasil Pengujian

1. Pada Pengujian Jarak Kemampuan Pengiriman dan Penerimaan Xbee dapat diambil kesimpulan bahwa selisih pH dan suhu untuk berbagai variasi jarak tidak terlalu besar (kecil).

2. Pada Pengujian *Error* data yang dikirimkan baik itu dari *router* 1 maupun dari *router* 2 dapat diterima dengan baik, ini ditunjukan dengan data yang dikirim dari *coordinator* ke *router* sama dengan data yang dikirim dari *router* ke *coordinator*.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Sistem WSN dalam pengukuran pH air dan suhu air pada prototipe ini menggunakan 4 *node* yaitu *node router 1*, *router 2*, *coordinator* dan *router end device*. *Node router 1* dan *2* terdiri dari modul mikrokontroler arduino uno sebagai otak keseluruhan sistem untuk menentukan pembacaan pH air oleh sensor pH meter kit, suhu air menggunakan DS18B20 dan pembacaan waktu oleh modul RTC. Data pH air, suhu air dan waktu dikirimkan ke *node* lainnya secara nirkabel menggunakan modul Xbee series 2 beserta *shield*. Sedangkan *node coordinator* dan *router end device* terdiri dari modul mikrokontroler arduino uno dan modul Xbee series 2 beserta *shield*.
2. Sistem melakukan pengukuran pH air, suhu air dan waktu secara real time melalui komunikasi nirkabel dan didapatkan hasil pengujian seperti:
 - a) Rata - rata persentase kesalahan pengukuran pH air menggunakan sensor pH meter kit dengan alat ukur sebenarnya berkisar antara 0,37% sampai dengan 1,86% .
 - b) Rata - rata persentase kesalahan pengukuran suhu air menggunakan sensor DS18B20 dengan alat ukur sebenarnya berkisar antara 0,608% sampai dengan 0,73%.
3. Berdasarkan pengujian terhadap protokol komunikasi WSN, pengiriman data dari setiap *node* ke *node* lainnya yang dituju benar dan diterima oleh sisi penerima.

4. Terdapat selisih 4,4% dari data pengiriman pH antara jarak 20 meter dan 100 meter pada *router 1* serta terdapat selisih 3,8% dari data pengiriman pH antara jarak 20 meter dan 100 meter pada *router 2* dan terdapat selisih 20,2% dari data pengiriman suhu antara jarak 20 meter dan 100 meter pada *router 1* serta terdapat selisih 10% dari data pengiriman suhu antara jarak 20 meter dan 100 meter pada *router 2*
5. Topologi yang digunakan pada Tugas Akhir ini menggunakan topologi *MESH NETWORK*, karena pengiriman data dapat dilakukan melalui *router 1* ke *coordinator*, *router 1* ke *router 2*, *router 2* ke *coordinator* dan *router 2* ke *router 1*.

5.2 Saran

Agar diperoleh hasil yang lebih baik dan sebagai pengembangan pada penelitian berikutnya sebaiknya mempertimbangkan langkah ini:

1. Apabila ingin lebih mendapatkan hasil yang lebih akurat, bisa ditambahkan dengan parameter lain, seperti parameter kekeruhan, warna, dan lain sebagainya.
2. Perancangan ini merupakan sebuah prototype, sehingga diharapkan dapat diimplementasikan pada kondisi yang real yaitu diterapkan langsung pada sungai yang sesungguhnya.

DAFTAR PUSTAKA

- Blum, J. 2010. *Exploring Arduino*. Sebastopol : Wiley.
- DfRobot.2013. Real Time Clock. (Diakses 10 November 2013,
http://www.dfrobot.com/wiki/index.php?title=Real_Time_Clock_Module_%28DS1307%29_%28SKU:DFR0151%29)
- DfRobot.2013. Waterproof DS18B20 Digital Temperature Sensor. (Diakses 10 November 2013, http://dfrobot.com/wiki/index.php?title=Waterproof_DS18B20_Digital_Temperature_Sensor_%28SKU:DFR0198%29)
- Dinas Kesehatan. 1990. PERATURAN MENTERI KESEHATAN Nomor : 416/MEN.KES/PER/IX/1990 *Tentang Syarat-syarat Dan Pengawasan Kualitas Air*.
- Faludi, R. 2010. *Building wireless sensor networking*. Sebastopol: O'Reilly.
- Huang Y.K., Pang A.C., Hsiu P.C., Zhuang W., and Liu P., Distributed Throughput Optimization for ZigBee Cluster-Tree Networks, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, 523-520, 2012.
- Kadir Abdul. 2013. *Panduan Praktis Mempelajari Aplikasi Mikrokontroler Dan Pemrogramannya Menggunakan Arduino*. Jakarta : Penerbit Andi Publisher.
- Mahida,U.N. 1984. *Pencemaran Air dan Pemanfaatan Limbah industri*. Cetakan Pertama. Jakarta : Penerbit CV. Rajawali.
- Melgar, E. R., & Diez, C. C. 2012. *Arduino and Kinect Projects: Design, Build, Blow Their Minds*. Apress.
- Wardhana WA. 1995. *Dampak Pencemaran Lingkungan*. Yogyakarta : Penerbit Andi Offset Jogjakarta.
- Warlina L. 2004. *Pencemaran air : sumber, dampak, dan penanggulangannya*. (Makalah pribadi).Bogor : Sekolah Pasca Sarjana Institut Pertanian Bogor.