

**SISTEM *MONITORING* TINGGI PERMUKAAN AIR PADA
PENAMPUNG MENGGUNAKAN KOMUNIKASI XBEE
BERBASIS MIKROKONTROLER**

LAPORAN AKHIR



oleh

**Renaldi Habib Kholik
E32140964**

**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2017**

**SISTEM *MONITORING* TINGGI PERMUKAAN AIR PADA
PENAMPUNG MENGGUNAKAN KOMUNIKASI XBEE
BERBASIS MIKROKONTROLER**

LAPORAN AKHIR



Sebagai salah satu syarat untuk memperoleh gelar Ahli Madya (A.Md.) Di
Program Studi Teknik Komputer
Jurusan Teknologi Informasi

oleh

**Renaldi Habib Kholik
E32140964**

**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2017**

**KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI JEMBER**

**SISTEM MONITORING TINGGI PERMUKAAN AIR PADA
PENAMPUNG MENGGUNAKAN KOMUKASI XBEE
BERBASIS MIKROKONTROLER**

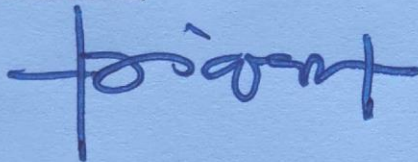
Telah diuji pada tanggal 06 September 2017
Telah dinyatakan Memenuhi Syarat

Tim Penguji:
Ketua,



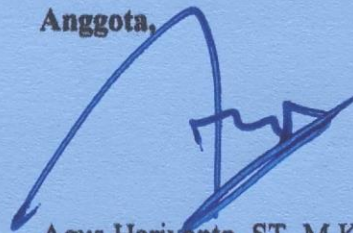
Surateno, S.Kom. M.Kom
NIP. 19790703 200312 1 001

Sekretaris,



Moch. Munih Dian W. S.Kom. MT
NIP. 19700831 199803 1 001

Anggota,

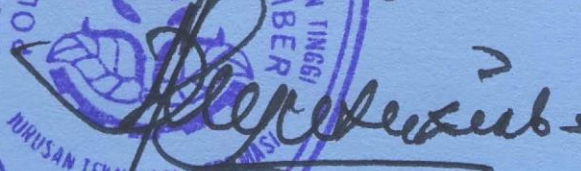


Agus Hariyanto, ST. M.Kom
NIP. 19780817 200312 1 005



Mengesahkan:

Ketua Jurusan Teknologi Informasi



Wahyu Kurnia Dewanto, S.Kom. MT
NIP. 19710408 200112 1 003

**KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI JEMBER**

**PROTOTYPE PENGENDALI PINTU GERBANG DENGAN PERINTAH
SUARA MENGGUNAKAN BLUETOOTH BERBASIS
MIKROKONTROLER**

Oleh :

**Renaldi Habib Kholik
NIM. E32140964**

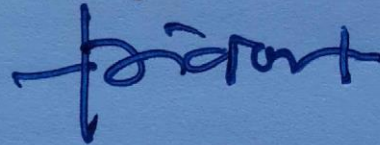
Diuji pada tanggal: 06 September 2017

Pembimbing I,



Surateno, S.Kom, M.Kom
NIP. 19790703 200312 1 001

Pembimbing II,



Moch. Munih Dian W. S.Kom, MT
NIP. 19700831 199803 1 001



Mengesahkan,
Ketua Jurusan Teknologi Informasi

Wahyu Kurnia Dewanto, S.Kom, MT
NIP. 19710408 200112 1 003

HALAMAN PERSEMBAHAN

Dengan mengucapkan syukur Alhamdulillah, karya sederhana ini dipersembahkan untuk orang-orang terkasih :

1. Kedua orang tua dan keluarga yang tidak pernah berhenti memberi dukungan dan doa yang tidak ada satupun kata di dunia ini yang pantas untuk menggambarkan perjuangan, pengorbanan dan didikan yang telah mereka berikan.
2. Bapak Surateno, S.Kom, M.Kom, Bapak Moch. Munih Dian W, S.Kom, MT serta seluruh Dosen dan Staf Jurusan Teknologi Informasi POLITEKNIK NEGERI JEMBER yang telah memberikan ilmu, bimbingan, semangat dan motivasi selama penyelesaian tugas akhir ini. .
3. Teman-teman seangkatan TKK 2014. Ahmad Munawir, Adi Surahman, Febri Bafufi B, Danial Rifqi T, Miftahol Arifin, Jefri Oktanuriawan, Firman Arief A, Rendi S, Hilman Fathoni serta teman-teman lainnya yang telah mendukung dan membantu selama pengerjaan tugas akhir ini.
4. Terima kasih untuk almamater tercinta Politeknik Negeri Jember.

MOTTO

“Hidup ini hanyalah untuk perjalanan pulang”

-Renaldi Habib Kholik-

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Renaldi Habib Kholik

NIM : E32140964

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam Laporan Akhir saya yang berjudul “Sistem Monitoring Tinggi Permukaan Air Pada Penampung Menggunakan Komunikasi XBee Berbasis Mikrokontroler” merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam daftar pustaka dibagian akhir Laporan Akhir.

Jember, 06 September 2017

Renaldi Habib Kholik
NIM. E32140964

PRAKATA

Puji syukur Alhamdulillah penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat serta hidayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir yang merupakan tahapan proses untuk memperoleh gelar Ahli Madya Teknologi Informasi di Politeknik Negeri Jember. Keberhasilan penulis dalam menyelesaikan Tugas Akhir ini tidak lepas dari bantuan orang-orang yang dengan sepenuh hati memberikan doa, bimbingan dan dukungan. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Direktur Politeknik Negeri Jember.
2. Ketua Jurusan Teknologi Informasi.
3. Ketua Program Studi Teknik Komputer.
4. Surateno, S.Kom, M,Kom selaku Pembimbing 1 Tugas Akhir.
5. Moch. Munih Dian W, S.Kom, MT selaku Pembimbing 2 Tugas Akhir.
6. Semua pihak yang telah membantu dalam penyusunan Tugas Akhir.

Penulis menyadari bahwa dalam menyusun laporan ini terdapat kekurangan dan keterbatasan, oleh karena itu kritik dan saran yang bersifat membangun demi kesempurnaan dimasa yang akan datang sangat diharapkan. Akhir kata penulis berharap semoga laporan Tugas Akhir ini dapat bermanfaat bagi seluruh pembaca.

Jember, 06 September 2017

Renaldi Habib Kholik
NIM. E32140964

RINGKASAN

Sistem Monitoring Tinggi Permukaan Air Pada Penampung Menggunakan Komunikasi Data XBee Berbasis Mikrokontroler, Renaldi Habib Kholik, NIM E32140964, Tahun 2017, 57 hlm., Teknologi Informasi, Politeknik Negeri Jember, Surateno, S.Kom, M.Kom (Pembimbing I), Moch. Munih Dian W, S.Kom, MT (Pembimbing II).

Penggunaan sistem *monitoring* sebagai media pemantau kondisi dari suatu objek dari jarak jauh semakin populer saat ini. *Monitoring* jarak jauh sudah banyak dibutuhkan dalam menunjang kegiatan masyarakat. Contoh dari kasus ini ialah *monitoring* tinggi permukaan air pada penampung yang letaknya sulit dijangkau oleh *operator*, sehingga keadaan ini akan menyebabkan terhambatnya proses *monitoring* yang seharusnya digunakan secara *real time*. Melihat keadaan tersebut diperlukan suatu sistem *montoring* yang efisien baik dalam instalasi, pengaplikasian maupun pengoperasian.

Dalam laporan akhir ini dibuat suatu sistem *monitoring* tinggi permukaan air pada penampung dengan transmisi data XBee. Dengan komponen utama berupa Mikrokontroler Arduino Uno sebagai pusat pengolah data dan sensor ultrasonik HC-SRF04 sebagai sensor jarak untuk membaca ketinggian permukaan air, lalu hasil dari pembacaan tersebut akan ditransmisikan secara nirkabel menggunakan Xbee ke komputer untuk kemudian ditampilkan tingkat ketinggian permukaan air pada penampung dengan tampilan grafis menggunakan aplikasi visual basic.

DAFTAR ISI

	Halaman
HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO.....	vi
SURAT PERNYATAAN	vii
PRAKATA	viii
RINGKASAN	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	viii
DAFTAR LAMPIRAN.....	ix
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat	3
BAB 2. TINJAUAN PUSTAKA.....	4
2.1 Pengertian <i>Monitoring</i>	4
2.2 Pengertian Air.....	4
2.3 Pengertian Penampung.....	4
2.4 XBee	5
2.5 Mikrokontroler	6
2.5.1 Mikrokontroler Arduino.....	8
2.5.2 Arduino Uno.....	8
2.6 Karya Tulis yang Mendahului (<i>State of The Art</i>).....	11
BAB 3. METODE KEGIATAN	13
3.1 Tempat dan Waktu Pelaksanaan.....	13
3.2 Alat dan Bahan.....	13
3.2.1 Alat.....	13
3.2.2 Bahan	14

3.3 Metode Kegiatan	15
3.3.1 Perancangan	15
3.3.2 Pembuatan	16
3.3.3 Pengujian	16
BAB 4. PEMBAHASAN	17
4.1 Perancangan Alat	17
4.1.1 Perancangan Keseluruhan Sistem	17
4.1.2 Perancangan <i>Hardware</i>	18
4.1.3 Perancangan <i>Software</i>	19
4.2 Pembuatan Alat	24
4.2.1 Pembuatan <i>Hardware</i>	24
4.2.2 Konfigutasi XBee	27
4.2.3 Pembuatan <i>Software</i>	30
4.3 Pengujian Alat	34
4.3.1 Uji Coba Koneksi XBee	34
4.3.2 Uji Coba Sensor Ultrasonik	35
BAB 5. KESIMPULAN DAN SARAN	37
5.1 Kesimpulan	37
5.2 Saran	37
DAFTAR PUSTAKA	38
LAMPIRAN	40

DAFTAR TABEL

	Halaman
2.1 Karya Tulis yang Mendahului	11
3.1 Daftar Alat Perangkat Keras yang Digunakan	13
3.2 Daftar Alat Perangkat Lunak yang Digunakan.....	14
3.3 Daftar Bahan yang dibutuhkan	14
4.3 Pengujian Jarak Koneksi XBee	34
4.4 Pengujian Sensor Ultrasonik dengan Kondisi Permukaan Air Tenang	35
4.5 Pengujian Sensor Ultrasonik dengan Kondisi Permukaan Air Tidak Tenang	36

DAFTAR GAMBAR

	Halaman
2.1 Mikrokontroler	7
2.2 Arduino Uno	9
3.1 Diagram Metode Kegiatan	15
4.1 Perancangan Sistem Kerja Alat	17
4.2 Perancangan <i>Hardware</i>	18
4.3 Perancangan <i>Flowchart</i> Program Arduino A	19
4.4 Perancangan <i>Flowchart</i> Program Arduino B	21
4.5 Perancangan <i>Flowchart</i> Aplikasi Visual Basic	23
4.6 Rangkaian Komponen Arduino B	24
4.7 Rangkaian Komponen Arduino A	25
4.8 Pemilihan Serial Port XBee	27
4.9 Konfigurasi XBee <i>Coordinator</i>	28
4.10 Konfigurasi XBee <i>Router</i>	29
4.11 <i>Testing</i> Koneksi XBee	29
4.12 Penulisan Program pada Arduino IDE	30
4.13 Pembuatan Objek Aplikasi Visual Basic	31
4.14 Penulisan <i>Coding</i> Aplikasi Visual Basic	31
4.15 Tampilan Aplikasi	32

DAFTAR LAMPIRAN

	Halaman
<i>A Source Code</i> Program Arduino A	40
<i>B Source Code</i> Program Arduino B.....	41
<i>C Source Code</i> Program Aplikasi Visual Basic.	47
D Lembar Revisi.....	54
E Biodata Penulis.	57

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Pada era globalisasi saat ini, kebutuhan akan keefektifitas dan efisiensi sangat diutamakan dalam berbagai bidang baik pada bidang industri, perusahaan, maupun rumah tangga. Hal tersebut telah menjadi pendorong dalam bidang teknologi untuk menciptakan suatu alat yang lebih efektif dan efisien agar dapat memberi kemudahan pada masyarakat dalam melakukan aktivitas sehari-hari.

Penggunaan sistem *monitoring* sebagai media pemantau kondisi dari suatu objek dari jarak jauh semakin populer saat ini. *Monitoring* jarak jauh sudah banyak dibutuhkan dalam menunjang kegiatan masyarakat. Contoh dari kasus ini ialah *monitoring* tinggi permukaan air pada penampung yang letaknya sulit dijangkau oleh *operator*, sehingga keadaan ini akan menyebabkan terhambatnya proses *monitoring* yang seharusnya digunakan secara *real time*. Melihat keadaan tersebut diperlukan suatu sistem *monitoring* yang efisien baik dalam instalasi, pengaplikasian maupun pengoperasian.

Penggunaan suatu sistem *monitoring* nirkabel dapat membuat proses pertukaran informasi dan komunikasi menjadi lebih cepat dan mudah. Namun kebanyakan dari sistem *monitoring* nirkabel yang telah dibuat selama ini masih menggunakan media *bluetooth* dan *wifi* sebagai media transmisi data yang notabene memiliki koneksi yang kurang efisien jika terhalang oleh suatu objek seperti tembok. Ataupun penggunaan jaringan internet sebagai media transmisi data, yang kita ketahui membutuhkan biaya yang cukup besar untuk mengaplikasikannya.

Maka pada laporan akhir ini dibuat suatu sistem *monitoring* tinggi permukaan air pada penampung dengan transmisi data XBee yang memiliki koneksi yang lebih efisien dan ekonomis. Berdasarkan *datasheet* XBee mampu terhubung hingga jarak 300 meter pada posisi *indoor* dan mampu terhubung hingga jarak 1,5 km pada posisi *outdoor*. Dengan komponen utama berupa sensor ultrasonik HC-SRF04 sebagai sensor jarak untuk membaca ketinggian permukaan air, lalu hasil dari pembacaan tersebut akan ditransmisikan secara nirkabel ke komputer untuk kemudian ditampilkan tingkat ketinggian permukaan air pada penampung dengan tampilan grafis menggunakan aplikasi visual basic. Dan kemudian pengguna dapat mengontrol tinggi permukaan air dengan menghidupkan pompa air. Sedangkan sebagai pusat pengolahan data, digunakan mikrokontroler Arduino Uno.

1.2 Rumusan Masalah

Pokok pembahasan dalam tugas akhir ini adalah :

1. Bagaimana membuat suatu sistem yang dapat menghitung tinggi permukaan air secara otomatis pada penampung air.
2. Bagaimana membuat suatu sistem *monitoring* jarak jauh secara nirkabel.
3. Bagaimana merancangan dan membuat program pada *Arduino UNO* untuk sistem *monitoring* nirkabel.

1.3 Batasan Masalah

Agar penulisan lebih terarah, dan tidak menyebar keluar dari topik masalah, maka pembahasan penulisan ini dibatasi ruang lingkup pembahasan sebagai berikut :

1. Pembacaan tinggi permukaan air hanya menggunakan 1 sensor ultrasonik.
2. Pembacaan tinggi permukaan air dengan sensor ultrasonik membutuhkan permukaan air yang tenang.
3. Penggunaan sensor ultrasonik untuk membaca tinggi permukaan air tidak bisa diaplikasikan pada permukaan air yang bergelombang.
4. Komunikasi *wireless* XBee S2B hanya dapat terhubung ke XBee dengan seri yang sama yaitu XBee S2B.
5. Aplikasi sistem *monitoring* hanya bisa berjalan di OS *Windows*.

1.4 Tujuan

Tujuan yang dicapai dalam laporan akhir ini adalah :

1. Membuat suatu alat ukur tinggi permukaan air pada bak penampung air.
2. Membuat suatu sistem *monitoring* yang lebih efektif dan efisien menggunakan komunikasi nirkabel.
3. Membuat suatu sistem *monitoring* dengan tampilan grafis menggunakan aplikasi visual basic.

1.5 Manfaat

Adapun manfaat dalam penulisan tugas akhir ini adalah:

1. Menghemat waktu manusia dalam melakukan kegiatan monitoring air pada penampung.
2. Dapat mengamalkan ilmu yang di dapat dari Kampus Politeknik Negeri Jember.
3. Sebagai syarat menyelesaikan studi di Kampus Politeknik Negeri Jember.

BAB 2. TINJAUAN PUSTAKA

2.1 Pengertian Monitoring

Monitoring adalah suatu proses pengumpulan dan menganalisis informasi dari penerapan suatu program termasuk mengecek secara reguler untuk melihat apakah kegiatan/program itu berjalan sesuai rencana sehingga masalah yang ditemui dapat diatasi.

Umumnya, *monitoring* digunakan dalam checking antara kinerja dan target yang telah ditentukan. *Monitoring* ditinjau dari hubungan terhadap manajemen kinerja adalah proses terintegrasi untuk memastikan bahwa proses berjalan sesuai rencana. *Monitoring* dapat memberikan informasi keberlangsungan proses untuk menetapkan langkah menuju ke arah perbaikan yang berkesinambungan. Pada pelaksanaannya, monitoring dilakukan ketika suatu proses sedang berlangsung. (Putro, 2014).

2.2 Pengertian Air

Air adalah suatu zat cair yang tidak mempunyai rasa, bau dan warna dan terdiri dari hidrogen dan oksigen dengan rumus kimia H_2O . Karena air mempunyai sifat yang hampir bisa digunakan untuk apa saja, maka air merupakan zat yang paling penting bagi semua bentuk kehidupan, sampai saat ini selain matahari yang merupakan sumber energi. Air dapat berupa air tawar dan air asin (air laut) yang merupakan bagian terbesar di bumi ini. Di dalam lingkungan alam proses, perubahan wujud, gerakan aliran air dan jenis air mengikuti suatu siklus keseimbangan dan dikenal dengan istilah siklus hidrologi (Yulianti, 2015).

2.3 Pengertian Penampung

Penampung berfungsi sebagai penyimpan air, baik dari hasil olahan maupun langsung dari sumber mata air. Selain itu, bak penampung berfungsi untuk mengatasi masalah naik turunnya kebutuhan air dan merupakan bagian dari pengelolaan distribusi air di masyarakat.

Bak penampung juga dapat digunakan untuk pengambilan air langsung seperti hidran umum. Bak penampung juga dapat memperbaiki mutu air melalui proses pengendapan. Bangunan bak penampung dapat berupa beton cor, pasangan bata, atau bak plastik (fiber) yang disesuaikan dengan kebutuhan dan kondisi. (Susianah, 2011).

2.4 XBee

Menerapkan komunikasi nirkabel (wireless communication) dalam aplikasi mikroelektronika sudah merupakan alternatif yang sering digunakan oleh pengembang saat komunikasi dengan kabel menemukan kendala. Media yang biasa digunakan antara lain wifi, bluetooth, inframerah dan yang paling mutakhir adalah XBee.

XBee atau Zigbee merupakan paduan dari kata zig yang berarti gerakan zig-zag, dan bee yang artinya lebah. Zigbee memiliki sifat komunikasi yang mirip dengan sifat komunikasi diantara para lebah yang melakukan gerakan – gerakan tak menentu dalam menyampaikan informasi adanya madu dari satu lebah ke lebah yang lain.

ZigBee adalah standar dari IEEE 802.15.4 untuk komunikasi data pada alat konsumen pribadi maupun untuk skala bisnis. ZigBee didesain dengan konsumsi daya yang rendah dan bekerja untuk jaringan personal tingkat rendah. Perangkat ZigBee biasa digunakan untuk mengendalikan sebuah alat lain maupun sebagai sebuah sensor yang wireless. ZigBee memiliki fitur dimana mampu mengatur jaringan sendiri, maupun mengatur pertukaran data pada jaringan[1]. Kelebihan dari ZigBee lainnya adalah membutuhkan daya rendah, sehingga bisa digunakan sebagai alat pengatur secara wireless yang penginstalan hanya perlu dilakukan sekali, karena hanya dengan satu baterai dapat membuat ZigBee bertahan hingga setahun.

Teknologi tanpa kabel atau wireless telah mengalami berkembang yang pesat dan penggunaan teknologi ini sendiri tidak lagi asing bagi masyarakat. Teknologi wireless yang banyak digunakan oleh masyarakat seperti bluetooth, mapun wifi, karena kedua perangkat ini sudah banyak diaplikasikan pada smartphone, laptop, dan beberapa gadget lainnya. Namun ZigBee sendiri bukan lah sebuah komunikasi yang digunakan untuk pengiriman data yang besar atau transfer rate yang tinggi. Bluetooth dan wifi merupakan sebuah standar yang bekerja untuk transfer rate dari tingkatan sedang hingga tinggi, sehingga cocok digunakan untuk pengiriman data yang besar. Sedangkan untuk sebuah device transfer rate rendah dapat kita gunakan standar ZigBee.

ZigBee adalah spesifikasi untuk protokol komunikasi tingkat tinggi yang mengacu pada standart IEEE 802.15.4 yang berhubungan dengan wireless personel area networks (WPANs). Teknologi dari ZigBee sendiri dimaksudkan untuk penggunaan pengiriman data secara wireless yang membutuhkan transmisi data rendah dan juga konsumsi daya rendah, dan juga tidak lebih mahal dibandingkan dengan WPANs lain seperti Bluetooth. Standar ZigBee sendiri lebih banyak diaplikasikan kepada system tertanam (embedded application) seperti pengendalian industri atau pengendali alat lain secara wireless, data

logging, dan juga sensor wireless dan lain-lain.

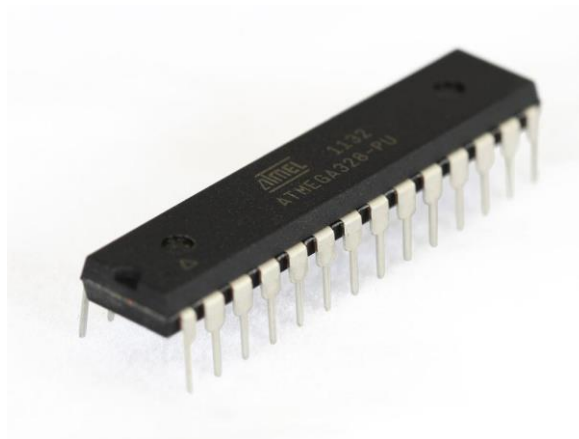
ZigBee memiliki transfer rate sekitar 250Kbps, yang lebih rendah dibandingkan dengan WPANs lain seperti bluetooth yang mempunyai transfer rate dengan 1Mbps. Sedangkan jarak atau range kerja dari ZigBee sendiri sekitar 76m, yang dimana jaraknya lebih jauh dibandingkan dengan Bluetooth. Dengan konsumsi daya yang rendah, maka sebuah alat yang menggunakan standar ZigBee dapat menggunakan sebuah baterai yang dapat membuat alat tersebut bertahan selama setengah sampai satu tahun. Prediksinya, bahwa semua smart home akan memiliki setidaknya 60 buah ZigBee dimana tiap ZigBee tersebut akan dapat saling berkomunikasi dan melakukan pekerjaan mereka secara bebas (Alif, 2017).

2.5 Mikrokontroler

Mikrokontroler adalah sebuah chip terintegrasi yang biasanya menjadi bagian dari sebuah embedded system (sistem yang didesain untuk melakukan satu atau lebih fungsi khusus yang real time). Mikrokontroler terdiri dari CPU, Memory, I/O port dan timer seperti sebuah komputer standar, tetapi karena didesain hanya untuk menjalankan satu fungsi yang spesifik dalam mengatur sebuah sistem, mikrokontroler ini bentuknya sangat kecil dan sederhana dan mencakup semua fungsi yang diperlukan pada sebuah chip tunggal.

Mikrokontroler berbeda dengan mikroprosesor, yang merupakan sebuah chip untuk tujuan umum yang digunakan untuk membuat sebuah komputer multi fungsi atau perangkat yang membutuhkan beberapa chip untuk menangani berbagai tugas. Mikrokontroler dimaksudkan untuk menjadi mandiri dan independen, dan berfungsi sebagai komputer khusus yang kecil.

Keuntungan besar dari mikrokontroler dibandingkan dengan menggunakan mikroprosesor yang lebih besar, adalah bahwa jumlah komponen dan biaya desain dari item yang dikendalikan dapat ditekan seminimum mungkin. Mikrokontroler biasanya didesain menggunakan teknologi CMOS (Complementary Metal Oxide Semiconductor), sebuah teknologi pembuatan chip terintegrasi (IC) efisien yang menggunakan daya kecil dan lebih kebal terhadap lonjakan listrik dibandingkan teknik yang lain. Gambar 2.1 merupakan salah satu contoh mikrokontroler.



Gambar 2.1. Mikrokontroler

Mikrokontroler (pengendali mikro) adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah chip. Mikrokontroler berbeda dengan mikroprosesor serba guna yang digunakan dalam sebuah PC, karena sebuah mikrokontroler umumnya telah berisi komponen sistem minimal mikroprosesor, yaitu memori, register dan antarmuka I/O yang terintegrasi bersama mikroprosesornya sendiri dalam sebuah chip.

Ditinjau dari segi arsitekturnya, mikroprosesor hanya merupakan single chip CPU, sedangkan mikrokontroler dalam IC-nya selain CPU juga terdapat device lain yang memungkinkan mikrokontroler berfungsi sebagai suatu single chip computer. Dalam sebuah IC mikrokontroler telah terdapat ROM, RAM, EPROM, serial interface dan paralel interface, timer, interrupt controller, converter analog ke digital, dan lainnya (tergantung feature yang melengkapi mikrokontroler tersebut). Sedangkan dari segi aplikasinya, mikroprosesor hanya berfungsi sebagai Central Processing Unit yang menjadi otak komputer, sedangkan mikrokontroler, dalam bentuknya yang mungil, pada umumnya ditujukan untuk melakukan tugas– tugas yang berorientasi kontrol pada rangkaian yang membutuhkan jumlah komponen minimum dan biaya rendah (Santoso, 2015).

2.5.1 Mikrokontroler Arduino

Arduino merupakan sebuah *platform* elektronik yang *open source*, berbasis pada *software* dan *hardware* yang fleksibel dan mudah digunakan, yang ditujukan untuk seniman, desainer dan setiap orang yang tertarik dalam membuat sebuah objek atau lingkungan yang interaktif.

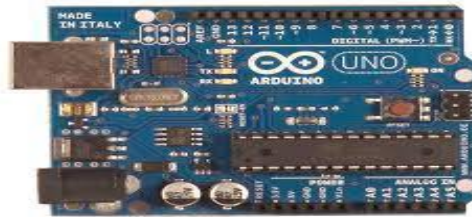
Arduino sebagai sebuah *platform* komputasi fisik (*Physical Computing*) yang *open source* pada *board input output* sederhana, yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespon situasi dan kondisi (Santoso, 2015).

2.5.2 Arduino Uno

Arduino UNO adalah sebuah board mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital input/output (6 di antaranya dapat digunakan sebagai output PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP header, dan sebuah tombol reset. Arduino UNO memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah computer dengan sebuah kabel USB atau mensuplainya dengan sebuah adaptor AC ke DC atau menggunakan baterai untuk memulainya.

Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-Serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai konverter USB-to-Serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial.

Nama “Uno” berarti satu dalam bahasa Italia, untuk menandai peluncuran Arduino 1.0 Uno dan versi 1.0 akan menjadi versi referensi dan Arduino. Uno adalah yang terbaru dalam serangkaian board USB Arduino, dan sebagai model referensi untuk platform Arduino. Gambar 2.2 Merupakan tampilan fisik dari Arduino Uno.



Gambar 2.2. Arduino Uno

a. Spesifikasi Arduino Uno

1. Microcontroller Atmega328
2. Operasi tegangan 5Volt
3. Input Tegangan (disarankan) 7-12V
4. Input Tegangan(batas) 6-20V
5. Pin I/O digital : 14 pin (dimana 6 bisa untuk PWM)
6. Pin Analog : 6 pin
7. Arus 50 mA
8. Flash Memory 32 Kb (Atmega328) yang 0,5 Kb digunakan oleh bootloader
9. SRAM 2 KB (Atmega328)
10. Clock Speed 16 MHz

b. Input Ouput

Masing-masing dari 14 pin UNO dapat digunakan sebagai input atau output, menggunakan perintah fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()` yang menggunakan tegangan operasi 5 volt. Tiap pin dapat menerima arus maksimal hingga 40mA dan resistor internal pull-up antara 20-50 kohm, beberapa pin memiliki fungsi kekhususan antara lain:

1. Serial: 0 (RX) dan 1 (TX). Sebagai penerima (RX) dan pemancar (TX) TTL serial data. Pin ini terkoneksi untuk pin korespondensi chip ATmega8U2 USB-toTTL Serial.
2. External Interrupts: 2 dan 3. Pin ini berfungsi sebagai konfigurasi trigger saat interupsi value low, naik, dan tepi, atau nilai value yang berubah-ubah
3. PWM: 3, 5, 6, 9, 10, dan 11. Melayani output 8-bit PWM dengan fungsi `analogWrite()`

4. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin yang support komunikasi SPI menggunakan SPI library
5. LED: 13. Terdapat LED indikator bawaan (built-in) dihubungkan ke digital pin 13, ketika nilai value HIGH led akan ON, saat value LOW led akan OFF
Uno memiliki 6 analog input tertulis di label A0 hingga A5, masing-masingnya memberikan 10 bit resolusi (1024). Secara asal input analog tersebut terukur dari 0 (ground) sampai 5 volt, itupun memungkinkan perubahan teratas dari jarak yang digunakan oleh pin AREF dengan fungsi analog Reference().

Sebagai tambahan, beberapa pin ini juga memiliki kekhususan fungsi antara lain:

1. TWI: pin A4 atau pin SDA dan A5 atau pin SCL. Support TWI communication menggunakan Wire library. Inilah pin sepasang lainnya di board UNO
2. AREF. Tegangan referensi untuk input analog. digunakan fungsi analog Reference().
3. Reset. Meneka jalur LOW untuk mereset mikrokontroler, terdapat tambahan tombol reset untuk melindungi salah satu blok.

c. Komunikasi

Arduino Uno memiliki fasilitas nomer untuk komunikasi dengan komputer atau hardware Arduino lainnya, atau dengan mikrokontroler. Pada ATmega328 menerjemahkan serial komunikasi UART TTL (5V) pada pin 0 (RX) dan 1 (TX). Pada ATmega16U2 serial komunikasinya dengan USB dan port virtual pada software di komputer. Perangkat lunak (firmware) 16U2 menggunakan driver standart USB COM dan tidak membutuhkan driver luar lainnya. Software Arduino bawaan telah menyertakan serial monitor yang sangat mudah membaca dan mengirim data dari dan ke Arduino. LED indikator TX dan RX akan kedip ketika data telah terkirim via koneksi USB-to-serial dengan USB pada komputer.

Software Serial library membolehkan banyak pin serial komunikasi pada Uno. ATmega328 juga support I2C (TWI) dan SPI *communication*. Software Arduino terbenam di dalamnya Wire library untuk memudahkan penggunaan bus I2C (Santoso, 2015).

2.6 Karya Tulis yang Mendahului (*State Of The Art*)

Dalam mengerjakan tugas akhir ini, peneliti menggunakan beberapa penelitian terdahulu sebagai panduan ataupun contoh yang nanti akan dijadikan acuan dan perbandingan dalam melakukan penelitian ini. Dalam state of the art ini terdapat 2 jurnal.

Tabel 2.1 Karya Tulis yang Mendahului

No	Judul	Penulis	Tahun	Perbedaan	Persamaan
1.	Alat Pendeteksi Tinggi Permukaan Air Secara Otomatis pada Bak Penampungan Air Menggunakan Sensor Ultrasonik Berdasarkan Mikrokontroler	Tegar Bhakti Prihantoro ¹ , Rizky Charli Wijaya Husni ²	2011	Menggunakan Mikrokontroler Atmega8535 dan dapat mengendalikan secara otomatis mesin pompa air.	Menggunakan Sensor Ultrasonik.
2	Perancangan Perangkat Pendeteksi Ketinggian Air Bak Pembesihan Ikan Nila Berbasis Mikrokontroler dan Web	Abdul Zulkifli	2014	Kontrol Perangkat menggunakan <i>Personal Computer</i> dan berbasis WEB.	Menggunakan Mikrokontroler Arduino Uno dan Sensor Ultrasonik.
3	Sistem <i>Monitoring</i> Tinggi Permukaan Air Pada Penampung Menggunakan Komunikasi Xbee Berdasarkan Mikrokontroler	Renaldi Habib Kholik	2017	Menggunakan mikrokontroler Arduino Uno dan Komunikasi <i>Wireless</i> Menggunakan XBee	Menggunakan Sensor Ultrasonik.

Berdasarkan data pada jurnal di atas. “Alat Pendeteksi Tinggi Permukaan Air Secara Otomatis pada Bak Penampungan Air Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler”, jurnal alumni STMIK GI MDP Palembang tahun 2011 oleh Tegar Bhakti Prihantoro¹, Rizky Charli Wijaya Husni². Dalam Tugas Akhir ini tersebut,. Alat yang dibuat ialah Pendeteksi Tinggi Permukaan Air menggunakan sensor ultrasonik berbasis mikrokontroler ATmega8535 yang dapat mengendalikan secara otomatis mesin pompa air. Dalam penelitian tersebut terdapat persamaan yaitu penulis sama-sama menggunakan sensor Ultrasonik untuk mengetahui ketinggian dari permukaan air.

Berdasarkan data diastugas akhirnya, “Perancangan Perangkat Pendeteksi Ketinggian Air Bak Pembelian Ikan nila Berbasis Mikrokontroler dan Web”, Jurnal Alumni Universitas Maritim Raja Ali Haji Tanjungpinang tahun 2014 oleh Abdul Zulkifli. Dalam Tugas Akhir tersebut Alat yang dibuat ialah Pendeteksi ketinggian air bak pembelian ikan nila berbasis mikrokontroler Arduino dan web. Dalam penelitian tersebut terdapat persamaan yaitu penulis sama-sama menggunakan sensor ultrasonik untuk mengetahui ketinggian dari permukaan air dan menggunakan Mikrokontroler Arduino.

Berdasarkan penelitian penelitian yang di atas “Sistem *Monitoring* Tinggi Permukaan Air Pada Penampung Menggunakan Komunikasi Xbee Berbasis Mikrokontroler”, laporan akhir ini nantinya akan dirancang dan dibangun oleh Renaldi Habib Kholik. Dalam laporan akhir ini akan dibuat dan dibangun sebuah Perangkat yang dapat mendeteksi tinggi permukaan air pada penampung, dengan inputan dari Sensor Ultrasonik, kemudian hasil dari pengukuran tersebut akan di transmisikan secara nirkabel menggunakan XBee untuk kemudian ditampilkan pada aplikasi *monitoring* PC.

BAB 3. METODE KEGIATAN

3.1 Tempat dan Waktu Pelaksanaan

Pelaksanaan dan pembuatan tugas akhir ini dilaksanakan di Kampus Politeknik Negeri Jember dan dikerjakan selama 6 bulan, dimulai dari bulan Februari - Juli 2017.

3.2 Alat dan Bahan Kegiatan

Alat dan bahan merupakan satu kesatuan yang penting dan harus diperhatikan dalam melakukan suatu kegiatan penelitian. Berikut alat dan bahan yang digunakan dalam pengerjaan tugas akhir ini.

3.2.1 Alat

Alat-alat yang dibutuhkan dalam pengerjaan tugas akhir ini dibagi menjadi 2 bagian yaitu perangkat keras dan perangkat lunak. a. Perangkat keras

Perangkat keras yang digunakan dalam pengerjaan tugas akhir ini antara lain:

Tabel 3.1 Daftar Alat Perangkat Keras yang Digunakan.

No.	Jenis	Spesifikasi
1	Solder	Solder 60 Watt
2	AVO Meter	Alat ukur Voltase, Arus, dsb
3	Gergaji	Gergaji besi
4	Penggaris	Penggaris Besi
6	Bor	Bor PCB dan Bor besi
7	Tang potong	Tang potong
8	Laptop	Asus A43S

b. Perangkat lunak

Perangkat lunak yang digunakan dalam pengerjaan tugas akhir antara lain :

Tabel 3.2 Daftar Alat Perangkat Lunak yang Digunakan.

No	Jenis	Spesifikasi
1	Aplikasi Arduino	Arduino IDE
2	Sistem Operasi	Windows 7 Home Premium
3	Aplikasi Penulisan Dokumen	Ms. Office 2010
4.	Aplikasi Visual Basic	Ms. Visual Studio 2010
5.	Aplikasi Konfigurasi XBee	Digi XCTU

3.2.2 Bahan

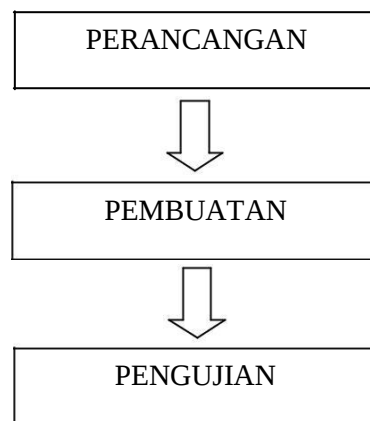
Bahan-bahan yang digunakan dalam pengerjaan tugas akhir ini antara lain :

Tabel 3.3 Daftar Bahan yang dibutuhkan

No	Jenis Bahan	Spesifikasi	Jumlah
1	Mikrokontroler Aduino	Arduino Uno R3	2 Buah
2	Sensor Ultrasonik	HC-SR04	1 Buah
3	<i>Push Button</i>	<i>Push Button</i>	1 Buah
4	Modul XBee	XBee S2B	2 Buah
5	Adapter XBee	Universal	2 Buah
6	Kabel Jumper	Male to female	Secukupnya
7	Papan PCB	PCB Matrix	Secukupnya
8	LED	LED 5mm	5 buah
9	Botol Air	Tinggi 20 cm	1 Buah
10	Box penampung air	Berbahan akrilik	1 Buah
11	Pompa Air	Pompa Aquarium 220V	1 Buah
12	Besi penyangga	Besi	Secukupnya

3.3 Metode Kegiatan

Metode kegiatan yang digunakan dalam proyek tugas akhir “Sistem Monitoring Tinggi Permukaan Air Pada Penampung Menggunakan Komunikasi XBee Berbasis Mikrokontroler” ini menggunakan beberapa tahapan metode. Adapun tahapan-tahapan metode tersebut dapat dijelaskan pada gambar di bawah ini.



Gambar 3.1 Diagram Metode Kegiatan

3.3.1 Perancangan

Pelaksanaan metode kegiatan ini terdiri dari tiga tahapan yaitu perancangan keseluruhan sistem, perancangan *hardware* dan perancangan *software*. Perancangan keseluruhan sistem dilakukan untuk membuat gambaran umum dari keseluruhan sistem sehingga dapat digunakan sebagai acuan dasar dalam pembuatan sistem. Dan perancangan *hardware* dilakukan agar mempermudah dalam langkah perakitan komponen dan pembuatan perangkat keras yang akan digunakan. Sedangkan perancangan *software* ialah terdiri dari *flowchart* atau diagram blok dari program yang akan digunakan pada sistem ini.

3.3.2 Pembuatan

Pada tahap ini dilakukan proses pembuatan alat Sistem Monitoring Tinggi Permukaan Air pada Penampung Menggunakan Komunikasi XBee Berbasis Mikrokontroler, berdasarkan perancangan yang telah dibuat sebelumnya. Pembuatan alat disini meliputi pembuatan *hardware*, konfigurasi XBee dan pembuatan *software*.

3.3.3 Pengujian

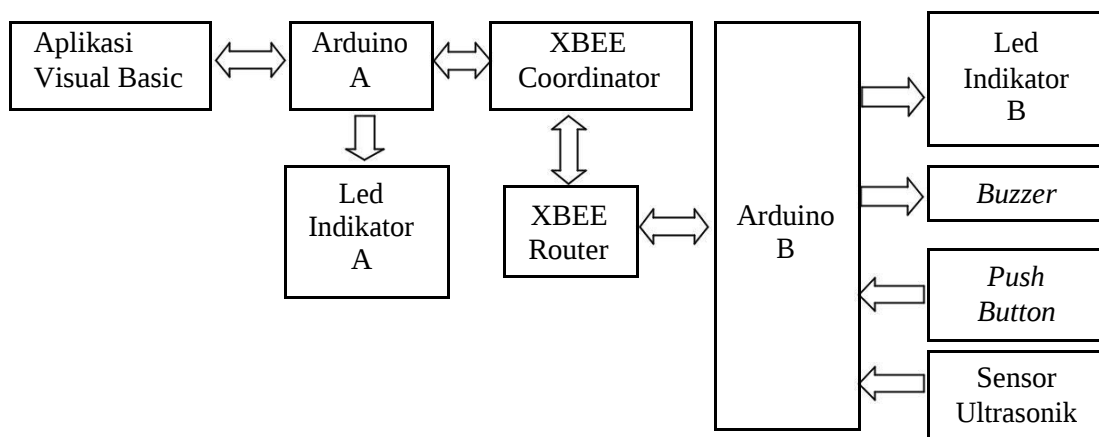
Dalam tahapan ini dilakukan pengujian terhadap alat yaitu uji coba koneksi XBee dan uji coba sensor ultrasonik. Uji coba koneksi XBee dilakukan untuk mengetahui seberapa jauh XBee mampu terhubung dengan kondisi terhalang oleh beberapa objek. Sedangkan uji coba sensor ultrasonik dilakukan untuk mengetahui tingkat keakuratan pembacaan sensor terhadap ketinggian permukaan air.

BAB 4. HASIL DAN PEMBAHASAN

4.1 Perancangan Alat

4.1.1 Perancangan Keseluruhan Sistem

Hasil dari perancangan sistem dideskripsikan pada Gambar 4.1.

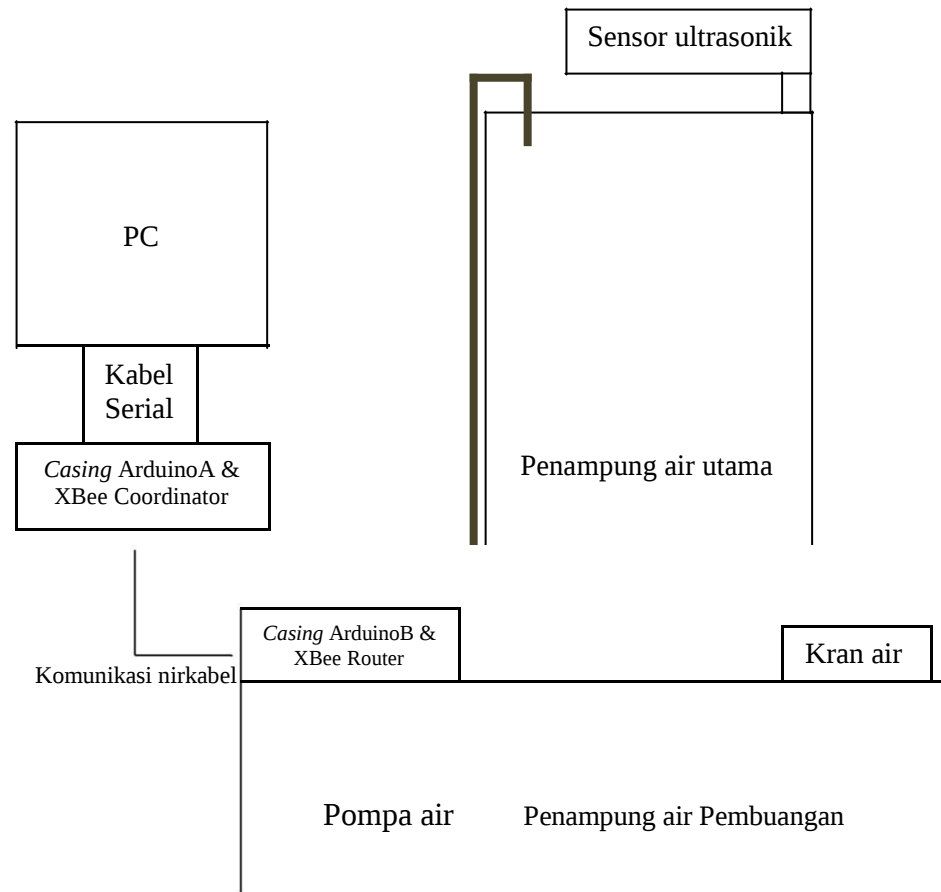


Gambar 4.1 Perancangan Sistem Kerja Alat

Perancangan Alat disini menggunakan beberapa komponen pokok antara lain Mikrokontroller Arduino Uno sebagai otak kontrol utama, XBee sebagai media komunikasi antara arduino A dan arduino B, led A sebagai indikator komunikasi data aktif, led indikator B sebagai indikator mode kontrol aktif, buzzer sebagai alarm, push button berfungsi untuk mengganti mode kontrol dan menyalakan pompa air, sensor ultrasonik untuk membaca tinggi permukaan air dan aplikasi visual basic sebagai output tampilan pembacaan sensor.

4.1.2 Perancangan *Hardware*

Hasil dari perancangan *hardware* dideskripsikan pada Gambar 4.2.



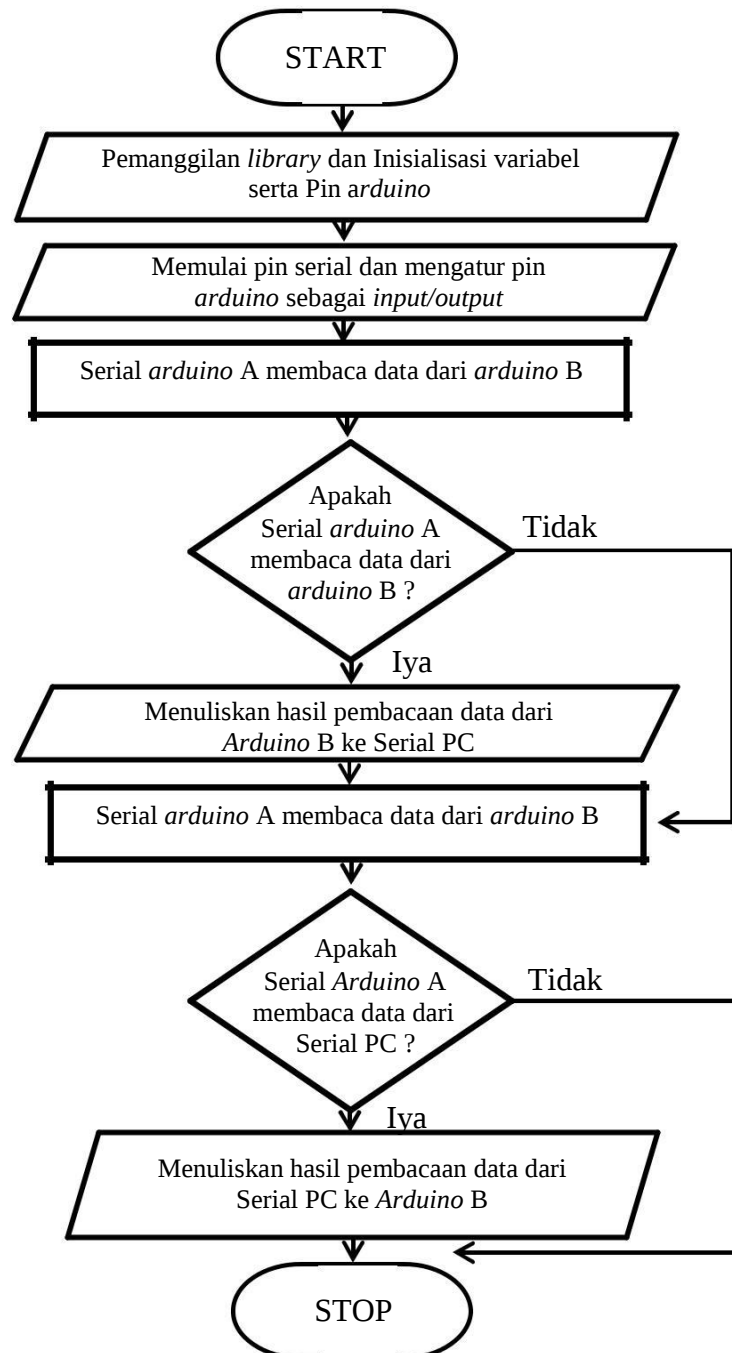
Gambar 4.2 Perancangan *Hardware*

Perancangan *hardware* disini menggunakan beberapa komponen pokok antara lain penampung air utama, penampung air pembuangan, *casing* arduino dan xbee yang semuanya berbahan akrilik, kran air yang berfungsi untuk menurunkan permukaan air, pompa air yang berfungsi untuk menaikkan permukaan air, kabel serial sebagai penghubung antara mikrokontroler arduino A dan PC, serta PC yang berfungsi sebagai *output* tampilan akhir dari pembacaan sensor.

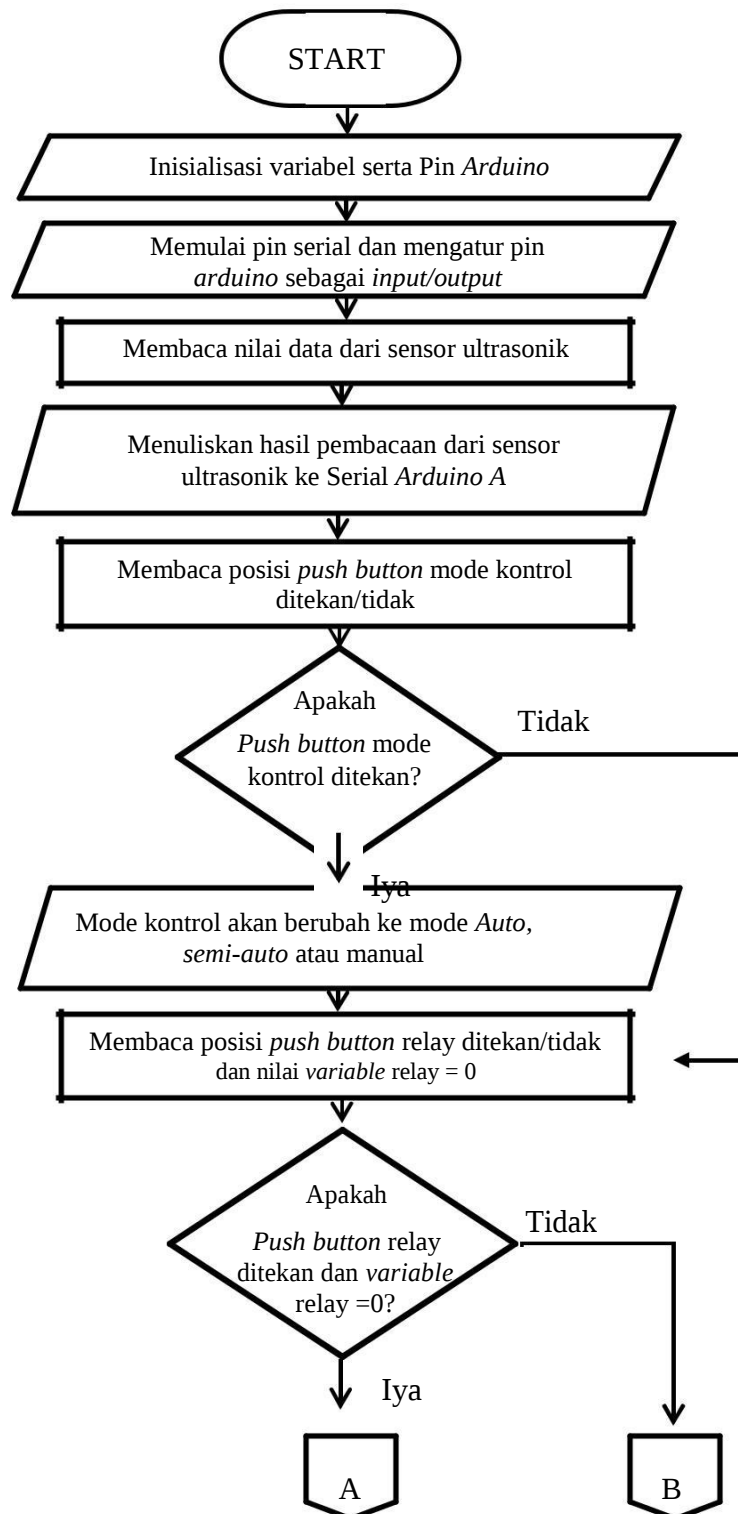
4.1.3 Perancangan Software

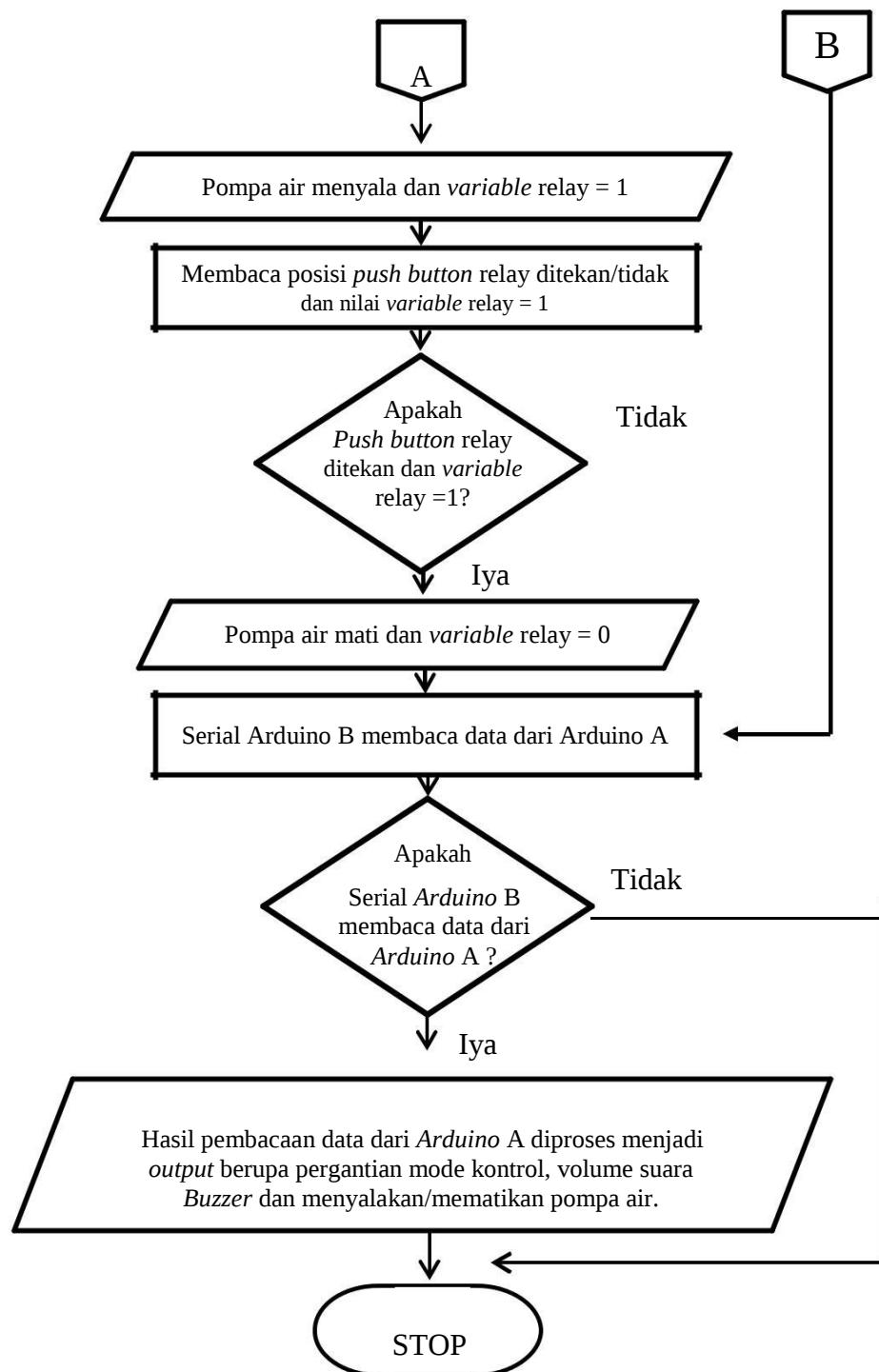
Untuk membuat program *Arduino* dan aplikasi *Visual Basic* diperlukan perancangan alur program dengan menggunakan *flowchart* sistem secara keseluruhan, Pembuatan *flowchart* ditampilkan pada gambar 4.3, 4.4 dan 4.5 :

a. Flowchart program Arduino A



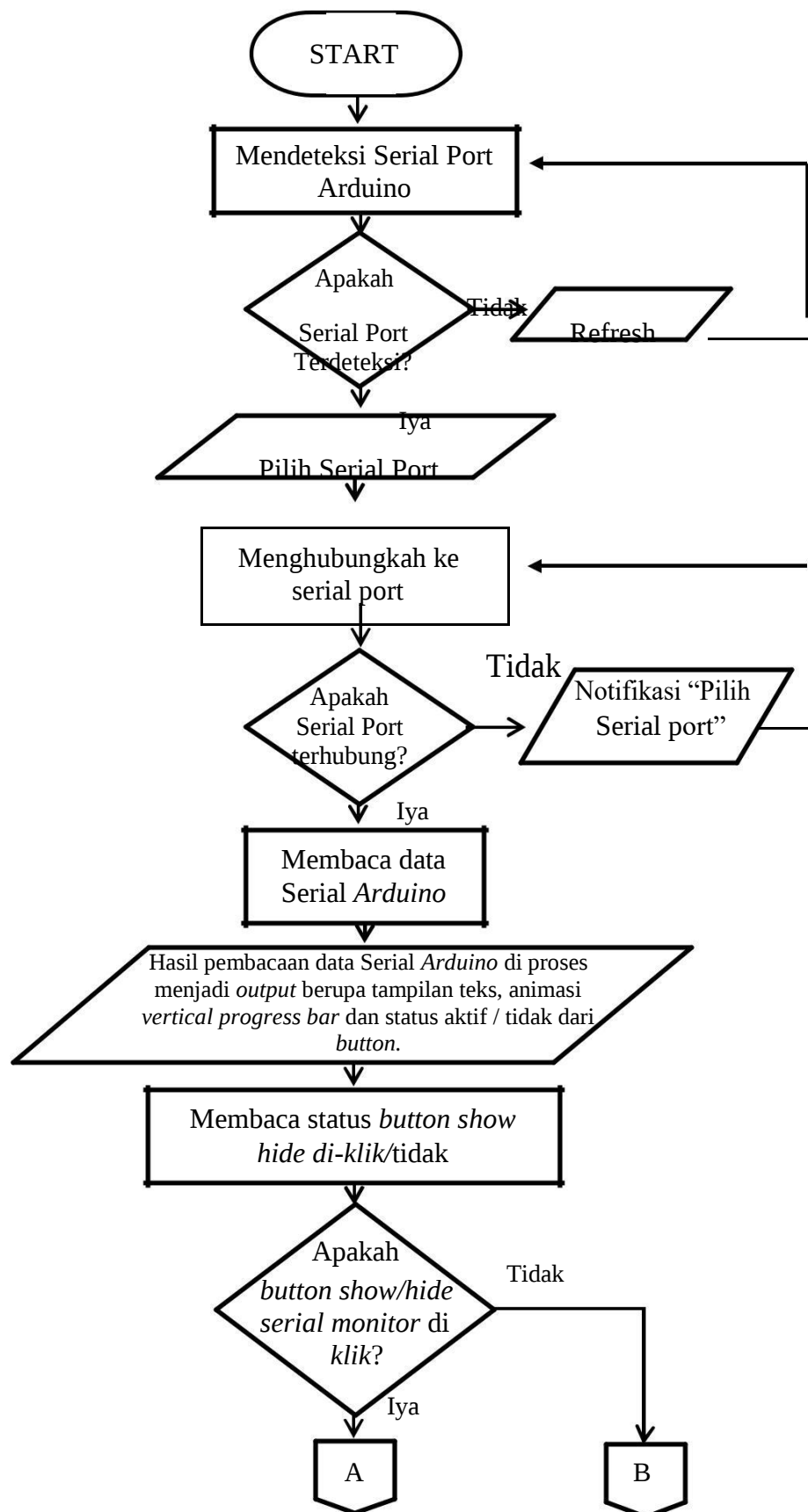
Gambar 4.3 Perancangan Flowchart Program Arduino A

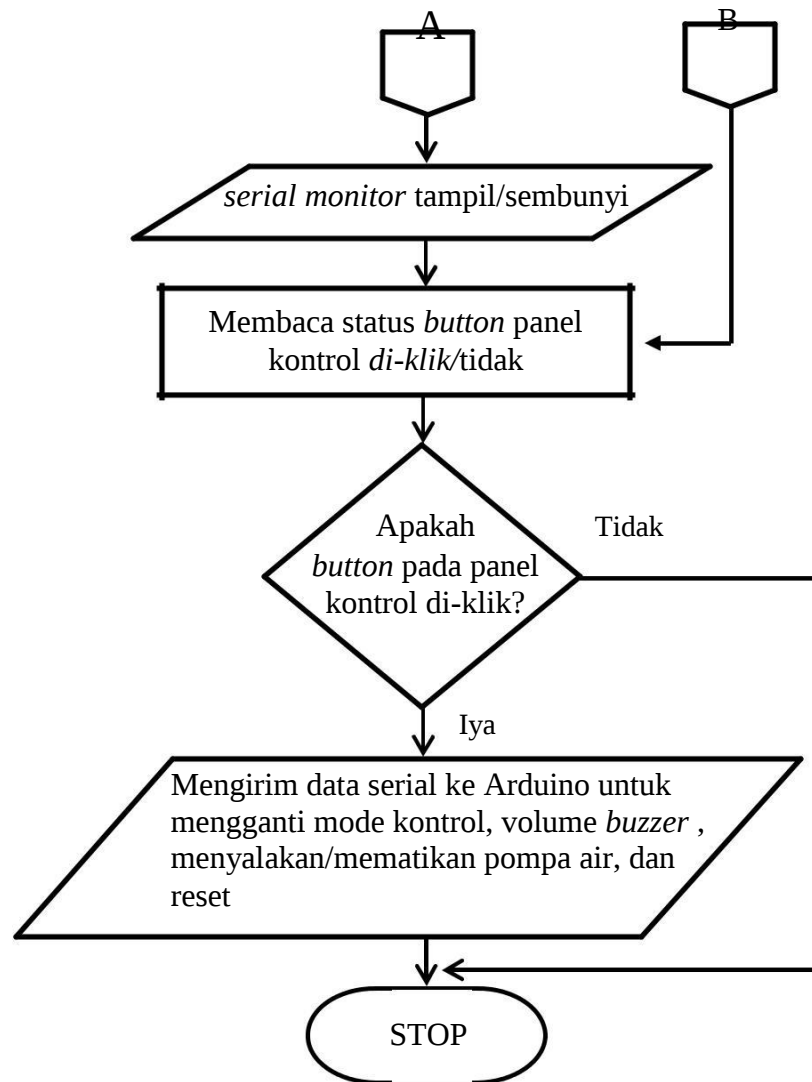
b. *Flowchart program Arduino B*



Gambar 4.4 Perancangan *Flowchart* Program Arduino B

c Flowchart aplikasi visual basic



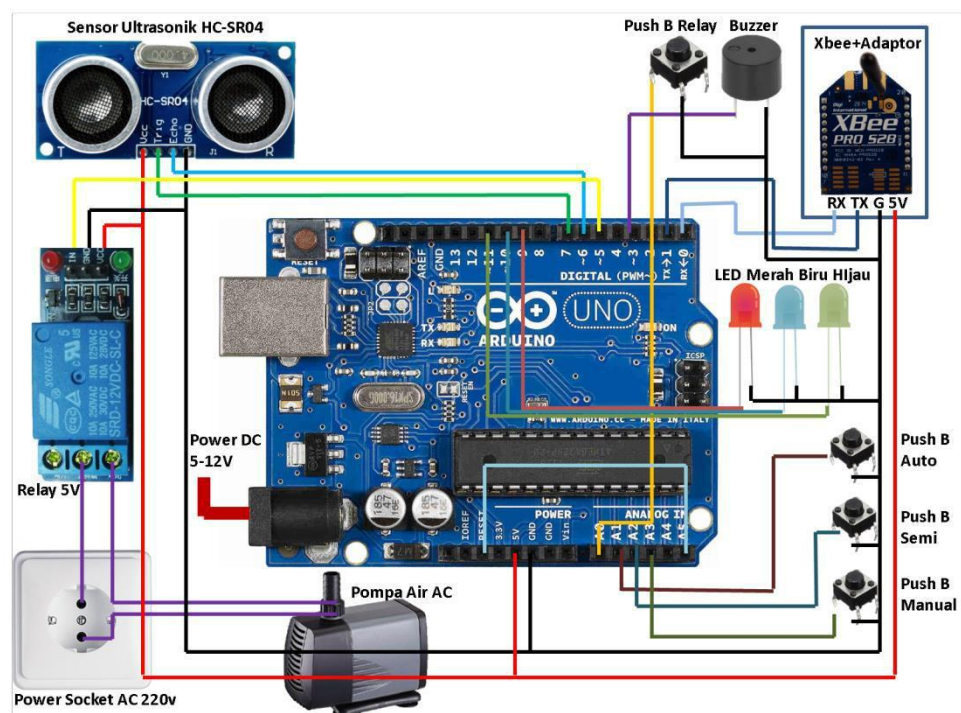


Gambar 4.5 Perancangan *Flowchart* aplikasi visual basic

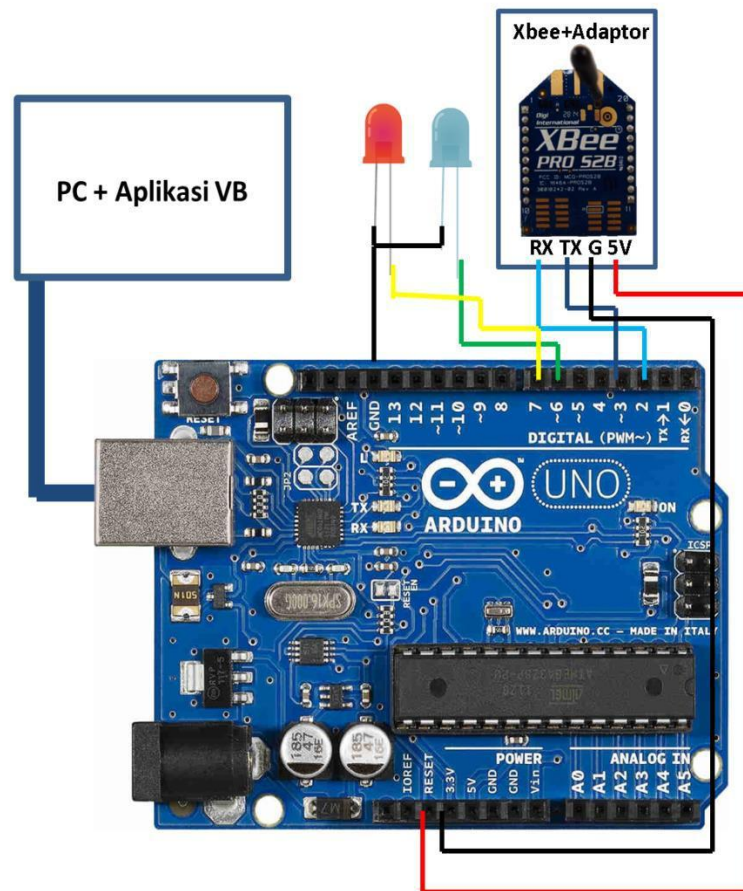
4.2 Pembuatan Alat

4.2.1 Pembuatan Hardware

Pada tahap ini, rancangan alat yang telah dibuat digunakan sebagai acuan untuk pembuatan *hardware*, pembuatan *hardware* disini menggunakan beberapa komponen pokok antara lain Mikrokontroler *Arduino Uno*, Sensor Ultrasonik, LED, *Push Button*, *Xbee*, Botol, dan Box plastik. Untuk kemudian dihubungkan antara satu dan lainnya seperti gambar 4.6 dan 4.7 :



Gambar 4.6 Rangkaian komponen Arduino B



Gambar 4.7 Rangkaian komponen Arduino A

a. Sensor Ultrasonik HC-SR04

Berfungsi untuk mendeteksi ketinggian permukaan air dengan memanfaatkan gelombang suara yang di hasilkan oleh sensor. *Transmitter* pada sensor akan memancarkan suatu gelombang suara yang kemudian akan memantul pada suatu medan padat ataupun cair, hasil dari pantulan tersebut akan diterima oleh *receiver* pada sensor untuk kemudian diketahui jarak yang telah ditempuh oleh gelombang suara yang telah dipantulkan.

b. Pompa Air AC

Pompa Air AC dengan dengan tegangan 220v digunakan untuk mengisi penampung dengan air atau menaikkan permukaan air.

c. Relay 5V

Berfungsi untuk mengendalikan sirkuit tegangan tinggi pada pompa air dengan bantuan dari signal tegangan rendah yang dihasilkan oleh mikrokontroler arduino, sehingga nyala dan mati pompa air bisa dikendalikan.

d. Buzzer

Berfungsi untuk memberi notifikasi jika ketinggian permukaan telah berada diatas atau dibawah ambang batas. Dan Buzzer juga akan berbunyi ketika *Push Button* ditekan.

e. Push Button

Berfungsi untuk mengganti mode kontrol dan menyalakan/mematikan pompa air tanpa menggunakan aplikasi *Visual Basic*.

f. LED

Berfungsi sebagai indikator mode kontrol yang sedang aktif pada *Arduino B*. Jika led hijau menyala artinya mode kontrol *manual* aktif, jika led biru menyala artinya mode kontrol *Semi-Auto* aktif, dan jika led merah menyala artinya mode kontrol *Auto* aktif. LED pada arduino A berfungsi sebagai indikator bahwa arduino sedang melakukan komunikasi data baik dengan arduino B maupun dengan PC.

g. XBee

Berfungsi untuk menghubungkan pin serial *Arduino A* dan *Arduino B* sehingga kedua *Arduino* bisa saling terhubung untuk mengirim dan menerima data.

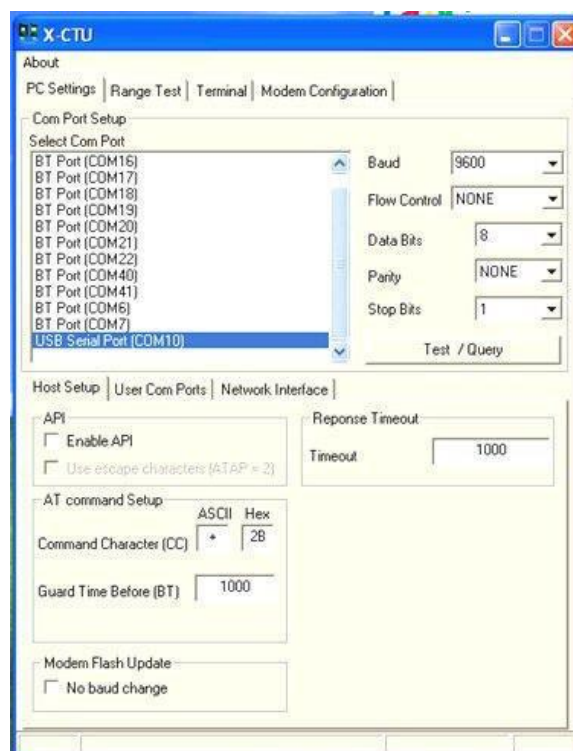
h. Aplikasi VB

Merupakan Perangkat lunak yang berfungsi untuk menampilkan hasil pembacaan sensor dengan tampilan grafis dan mengontrol Arduino B seperti menghidupkan dan mematikan pompa air, mengganti mode kontrol ke mode *auto*, *semi-auto*, *manual* serta mengatur volume suara *buzzer*.

4.2.2 Konfigurasi XBee

Agar kedua perangkat XBee bisa saling terhubung, maka diperlukan suatu langkah konfigurasi untuk mengatur fungsi dari masing-masing XBee, baik sebagai *Coordinator* maupun sebagai *Router*. XCTU adalah software khusus yang diperuntukkan untuk memberikan konfigurasi kepada XBee. a. Memilih Serial Port XBee

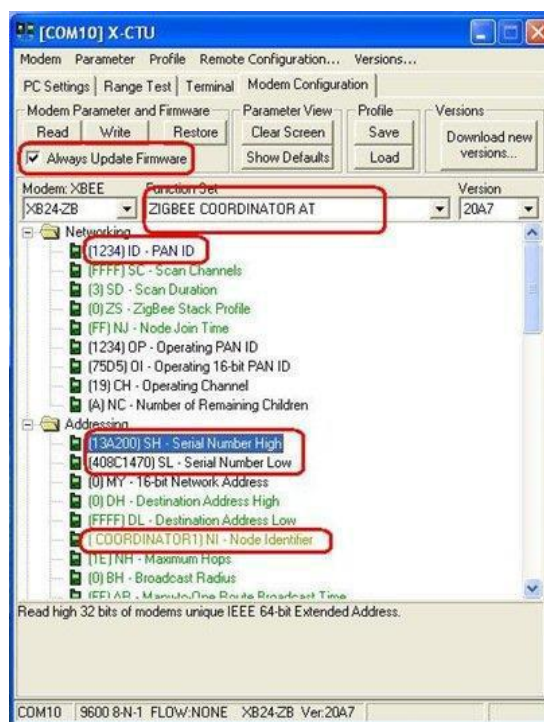
Pemilihan Serial Port bertujuan untuk memilih perangkat XBee mana yang akan dikonfigurasi seperti pada gambar 4.8.



Gambar 4.8 Pemilihan Serial Port XBee

b. Konfigurasi XBee sebagai *Coordinator*

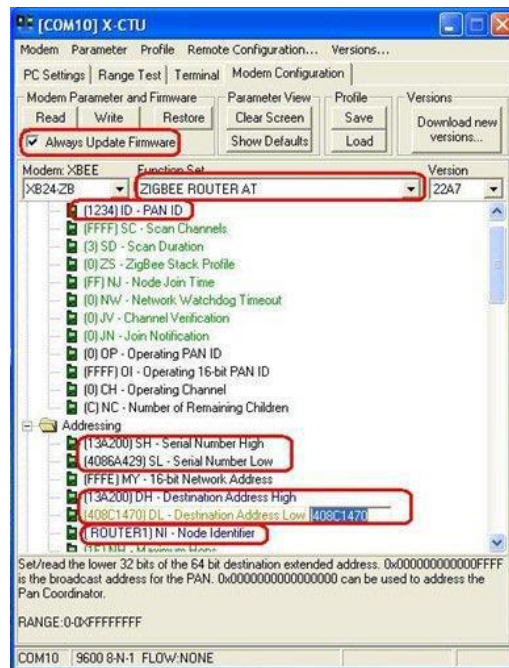
Dalam Setiap jaringan wajib memiliki satu buah *coordinator* yang harus selalu online. Sesuai dengan namanya *coordinator* berperan mengatur kondisi jaringan. *Coordinator* bertanggung jawab selama inisialisasi jaringan dan memilih berbagai parameter jaringan. Dalam suatu jaringan XBee hanya satu yang berfungsi sebagai *coordinator*. Untuk konfigurasi dapat dilihat pada gambar 4.9.



Gambar 4.9 Konfigurasi XBee *Coordinator*

c. Konfigurasi XBee sebagai *Router*

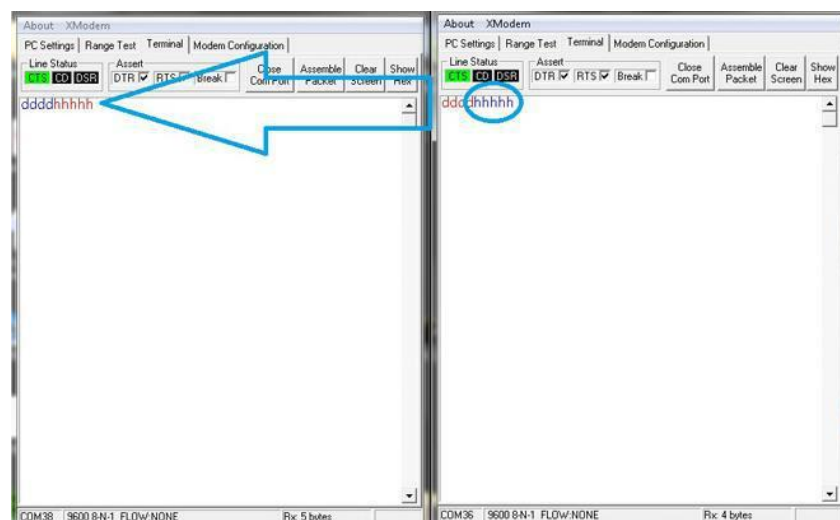
Router berfungsi sebagai perantara antar node, serta menyampaikan data dari perangkat lain. *Router* dapat terhubung dengan jaringan yang sudah ada. Dalam suatu jaringan XBee diizinkan untuk memiliki banyak router, selama PAN ID dari *router* dan *coordinator* sama maka dapat dimungkinkan kedua perangkat XBee akan saling terhubung. Konfigurasi XBee *router* dapat pada gambar 4.10.



Gambar 4.10 Konfigurasi XBee Router

d. Testing Koneksi XBee

Jika kedua perangkat XBee telah dikonfigurasi langkah selanjutnya adalah *testing* koneksi XBee untuk memastikan bahwa XBee telah terhubung dengan baik. Langkah ini dilakukan dengan membuka tab terminal pada kedua *windows* konfigurasi XBee. Setelah itu ketik suatu teks pada salah satu *windows* terminal Xbee, jika tersebut muncul pada *windows* terminal XBee yang lainnya maka dapat dipastikan XBee telah terhubung dengan baik seperti pada gambar 4.11

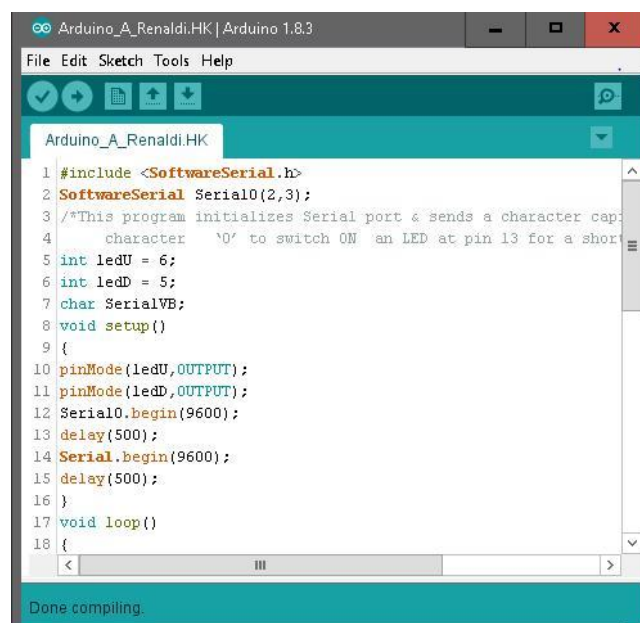


Gambar 4.11 Testing Koneksi XBee

4.2.3 Pembuatan Software

a. Pembuatan program mikrokontroler *Arduino*

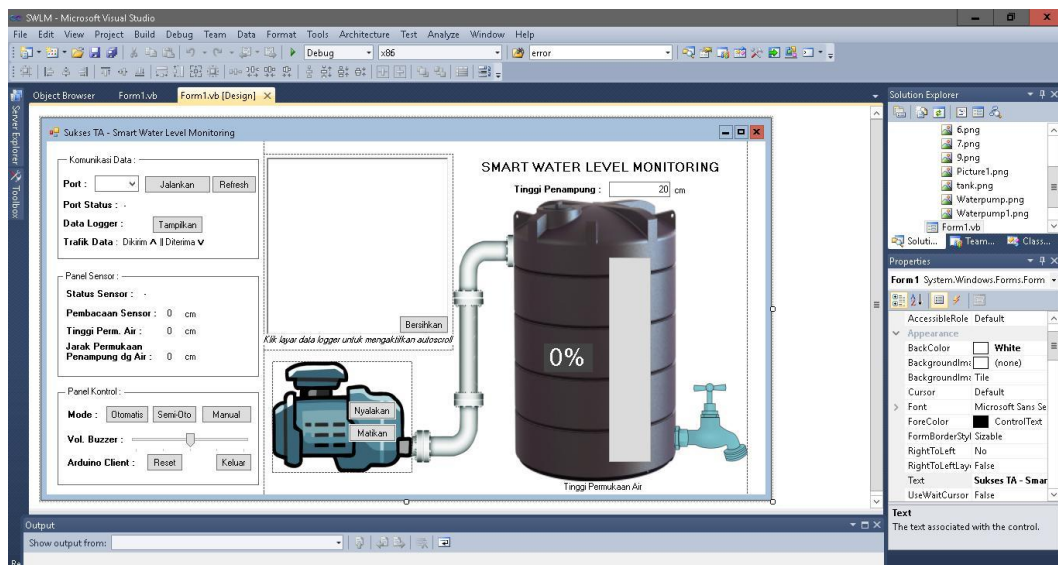
Dalam penggunaan *Arduino* sebagai mikrokontroler, terdapat aplikasi khusus yang memang sudah disiapkan untuk merancang program dengan basis bahasa pemrograman C yang nantinya bisa digunakan untuk memberikan perintah pada *board* arduino. Software ini sangat mudah dalam pengoperasiannya dimana juga bisa memudahkan kita untuk bisa langsung terhubung pada *Arduino*. Tampilan Aplikasi *Arduino IDE* dapat dilihat pada Gambar 4.12.



Gambar 4.12 Penulisan Program pada Arduino IDE

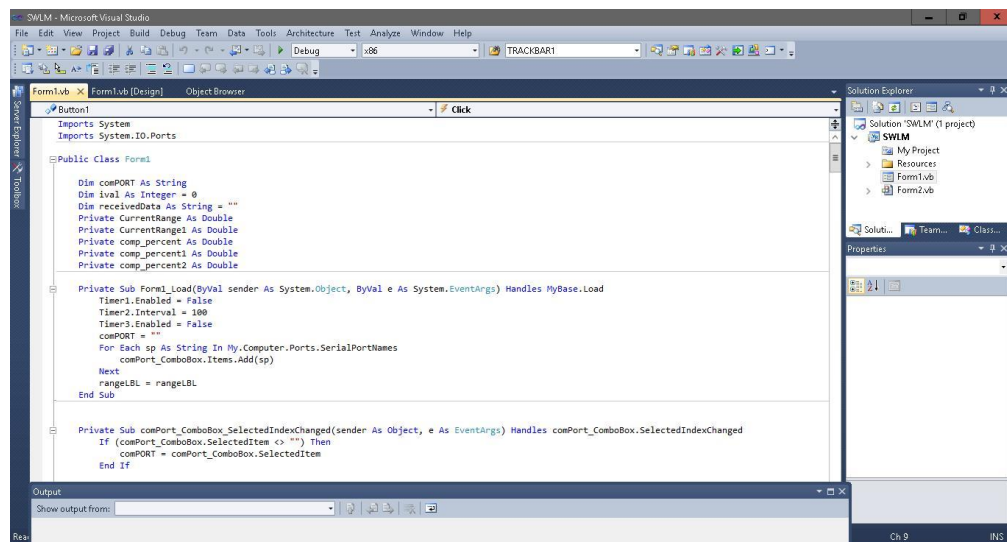
b. Pembuatan Aplikasi Visual Basic

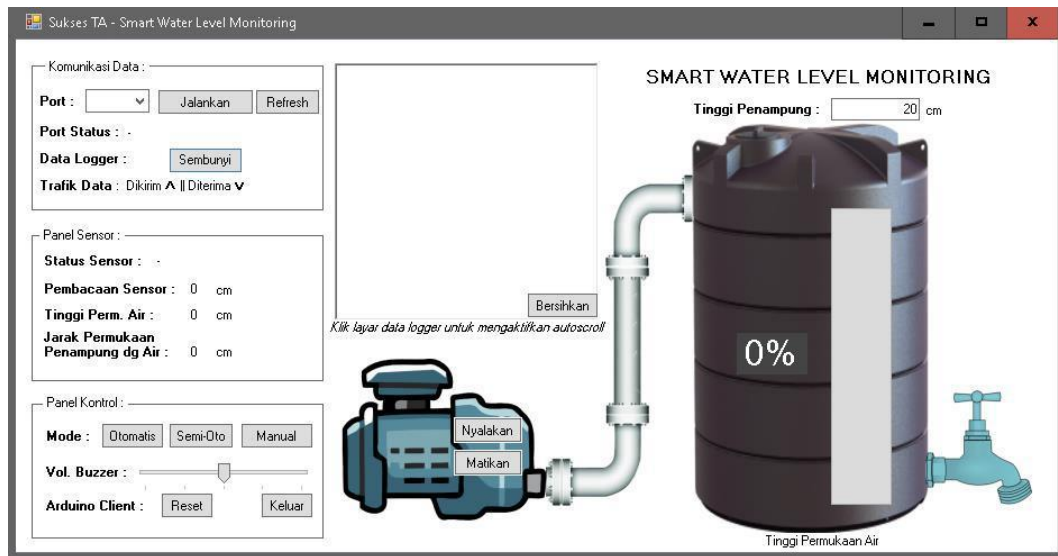
Pembuatan aplikasi Visual Basic pada proyek tugas akhir ini menggunakan aplikasi Microsoft Visual Studio 2010. Dengan tampilan GUI Membuat aplikasi ini mudah untuk dioperasikan. Pembuatan objek-objek dalam aplikasi pun sangat mudah dengan *drag and drop* objek pada menu toolbox seperti yang ditampilkan pada gambar 4.13.



Gambar 4.13 Pembuatan Objek Aplikasi Visual Basic

selanjutnya pada masing-masing objek yang telah dibuat diberikan perintah atau fungsi dengan *coding* berbasis *visual basic*. Seperti yang ditampilkan pada gambar 4.14.

Gambar 4.14 Penulisan *coding* Aplikasi Visual Basic



Gambar 4.15 Tampilan Aplikasi

Gambar 4.15 merupakan hasil akhir dari pembuatan aplikasi *visual basic* yang dimana pada aplikasi tersebut terdapat beberapa panel maupun *button* seperti pada panel komunikasi data yang berfungsi untuk mengatur komunikasi data serial antara aplikasi dan arduino.

1. Panel Komunikasi data terdiri dari menu port yang berfungsi untuk memilih port serial yang akan digunakan, port status untuk menampilkan status aktif atau tidak port yang telah dipilih, data logger untuk menampilkan data serial yang diterima dari arduino dan trafik data berfungsi untuk menampilkan aktivitas komunikasi data yang terjadi
2. Panel Sensor terdiri dari menu status sensor yang berfungsi untuk menampilkan kondisi sensor dalam keadaan baik atau tidak, pembacaan sensor untuk menampilkan nilai pembacaan sensor yang sebenarnya dan tinggi permukaan air serta jarak permukaan penampung dengan air yang semuanya dalam satuan cm.
3. Pada Panel Kontrol terdapat menu Mode yang merupakan mode kontrol yang akan dipakai dalam mengoperasikan sistem. Cara kerja pada mode otomatis ialah jika tinggi permukaan air berada dibawah ambang batas atau habis maka pompa akan menyala dengan sendirinya namun jika permukaan air berada diatas ambang batas atau air penuh maka pompa air

akan mati dengan sendirinya, pada penerapan atau implementasinya mode kontrol sangat berguna untuk penampung air yang tidak membutuhkan perawatan dan pemanatauan yang banyak karena semua kendali pompa air sudah berjalan dengan otomatis.

Cara kerja mode kontrol Manual ialah jika permukaan air dibawah ambang batas atau air habis dan jika permukaan air diatas ambang batas atau penuh maka pada kedua kondisi tersebut sistem hanya membunyikan *buzzer* dengan kendali pompa air untuk menghidupkan atau mematikan secara manual. Mode kontrol ini sangat berguna untuk keperluan perawatan penampung air seperti untuk penjernihan atau pengurusan air dengan cara meluapkan ataupun membuang air melalui kran.

Dan untuk menambah variasi pada mode kontrol sistem ini yang mungkin diperlukan untuk suatu keadaan tertentu maka dibuatlah mode kontrol Semi-Oto cara kerjanya ialah jika permukaan air dibawah ambang batas maka hanya akan membunyikan *buzzer* dan jika permukaan air berada diatas ambang batas atau air penuh maka pompa air akan mati.

Dan pada panel kontrol juga terdapat menu vol. *buzzer* yang berfungsi untuk mengatur volume dari *buzzer* dan Arduino client yang berfungsi untuk me-reset arduino client dan *button* keluar untuk keluar dari aplikasi.

4. Pada panel pompa air terdapat dua *button* yaitu nyalakan yang berfungsi untuk menyalakan pompa air dan matikan yang berfungsi untuk mematikan pompa air.
5. Pada panel penampung air terdapat *vertical progress bar* yang berfungsi untuk menampilkan persentase tinggi permukaan air dengan tampilan animasi dan menu tinggi penampung yang berfungsi untuk mengubah tinggi penampung air yang sedang digunakan dalam satuan cm.

4.3 Pengujian Alat

Pengujian alat bertujuan untuk mengetahui apakah alat sudah bekerja sesuai harapan dan perencanaan yang telah dibuat sebelumnya. Pengujian ini meliputi dua tahapan yaitu uji coba koneksi xbee dan uji coba sensor ultrasonik. Agar mendapatkan data yang valid, kegiatan pengujian ini dilakukan bersama dosen pembimbing utama yang telah berpengalaman dalam bidang mikrokontroler dan jaringan.

4.3.1 Uji Coba Koneksi XBee

Pada tahap ini dilakukan pengujian untuk mengetahui seberapa jauh komunikasi kedua XBee mampu terhubung agar dapat diketahui jangkauan efektif koneksi XBee.

Data hasil pengujian jarak koneksi XBee dapat dilihat pada tabel 4.3.

Tabel 4.3. Pengujian Jarak Koneksi XBee

No	Pengujian koneksi XBee		
	Percobaan Jarak	Objek Penghalang	Status Koneksi
1	10 Meter	1 Tembok	Terhubung Lancar
2	30 Meter	3 Tembok	Terhubung Lancar
3	32 Meter	4 Tembok	Terhubung Koneksi Lambat
4	35 Meter	5 Tembok	Tidak Terhubung
5	50 Meter	1 Tembok	Terhubung
6	70 Meter	1 Tembok	Terhubung
7	90 Meter	1 Tembok	Terhubung Koneksi Lambat
8	100 Meter	1 Tembok dan 1 Pohon	Tidak Terhubung
9	350 Meter	-	Terhubung
10	355 Meter	1 Pohon	Tidak Terhubung

Berdasarkan data pengujian pada tabel 4.3. bahwa objek penghalang sangat mempengaruhi komunikasi XBee. Dalam pengujian ini digunakan tembok sebagai objek penghalang karena keterbatasan akses dan penggunaan perangkat serta kondisi medan yang tidak memungkinkan untuk melakukan uji coba dengan maksimal.

4.3.2 Uji Coba Sensor Ultrasonik

Uji coba sensor ultrasonik dilakukan pada tahap ini untuk mengetahui tingkat keakuratan dari pembacaan sensor terhadap ketinggian permukaan air. Pengujian ini dibagi menjadi dua kondisi yang berbeda. Yaitu kondisi pertama dengan permukaan air yang tenang dan yang kedua dengan permukaan air yang tidak tenang atau bergelombang.

Berikut merupakan tabel pengujian sensor ultrasonik dapat dilihat pada tabel 4.4. dan tabel 4.5.

Tabel 4.4. Pengujian Sensor Ultrasonik dengan Kondisi Permukaan Air Tenang.

No	Kondisi Permukaan Air	Alat pengukuran ketinggian perm. Air		Error Pembacaan Sensor	Hasil
		Penggaris	sensor		
1	Permukaan air naik	2 cm	2 cm	0%	Akurat
2	Permukaan air naik	5 cm	5 cm	0%	Akurat
3	Permukaan air naik	10 cm	10 cm	0%	Akurat
4	Permukaan air naik	15 cm	15 cm	0%	Akurat
5	Permukaan air naik	20 cm	20 cm	0%	Akurat
6	Permukaan air turun	18 cm	18 cm	0%	Akurat
7	Permukaan air turun	15 cm	15 cm	0%	Akurat
8	Permukaan air turun	10 cm	10 cm	0%	Akurat
9	Permukaan air turun	5 cm	5 cm	0%	Akurat
10	Permukaan air turun	2 cm	2 cm	0%	Akurat

Tabel 4.5. Pengujian Sensor Ultrasonik dengan Kondisi Permukaan Air Tidak Tenang.

No	Kondisi Permukaan Air	Alat pengukuran ketinggian perm. air		Error Pembacaan Sensor	Hasil
		Penggaris	sensor		
1	Permukaan air naik	2 cm	3 cm	50%	Tidak Akurat
2	Permukaan air naik	5 cm	6 cm	20%	Tidak Akurat
3	Permukaan air naik	10 cm	9 cm	10%	Tidak Akurat
4	Permukaan air naik	15 cm	14 cm	6,6%	Tidak Akurat
5	Permukaan air naik	20 cm	19 cm	5%	Tidak Akurat
6	Permukaan air turun	18 cm	19 cm	5,5%	Tidak Akurat
7	Permukaan air turun	15 cm	16 cm	6,6%	Tidak Akurat
8	Permukaan air turun	10 cm	11 cm	10%	Tidak Akurat
9	Permukaan air turun	5 cm	6 cm	20%	Tidak Akurat
10	Permukaan air turun	2 cm	3 cm	50%	Tidak Akurat

Berdasarkan data pengujian pada tabel 4.4 dan 4.5 dapat disimpulkan bahwa tingkat keakuratan pembacaan sensor ultrasonik sangat dipengaruhi oleh kondisi permukaan air. Jika permukaan air tenang maka hasil pembacaan sensor akan akurat, dan jika permukaan air tidak tenang maka hasil pembacaan sensor tidak akurat atau terdapat error dari hasil pengukuran sebenarnya.

BAB 5 KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan beberapa tahap dalam pengerjaan tugas akhir ini, maka didapatkan kesimpulan sebagai berikut :

- a. Sistem Monitoring tinggi permukaan air pada penampung dapat menggunakan transmisi data nirkabel dengan *output* tampilan aplikasi *Visual Basic*.
- b. Komunikasi data nirkabel dengan menggunakan *XBee* dapat bekerja sesuai dengan kebutuhan yaitu komunikasi data *indoor* mampu mencapai jarak 30 m dan *outdoor* 1,5 km.
- c. Sistem kontrol *Arduino* dapat menggunakan aplikasi *Visual Basic* dan *Push Button*.
- d. Sensor ultrasonik mampu membaca tinggi permukaan air dengan presisi dengan syarat kondisi permukaan air harus tenang.

5.2 Saran

Adapun saran yang dapat diajukan untuk pengembangan *Sistem monitoring tinggi permukaan air pada penampung menggunakan komunikasi Xbee berbasis mikrokontroler* adalah sebagai berikut:

- a. Dianjurkan untuk menggunakan sensor pembaca jarak dengan kualitas yang lebih baik, seperti sensor LADAR (*Laser Detection And Ranging*) atau sensor sonar untuk mendeteksi kedalaman air.
- b. Disarankan untuk menambah sensor pendeteksi keberadaan air pada penampung, sehingga air yang masuk dan keluar juga dapat di monitoring.
- c. Disarankan untuk menambah sensor kekeruhan dan ph pada penampung untuk memonitoring kualitas air.

DAFTAR PUSTAKA

- Alif, M. 2017. *“Perancangan Sistem Komunikasi Menggunakan XBee Pro S2 IEEE 802.15.4”* Yogyakarta: Universitas Gadjah Mada.
- Hadyatma, A. 2015. *”Sistem Keamanan Rumah Nirkabel Berbasis Mikrokontroler menggunakan Modul XBee Pro”*. Salatiga: Universitas Kristen Satya Wacana.
- Komaludin, D. 2014 *“Prototype Pendeteksi Ketinggian Permukaan Air Maksimum dan Minimum menggunakan Sensor Ultrasonik Berbasis Mikrokontroler”* Bandung: STMIK LPKIA.
- Permana, F. 2009. *“Pembuatan Sistem Monitoring Ketinggian Air dengan Sensor Ultrasonik Berbasis Mikrokontroler ATmega8535”*. Semarang: Universitas Diponegoro.
- Prihantoro, T. 2011. *“Alat Pendeteksi Tinggi Permukaan Air Secara Otomatis pada Bak Penampung Air Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler”*. Palembang: STMIK GI MDP.
- Putro, M. 2014. *“Rancang Bangun Sistem Informasi Monitoring Pada Koperasi Setia Bhakti Wanita Berbasis Web”*. Surabaya: Stikom.
- Santoso, Hari. 2015. *“Belajar Arduino Untuk Pemula”*.
<http://www.elangsakti.com> [20 Agustus 2017].

- Susianah, T. 2011. *“Air Hujan Sebagai Alternatif Pemenuhan Kebutuhan Air Minum di Kecamatan Ranuyoso Kabupaten Lumajang”* Surabaya: Institut Teknologi Sepuluh Nopember.
- Yulianti, A. 2015. *“Prototype Alat Pengolahan Air Laut Menjadi Air Minum (Pengaruh Variasi Packing Filter Terhadap Kualitas Air dengan Analisa DO, Salinitas, dan Kontivitas)”*. Palembang: Politeknik Negeri Sriwijaya.
- Zulkifli, A. 2014. *“Perancangan Perangkat Pendeteksi Ketinggian Air pada Bak Pembenihan Ikan Nila Berbasis Mikrokontroler dan Web”*. Tanjungpinang: Universitas Maritin Raja Ali Haji.

Lampiran A : Source Code Program Arduino A

```
#include <SoftwareSerial.h>
SoftwareSerial Serial0(2,3);
int ledU = 6;
int ledD = 5;
char SerialVB;
void setup()
{
  pinMode(ledU,OUTPUT);
  pinMode(ledD,OUTPUT);
  Serial0.begin(9600);
  delay(500);
  Serial.begin(9600);
  delay(500);
}
void loop()
{
  if (Serial0.available() >0){
    Serial.write(Serial0.read());
  }

  else if (Serial.available())
  {
    SerialVB = Serial.read();
    Serial0.write(SerialVB);
    if(SerialVB == 'F'){
      analogWrite(ledD,10); //Led Rx On
      delay(50);
      analogWrite(ledD,0);
    }
    if(SerialVB == 'G'){
      analogWrite(ledD,0);
    }
    if(SerialVB == 'H'){
      analogWrite(ledU,10); //Led Tx On
    }
    if(SerialVB == 'I'){
      analogWrite(ledU,0);
    }
    if(SerialVB == 'M'){ //reset
      Serial0.write('R');
      digitalWrite(ledD,HIGH);
      digitalWrite(ledU,HIGH);
      delay(1000);
    }
  }
}
```

Lampiran B : Source Code Program Arduino B

```

unsigned long previousMillis = 0;
long interval = 1000;
unsigned long previousMillis1 = 0;
long interval1 = 2000;
unsigned long previousMillis2 = 0;
long interval2 = 1000;

const int echopin = 6;
const int trigpin = 7;
int relay=5;
int res = A5;
int buzzer = 3;
int pushR = A0;
int pushA = A1;
int pushS = A2;
int pushM = A3;
int ledA = 9;
int ledS = 10;
int ledM = 11;
long duration;
int distance;
int rel;
const int nyala = HIGH;
const int mati = LOW;
int pushRelay = LOW;
int autoState = LOW;
int semiState = HIGH;
int manuState = LOW;
int buzzState = HIGH;
int buzz;
int Nbuzz = 2;

String modePrint = "se";
String speedPrint = "me";
String relayPrint;
String buzzPrint = "du";

char readChar;
void setup()
{
    digitalWrite(res,HIGH);
    delay(200);
    pinMode(relay,OUTPUT);
    Serial.begin(9600);
    pinMode(trigpin, OUTPUT); // Sets the trigPin as an Output
    pinMode(echopin, INPUT); // Sets the echoPin as an Input

```

```

pinMode(pushR, INPUT);
pinMode(pushA, INPUT);
pinMode(pushS, INPUT);
pinMode(pushM, INPUT);
pinMode(ledA, OUTPUT);
pinMode(ledS, OUTPUT);
pinMode(ledM, OUTPUT);
pinMode(res, OUTPUT);
digitalWrite(pushR, HIGH);
digitalWrite(pushA, HIGH);

digitalWrite(pushS, HIGH);

digitalWrite(pushM, HIGH);
}
void loop()
{
  unsigned long currentMillis = millis();
  unsigned long currentMillis1 = millis();
  unsigned long currentMillis2 = millis();
  if(Serial.available()>0){
    readChar = Serial.read();
    if(readChar == 'A'){
      rel = nyala;
    }
    if(readChar == 'B'){
      rel = mati;
    }
    if(readChar == 'C'){
      autoState = HIGH;
      semiState = LOW;
      manuState = LOW;
      modePrint = "au";
    }
    if(readChar == 'D'){
      autoState = LOW;
      semiState = HIGH;
      manuState = LOW;
      modePrint = "se";
    }
    if(readChar == 'E'){
      autoState = LOW;
      semiState = LOW;
      manuState = HIGH;
      modePrint = "ma";
    }
    if(readChar == 'J'){
      interval = 2000;
      speedPrint = "lo";
    }
  }
}

```

```

    }
    if(readChar == 'K'){
        interval = 1000;
        speedPrint = "me";
    }
    if(readChar == 'L'){
        interval = 100;
        speedPrint = "fa";
    }
    if(readChar == 'M'){
        digitalWrite(res,LOW);
        delay(100);
    }
    if(readChar == '0'){
        Nbuzz = 0;
        buzzPrint = "no";
    }
    if(readChar == '1'){
        Nbuzz = 1;
        buzzPrint = "sa";
    }
    if(readChar == '2'){
        Nbuzz = 2;
        buzzPrint = "du";
    }
    if(readChar == '3'){
        Nbuzz = 3;
        buzzPrint = "ti";
    }
    if(readChar == '4'){
        Nbuzz = 4;
        buzzPrint = "em";
    }
}

if(digitalRead(pushS) == LOW){
    digitalWrite(buzzer,HIGH);
    delay(100);
    digitalWrite(buzzer,LOW);
    autoState = LOW;
    semiState = HIGH;
    manuState = LOW;
    modePrint = "se";
}
else if(digitalRead(pushM) == LOW){
    digitalWrite(buzzer,HIGH);
    delay(100);
    digitalWrite(buzzer,LOW);

```

```

        autoState = LOW;
        semiState = LOW;
        manuState = HIGH;
        modePrint = "ma";
    }
    else if(digitalRead(pushA) == LOW){
        digitalWrite(buzzer,HIGH);
        delay(100);
        digitalWrite(buzzer,LOW);
        autoState = HIGH;
        semiState = LOW;
        manuState = LOW;
        modePrint = "au";
    }
    if (currentMillis - previousMillis >= interval) {
        // save the last time you blinked the
        LED previousMillis = currentMillis;
        digitalWrite(trigpin, LOW);
        delayMicroseconds(2);
        // Sets the trigPin on HIGH state for 10 micro
        seconds digitalWrite(trigpin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigpin, LOW);
        // Reads the echoPin, returns the sound wave travel time
        in microseconds
        duration = pulseIn(echopin, HIGH);
        // Calculating the distance
        int distance= duration*0.034/2-5;
        Serial.print(distance);
        Serial.println( " cm");
        //-----
        if(autoState == HIGH){
            analogWrite(ledA,10);
            analogWrite(ledS,0);
            analogWrite(ledM,0);
            buzzState = LOW;
            if (distance <= 0){
                rel = mati;
                delay(100);
            }
            if (distance >= 18){
                rel = nyala;
                delay(100);
            }
        }
    }
    //-----
    if(semiState == HIGH){
        analogWrite(ledA,0);
        analogWrite(ledS,10);
    }

```

```

    analogWrite(ledM,0);
    if (distance <= 0){
        rel = mati;
    }

    if (distance >= 17 ){
        buzzState = HIGH;
    }

    if (distance < 17){
        buzzState = LOW;
    }

    if(digitalRead(pushR)==LOW){
        rel = nyala;
        buzzState = HIGH;
        delay(200);
    }
}
//-----
    if(manuState == HIGH){
        analogWrite(ledA,0);
        analogWrite(ledS,0);
        analogWrite(ledM,10);
        if (distance >= 2 || distance <= 16){
            buzzState=LOW;
            delay(100);
        }
    }
/*
    if (distance<=2){
        buzzState=HIGH;
        delay(100);
    }*/
    if (distance>=17&&rel==mati||distance<=1&&rel==nyala){
        buzzState=HIGH;
        delay(100);
    }

    else{
        buzzState=LOW;
        delay(100);
    }
    if(digitalRead(pushR)==LOW&&pushRelay==LOW){
        digitalWrite(buzzer,HIGH);
        delay(100);
        digitalWrite(buzzer,LOW);
        rel = nyala;
        delay(500);
        pushRelay=HIGH;
    }
    else if(digitalRead(pushR)==LOW&&pushRelay==HIGH){
        digitalWrite(buzzer,HIGH);
        delay(100);
        digitalWrite(buzzer,LOW);
    }

```

```

        rel = mati;
        delay(500);
        pushRelay=LOW;
    }
}
//-----
}
if (currentMillis1 - previousMillis1 >= interval1) {
    // save the last time you blinked the
    LED previousMillis1 = currentMillis1;
    Serial.print(modePrint);
    Serial.print(" ");
    Serial.print(speedPrint);
    Serial.print(" ");
    Serial.print(relayPrint);
    Serial.print(" ");
    Serial.println(buzzPrint);
}

if(rel == nyala){
    digitalWrite(relay,HIGH);
    relayPrint="on";
}
if(rel == mati){
    digitalWrite(relay,LOW);
    relayPrint="off";
}
if(buzzState == HIGH){
    if (currentMillis2 - previousMillis2 >= interval2) {
        // save the last time you blinked the
        LED previousMillis2 = currentMillis2;
        if (buzz == 0) {
            buzz = Nbuzz*63;
        } else {
            buzz = 0;
        }
        analogWrite(buzzer, buzz);
    }
}
if(buzzState == LOW){
    analogWrite(buzzer, 0);
}
}

```

Lampiran C : Source Code Program Aplikasi Visual Basic

```
Imports System

Imports System.IO.Ports

Public Class Form1

    Dim comPORT As String
    Dim ival As Integer = 0
    Dim receivedData As String = ""
    Private CurrentRange As Double
    Private CurrentRange1 As Double
    Private comp_percent As Double
    Private comp_percent1 As Double
    Private comp_percent2 As Double

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Timer1.Enabled = False
        Timer2.Interval = 100
        Timer3.Enabled = False
        comPORT = ""
        For Each sp As String In My.Computer.Ports.SerialPortNames
            comPort_ComboBox.Items.Add(sp)
        Next
        rangeLBL = rangeLBL
    End Sub

    Private Sub comPort_ComboBox_SelectedIndexChanged(sender As
Object, e As EventArgs) Handles comPort_ComboBox.SelectedIndexChanged
        If (comPort_ComboBox.SelectedItem <> "") Then
            comPORT = comPort_ComboBox.SelectedItem
        End If
    End Sub

    Private Sub connect_BTN_Click(sender As Object, e As EventArgs)
Handles connect_BTN.Click
        If (connect_BTN.Text = "Connect") Then
            If (comPORT <> "") Then
                SerialPort1.Close()
                SerialPort1.PortName = comPORT
                SerialPort1.BaudRate = 9600
                SerialPort1.DataBits = 8
                SerialPort1.Parity = Parity.None
                SerialPort1.StopBits = StopBits.One
                SerialPort1.Handshake = Handshake.None
                SerialPort1.Encoding = System.Text.Encoding.Default
'very important!
                SerialPort1.ReadTimeout = 10000
                SerialPort1.Open()
                connect_BTN.Text = "Disconnect"
                Timer1.Enabled = True
                Timer3.Enabled = True
                Timer_LBL.Text = "ON"
            End If
        End If
    End Sub

End Class
```

```

Else
    MsgBox("Pilih COM port Terlebih dahulu")
End If
Else
    SerialPort1.Close()
    connect_BTN.Text = "Connect"
    Timer1.Enabled = False
    Timer3.Enabled = False
    Timer_LBL.Text = "OFF"
End If
If (connect_BTN.Text = "Connect") Then
    Label18.Visible = False
    Label17.Visible = False
End If
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    receivedData = ReceiveSerialData()
    Dim s As String = receivedData
    If s.Contains(" cm") Then
        s = s.Replace("cm", "")
        If IsNumeric(s) Then
            CurrentRange = CDb1(s)
            rangeLBL.Invoke(New UpdateRangeBoxDelegate(AddressOf
UpdateRangeBox), CurrentRange.ToString)
        End If
    End If

    Dim a As String = receivedData
    If a.Contains("cm") Then
        Label18.Visible = True
        SerialPort1.Write("F")
    ElseIf (Label18.Visible = True) Then
        Label18.Visible = False
        SerialPort1.Write("G")
    End If
    Dim m As String = receivedData
    If m.Contains("au") Then
        Button4.ForeColor = Color.White
        Button4.BackColor = Color.Green

        Button2.ForeColor = Color.Black
        Button2.BackColor = Color.LightGray

        Button5.ForeColor = Color.Black
        Button5.BackColor = Color.LightGray
        Button1.Visible = False
        Button6.Visible = False
        Label17.Visible = False
        SerialPort1.Write("I")
    End If
    If m.Contains("se") Then
        Button2.ForeColor = Color.White
        Button2.BackColor = Color.Green

        Button4.ForeColor = Color.Black
        Button4.BackColor = Color.LightGray

```

```

        Button5.ForeColor = Color.Black
        Button5.BackColor = Color.LightGray
        Button1.Visible = True
        Button6.Visible = False
        Label17.Visible = False
        SerialPort1.Write("I")
    End If
    If m.Contains("ma") Then
        Button5.ForeColor = Color.White
        Button5.BackColor = Color.Green

        Button4.ForeColor = Color.Black
        Button4.BackColor = Color.LightGray

        Button2.ForeColor = Color.Black
        Button2.BackColor = Color.LightGray
        Button1.Visible = True
        Button6.Visible = True
        Label17.Visible = False
        SerialPort1.Write("I")
    End If

    If m.Contains("on") Then
        Timer2.Start()
        Button1.ForeColor = Color.White
        Button1.BackColor = Color.Green

        Button6.ForeColor = Color.Black
        Button6.BackColor = Color.LightGray
    End If
    If m.Contains("off") Then
        Timer2.Stop()
        Button6.ForeColor = Color.White
        Button6.BackColor = Color.Green

        Button1.ForeColor = Color.Black
        Button1.BackColor = Color.LightGray
    End If

    comp_percent = ((Val(rangeLBL.Text) / Val(TextBox1.Text) *
100) - 100) * (1 - 2)
    PersenLBL.Invoke(New UpdateProgressDelegate(AddressOf
UpdateProgress), comp_percent.ToString + "%")
    comp_percent = ((Val(rangeLBL.Text) / Val(TextBox1.Text) *
100) - 100) * (1 - 2)
    pbVer1.Invoke(New UpdatehighBoxDelegate(AddressOf UpdateBar),
comp_percent.ToString)
    comp_percent2 = ((Val(rangeLBL.Text) - Val(TextBox1.Text)) *
(1 - 2))
    pbVer1.Invoke(New UpdateBarDelegate(AddressOf UpdatehighBox),
comp_percent2.ToString)
    comp_percent1 = CurrentRange + 4
    pbVer1.Invoke(New UpdatesenBoxDelegate(AddressOf
UpdatesenBox), comp_percent1.ToString)

    '
    RichTextBox1.Text &= receivedData
    RichTextBox1.AppendText(receivedData)
    'RichTextBox1.SelectionStart = RichTextBox1.TextLength

```

```

        'RichTextBox1.ScrollToCaret()
    End Sub

    Delegate Sub UpdatehighBoxDelegate(ByVal t As String)
    Public Sub UpdatehighBox(ByVal t As String)
        highLBL.Text = t
    End Sub

    Delegate Sub UpdatesenBoxDelegate(ByVal t As String)
    Public Sub UpdatesenBox(ByVal t As String)
        senLBL.Text = t
    End Sub

    Delegate Sub UpdateRangeBoxDelegate(ByVal t As String)
    Public Sub UpdateRangeBox(ByVal t As String)
        rangeLBL.Text = t
    End Sub

    Delegate Sub UpdateProgressDelegate(ByVal t As String)
    Public Sub UpdateProgress(ByVal t As String)
        PersenLBL.Text = t
    End Sub

    Delegate Sub UpdateBarDelegate(ByVal t As String)
    Public Sub UpdateBar(ByVal t As Integer)
        If (t < 0) Then
            pbVer1.Value = 0
            Label11.Text = "Sensor Error!!!"
            Label11.ForeColor = Color.Red
        Else
            pbVer1.Value = t
            Label11.Text = "Sensor OK"
            Label11.ForeColor = Color.Green
        End If
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick

        If PictureBox2.Visible = True Then 'gambar pompa
            bergerak PictureBox2.Visible = False
        Else
            PictureBox2.Visible = True
        End If
    End Sub

    Function ReceiveSerialData() As String
        Dim Incoming As String
        Try
            Incoming = SerialPort1.ReadExisting()
            If Incoming Is Nothing Then
                Return "nothing" & vbCrLf
            Else
                Return Incoming
            End If
        Catch ex As TimeoutException
            Return "Error: Serial Port read timed out."
        End Try
    End Function

```

```

        Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
            Me.Close()

        End Sub

        Private Sub Button6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button6.Click
            If (comPORT <> "") Then
                SerialPort1.Write("BBBBB") 'Mematikan Pompa
                SerialPort1.Write("H") 'Led Serial Write on
                Label17.Visible = True 'Indikator Serial Write
            Else
                MsgBox("Pilih COM port Terlebih dahulu")
            End If

        End Sub

        Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
            If (comPORT <> "") Then
                SerialPort1.Write("AAAAA") 'Menghidupkan Pompa
                SerialPort1.Write("H") 'Led Serial write on
                Label17.Visible = True 'indikator serial write
            Else
                MsgBox("Pilih COM port Terlebih dahulu")
            End If

        End Sub

        Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
            If (comPORT <> "") Then
                SerialPort1.Write("CCCCC") 'Mode kontrol auto
                SerialPort1.Write("H") 'led serial write on
                Label17.Visible = True 'indikator serial write
            Else
                MsgBox("Pilih COM port Terlebih dahulu")
            End If

        End Sub

        Private Sub Button2_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button2.Click
            If (comPORT <> "") Then
                SerialPort1.Write("DDDDD") 'Mode Kontrol semi
                auto SerialPort1.Write("H") 'Led Serial Write on
                Label17.Visible = True 'Indikator Serial Write
            Else
                MsgBox("Pilih COM port Terlebih dahulu")
            End If

        End Sub

        Private Sub Button5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button5.Click
            If (comPORT <> "") Then
                SerialPort1.Write("EEEEEE") 'mode kontrol manual
                SerialPort1.Write("H") 'Led Serial Write on
                Label17.Visible = True 'Indikator Serial Write
            Else
                MsgBox("Pilih COM port Terlebih dahulu")
            End If

        End Sub

```

```

        End If
    End Sub

    Private Sub Refresh_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Refresh.Click
        Application.Restart()
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        If (comPORT <> "") Then
            SerialPort1.Write("J") 'serial read speed slow
            SerialPort1.Write("H") 'Led Serial Write on
            Label17.Visible = True 'Indikator Serial Write
        Else
            MsgBox("Pilih COM port Terlebih dahulu")
        End If
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        If (comPORT <> "") Then
            SerialPort1.Write("KKKKK") 'serial read speed medium
            SerialPort1.Write("H") 'Led Serial Write on
            Label17.Visible = True 'Indikator Serial Write
        Else
            MsgBox("Pilih COM port Terlebih dahulu")
        End If
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        If (comPORT <> "") Then
            SerialPort1.Write("LLLLL") 'serial read speed high
            SerialPort1.Write("H") 'Led Serial Write on
            Label17.Visible = True 'Indikator Serial Write
        Else
            MsgBox("Pilih COM port Terlebih dahulu")
        End If
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button14.Click
        If (comPORT <> "") Then
            SerialPort1.Write("MMMMM") 'reset arduino client
            SerialPort1.Write("H") 'Led Serial Write on
            Label17.Visible = True 'Indikator Serial Write
        Else
            MsgBox("Pilih COM port Terlebih dahulu")
        End If
    End Sub

    Private Sub TrackBar1_Scroll(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TrackBar1.Scroll
        If (comPORT <> "") Then
            'Label26.Text = String.Format("volume:{0}",
            TrackBar1.Value)
            SerialPort1.Write(TrackBar1.Value) 'ganti volume +-
            SerialPort1.Write("H") 'Led Serial Write on

```

```

        Label17.Visible = True 'Indikator Serial Write
    Else
        MsgBox("Pilih COM port Terlebih dahulu")
    End If
End Sub

Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick

    End Sub

Private Sub Button11_Click_1(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles Button11.Click
    RichTextBox1.Text = " " 'serial monitor text clear
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button10.Click
    If (Button10.Text = "Show") Then 'show and hide serial
monitor
        Panel4.Visible = True
        Button10.Text = "Hide"
    ElseIf (Button10.Text = "Hide") Then
        Panel4.Visible = False
        Button10.Text = "Show"
    End If
End Sub
End Class

```

Lampiran D : Lembar Revisi

8/29/2017

Lembar Revisi Seminar/Sidang Tugas Akhir



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI JEMBER

LEMBAR REVISI
UJIAN TUGAS AKHIR

Nama : RENALDI HABIB KHOLIK
NIM : E32140964
Program Studi : D3-TEKNIK KOMPUTER
Jurusan : TEKNOLOGI INFORMASI
Judul Tugas Akhir : SISTEM MONITORING TINGGI PERMUKAAN AIR PADA PENAMPUNG MENGGUNAKAN KOMUNIKASI XBEE BERBASIS MIKROKONTROLER

Tanggal Ujian : RABU/06-09-2017
 Tempat Ujian : GEDUNG TEKNOLOGI INFORMASI-R. UJIAN 2
 Batas Akhir Revisi : -

[illegible]

Koordinator Tugas Akhir,

~~CONFIDENTIAL~~

Yogiswara, S.T. MT
NIP. 197009292003121001

Jember, 29-08-2017
Ketua Penguji,

Jan

Surateno, S.Kom, M.Kom
NIP. 197907032003121001

8/29/2017

Lembar Revisi Seminar/Sidang Tugas Akhir

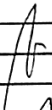
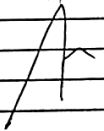
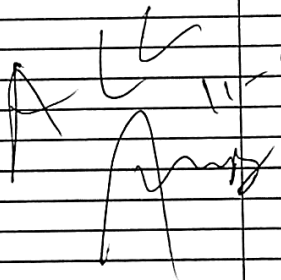


KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI JEMBER

LEMBAR REVISI
UJIAN TUGAS AKHIR

Nama : RENALDI HABIB KHOLIK
NIM : E32140964
Program Studi : D3-TEKNIK KOMPUTER
Jurusan : TEKNOLOGI INFORMASI
Judul Tugas Akhir : SISTEM MONITORING TINGGI PERMUKAAN AIR PADA PENAMPUNG MENGGUNAKAN KOMUNIKASI XBEE BERBASIS MIKROKONTROLER

Tanggal Ujian : RABU/06-09-2017
 Tempat Ujian : GEDUNG TEKNOLOGI INFORMASI-R. UJIAN 2
 Batas Akhir Revisi : -

No	Revisi	Paraf Dosen
1	cek data fokus	
2	cek daftar pustaka	
3	perbaikan data ujian	
		
		11-9-2019

Koordinator Tugas Akhir,

Yogiswari, S. H. MT
NIP. 197009292003121001

Jember, 29-08-2017
Anggota Penguji ,

Agus Hariyanto, ST, M.Kom
NIP. 197808172003121005

Lampiran G : Biodata Penulis**Renaldi Habib Kholik**

D-III Teknik Komputer

Jurusan Teknologi Informasi,

Politeknik Negeri Jember



+6283852238885

Email : rhkrendh@gmail.com

IDENTITAS DIRI

Nama : Renaldi Habib Kholik
 Jenis Kelamin : Laki-laki
 Tempat / Tgl Lahir : Jember, 28 Agustus 1996
 Agama : Islam
 Kewarganegaraan : Indonesia
 Alamat : Dusun Onjur RT 01 RW 01 Desa Sempolan
 Kecamatan Silo Kabupaten Jember

LATAR BELAKANG PENDIDIKAN**Pendidikan Formal :**

2000 - 2002 : TK RA Perwanida 04 Sempolan, Jember
 2002 - 2008 : SD Negeri 03 Sempolan, Jember
 2008 - 2011 : SMP Negeri 1 Silo, Jember
 2011 - 2014 : SMA Nurul Islam, Jember
 2014 - sekarang : D-III Teknik Komputer, Jurusan Teknologi Informasi,
 Politeknik Negeri Jember

PENGALAMAN KERJA

-