

IMPLEMENTASI SISTEM MONITORING LUAPAN AIR PADA SELOKAN MENGGUNAKAN METODE FUZZY

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aulia Tri Wulandari Anwar

NIM: 125150300111048



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2017

PENGESAHAN

IMPLEMENTASI SISTEM MONITORING LUAPAN AIR PADA SELOKAN
MENGUNAKAN METODE *FUZZY*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Aulia Tri Wulandari Anwar

NIM: 125150300111048

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Februari 2017

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Mochammad Hannats H. I., S.ST, M.T

NIK: 201405 881229 1 001

Rekyan Regasari Mardi Putri, S.T, M.T

NIK: 201102 770414 2 001

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Februari 2017



Aulia Tri Wulandari Anwar

NIM: 125150300111048

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan juga hidayah-Nya, sehingga laporan skripsi yang berjudul “Implementasi Sistem Monitoring Luapan Air Pada Selokan Menggunakan Metode *Fuzzy*” ini dapat diselesaikan.

Penulis menyadari bahwa skripsi ini tidak terlepas dari bantuan beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih yang sebesar-besarnya kepada:

1. Ibunda Triyani, Ayahanda Kusnadi Anwar, M. Alfian Rosidi Anwar, Aditya Rosyid Dwi Anwar, dan seluruh keluarga besar atas doa, nasehat, kasih sayang, perhatian, dan segala bentuk dukungannya.
2. Bapak Mochammad Hannats Hanafi I., S.ST, M.T selaku dosen pembimbing pertama yang dengan sabar selalu memberikan dukungan dan arahan dalam mengerjakan skripsi ini.
3. Ibu Rekyan Regasari Mardi Putri, S.T, M.T selaku dosen pembimbing kedua yang dengan sabar selalu membantu dalam menyelesaikan laporan skripsi ini.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer.
5. Teman-teman kelinci reborn tercinta yaitu Tri Oktavia Mayasari, Alif Rahmawan Putra, Andrika Wahyu Wicaksono, Apry Nur Sudyanto, Bella Aulia Rahmataufani, Dwi Arif Afrianto, Ekky Deva Rizkyan, Faizal Ramadhan, Fajar Indra Yunanto, Rachmat Eko Prasetyo, Gatut Prasajo, Helmi Nizar, M Wildan Alauddin, Muhammad Wildan, Muhlis Agung Saputro, Panji Putera, Rizky Putra Pratama dan Thommas Oddy Chrisdwianto atas semangat, dan kesediannya meluangkan waktu dan tenaga selama pengerjaan skripsi.
6. Aneke Deborah atas doa, dukungan, dan kesediannya meluangkan waktu dan tenaga untuk menjadi teman diskusi selama pengerjaan skripsi.
7. Teman-teman program studi teknik komputer, khususnya angkatan 2012 yang selalu memberikan semangat dan dukungan dalam penyelesaian pengerjaan skripsi.
8. Semua orang yang selalu mendoakan dan memberi semangat dalam pengerjaan skripsi yang tidak dapat disebutkan penulis satu persatu, terimakasih atas segala bentuk dukungannya.

Penulis sadar akan banyaknya kekurangan yang ada pada laporan ini, sehingga penulis berharap adanya penyempurnaan oleh pihak-pihak terkait. Semoga laporan ini memberikan manfaat dan juga dapat menjadi referensi dalam penyempurnaan penelitian ini selanjutnya.

Malang, 2 Februari 2017

Penulis

aulia.3wulandari@gmail.com

ABSTRAK

Banjir merupakan peristiwa terbenamnya daratan karena volume air yang meningkat dan merupakan salah satu fenomena alam yang sering terjadi di berbagai wilayah yang sangat merusak dan merugikan masyarakat. Banjir juga bisa terjadi karena adanya luapan air berlebihan yang terjadi di suatu tempat karena adanya hujan besar yang salah satunya adalah luapan yang terjadi pada selokan dan menyebabkan air keluar dari batasan alaminya. Oleh karena itu dibangun sebuah sistem yang dapat monitoring selokan yang meluap dengan memanfaatkan sensor ultrasonik (PING) untuk mendeteksi ketinggian air dan sensor G1/2 untuk mendeteksi kecepatan air pada selokan. Data dari sensor kemudian diolah oleh mikrokontroler Arduino Nano dengan menggunakan logika *fuzzy*. Karena jarak yang relatif jauh antara alat dan lokasi monitoring maka hasil olahan data akan dikirimkan secara *wireless* oleh modul *NRF24L01*, kemudian *LED* pada lokasi monitoring akan menyala sesuai data yang diterima yaitu merah untuk bahaya, kuning untuk waspada, dan hijau untuk normal. Sehingga dapat segera di cek apakah air dalam selokan tersebut tersumbat agar dapat segera dibersihkan atau memang meluap dengan arus air yang deras agar dapat segera ditanggulangi. Dalam penelitian ini didapatkan bahwa alat bekerja dengan baik, kesesuaian fungsional logika *fuzzy* adalah 100% dan rata-rata waktu pemrosesan adalah 1001,33 ms.

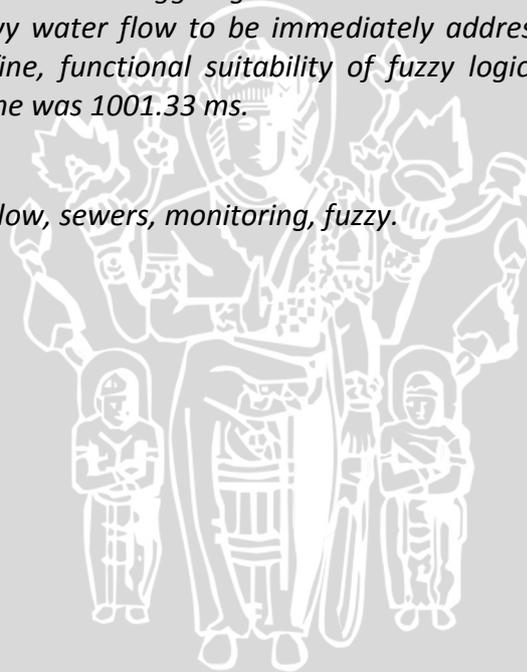
Kata kunci: banjir, luapan, selokan, monitoring, *fuzzy*.



ABSTRACT

Flood is the setting of the event mainland because of increased water volume and is one natural phenomenon that often occurs in various regions is extremely destructive and harmful to society. Flood can also occur due to excessive overflow that occurs in a place for their big rain, one of which is the overflow in sewers and causing the water out of their natural limits. Therefore built a system for monitoring sewer overflows by using ultrasonik sensors (PING) to detect water level and sensor G1/2 to detect the speed of the water in the ditch. Data from the sensors is then processed by a microcontroller Arduino Nano by using fuzzy logic. Karana distant relative distance between the tool and monitoring the location of the processed data will be transmitted wirelessly by NRF24L01 module, then the LED on the location of the monitoring will be lit according to the data received, namely red for danger, yellow for the alert, and green for normal. So it can be checked whether the water in clogged gutters in order to be cleaned or indeed overflowing with heavy water flow to be immediately addressed. It was found that the tools work fine, functional suitability of fuzzy logic is 100% and the average processing time was 1001.33 ms.

Keywords: flood, overflow, sewers, monitoring, fuzzy.

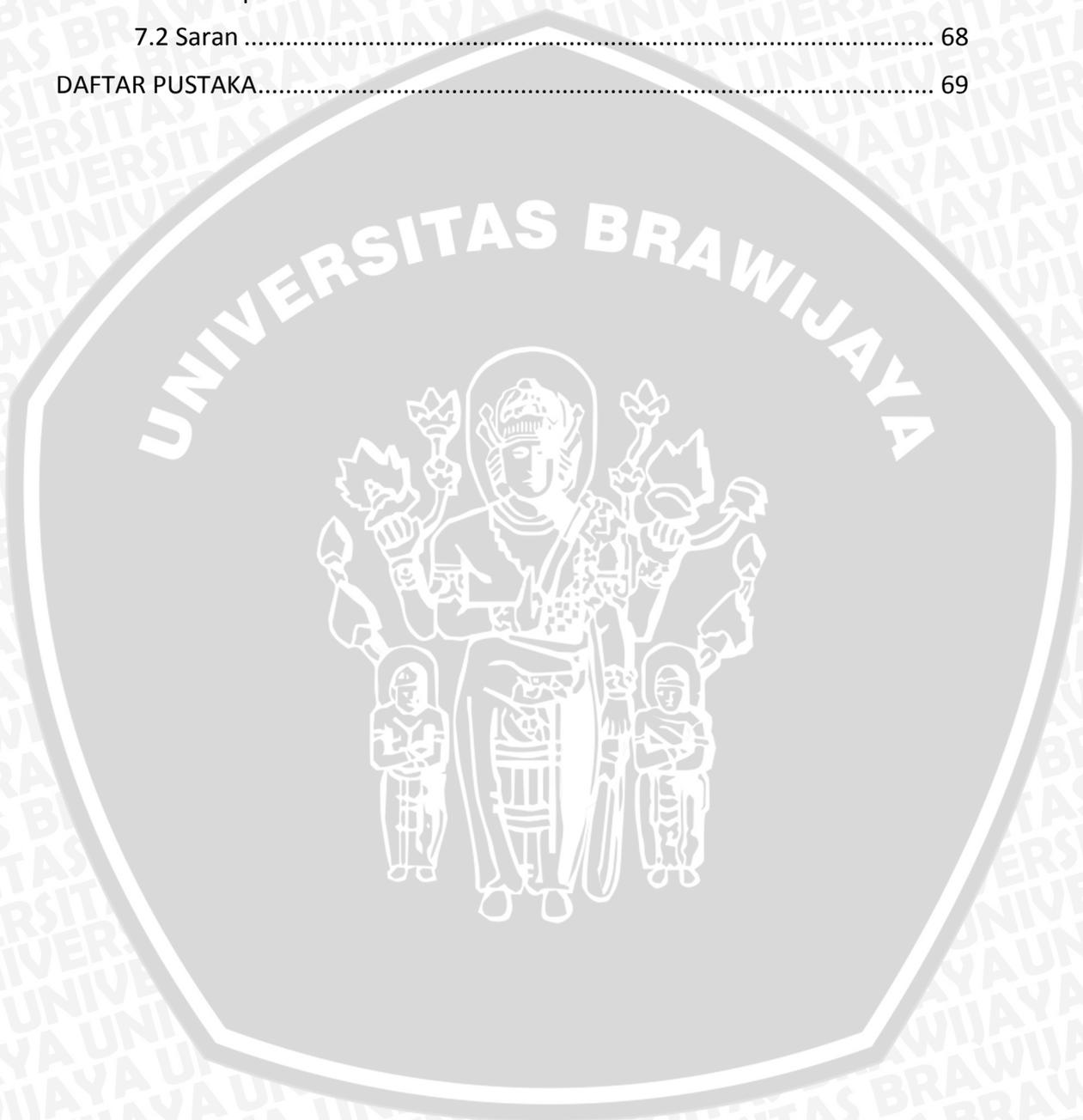


DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah.....	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	5
2.2.1 Banjir	6
2.2.2 Selokan	6
2.2.3 Arduino Nano.....	7
2.2.4 Sensor.....	10
2.2.5 Modul <i>Wireless NRF24L01</i>	13
2.2.6 <i>LED</i>	14
2.2.7 Logika <i>Fuzzy</i>	14
2.2.8 <i>Wireless Sensor Network</i>	20
BAB 3 METODOLOGI	21
3.1 Studi Literatur	22
3.2 Analisis Kebutuhan	22
3.3 Perancangan Sistem.....	23

3.4 Implementasi Sistem	24
3.5 Pengujian dan Analisis	24
3.6 Penarikan Kesimpulan	25
BAB 4 REKAYASA KEBUTUHAN.....	26
4.1 Deskripsi Umum.....	26
4.1.1 Perspektif Sistem.....	26
4.1.2 Karakteristik Pengguna	26
4.1.3 Lingkungan Sistem Operasi	26
4.1.4 Asumsi dan Ketergantungan	26
4.2 Rekayasa Kebutuhan.....	27
4.2.1 Kebutuhan Perangkat Keras.....	27
4.2.2 Kebutuhan Perangkat Lunak.....	28
4.2.3 Kebutuhan Fungsional.....	28
BAB 5 PERANCANGAN DAN IMPLEMENTASI	30
5.1 Perancangan Sistem.....	30
5.1.1 Perancangan Perangkat keras.....	30
5.1.2 Perancangan Perangkat Lunak.....	37
5.1.3 Perancangan Logika <i>Fuzzy</i>	39
5.1.4 Perancangan <i>Fuzzy Sugeno</i>	40
5.2 Implementasi Sistem	47
5.2.1 Implementasi <i>Prototype</i> Selokan	47
5.2.2 Implementasi Perangkat Keras Pada <i>Node</i> Sensor	47
5.2.3 Implementasi Perangkat Keras Pada <i>Node</i> Pos Jaga	48
5.2.4 Implementasi Perangkat Lunak Pada <i>Node</i> Sensor	48
5.2.5 Implementasi Perangkat Lunak Pada <i>Node</i> Pos Jaga.....	54
BAB 6 PENGUJIAN DAN ANALISIS.....	56
6.1 Pengujian dan Analisis	56
6.1.1 Pengujian Sensor Ultrasonik (PING).....	56
6.1.2 Pengujian Sensor Kecepatan Air	57
6.1.3 Pengujian Arduino Nano	58
6.1.4 Pengujian NRF24L01	58
6.1.5 Pengujian <i>LED</i>	59

6.1.6 Pengujian Fungsional Logika <i>Fuzzy</i>	60
6.1.7 Pengujian Waktu Pemrosesan	65
BAB 7 PENUTUPAN.....	68
7.1 Kesimpulan.....	68
7.2 Saran	68
DAFTAR PUSTAKA.....	69



DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Nano.....	8
Tabel 2.2 Spesifikasi Sensor Ultrasonik (PING)	12
Tabel 2.3 Spesifikasi Sensor Kecepatan Air.....	13
Tabel 2.4 Spesifikasi Modul <i>Wireless</i> NRF24L01.....	14
Tabel 5.1 Keterangan Pin Sensor Ultrasonik (PING)	32
Tabel 5.2 Keterangan Pin Sensor Kecepatan Air G1/2.....	33
Tabel 5.3 Keterangan Pin Modul NRF24L01 <i>Node</i> Sensor	33
Tabel 5.4 Keterangan Pin Modul NRF24L01 <i>Node</i> Pos Jaga	35
Tabel 5.5 Keterangan <i>LED</i>	36
Tabel 5.6 Keterangan <i>Rule Fuzzy</i>	45
Tabel 6.1 Hasil Pengujian Sensor Ultrasonik.....	56
Tabel 6.2 Hasil Pengujian Sensor Kecepatan Air.....	57
Tabel 6.3 Hasil Pengujian Arduino Nano.....	58
Tabel 6.4 Hasil Pengujian Fungsional Logika <i>Fuzzy</i>	62
Tabel 6.5 Hasil Pengujian Waktu Pemrosesan.....	67



DAFTAR GAMBAR

Gambar 2.1 Contoh selokan.....	7
Gambar 2.2 Arduino Nano Bagian Depan.....	7
Gambar 2.3 Arduino Nano Bagian Belakang.....	8
Gambar 2.4 Pin <i>Mapping</i> Arduino Nano.....	9
Gambar 2.5 Logo Arduino IDE.....	10
Gambar 2.6 Sensor Ultrasonik (PING).....	11
Gambar 2.7 Pancaran Gelombang Ultrasonik.....	11
Gambar 2.8 Sensor Kecepatan Air G1/2 1	12
Gambar 2.9 Modul NRF24L01	13
Gambar 2.10 <i>LED</i>	14
Gambar 2.11 Representasi linier naik.....	16
Gambar 2.12 Representasi linier turun.....	16
Gambar 2.13 Representasi Kurva Segitiga	17
Gambar 2.14 Representasi Kurva Trapesium	18
Gambar 3.1 Diagram Alir Metode Penelitian.....	21
Gambar 3.2 Diagram Blok Sistem	23
Gambar 4.1 Ilustrasi Kebutuhan Sistem.....	27
Gambar 4.2 Kebutuhan Perangkat Keras.....	27
Gambar 4.3 Kebutuhan Perangkat Lunak	28
Gambar 5.1 Diagram Alir Perancangan Perangkat Keras.....	30
Gambar 5.2 Rancangan <i>Prototype</i> Selokan.....	31
Gambar 5.3 Rancangan Sensor ultrasonik (PING) Ke Mikrokontroler.....	32
Gambar 5.4 Rancangan Kecepatan Air G1/2 Ke Mikrokontroler.....	32
Gambar 5.5 Rancangan Modul NRF24L01 Ke Mikrokontroler	33
Gambar 5.6 Rancangan <i>Node</i> Sensor.....	34
Gambar 5.7 Rancangan Mikrokontroler Dengan Modul NRF24L01	35
Gambar 5.8 Rancangan Mikrokontroler Dengan <i>LED</i>	36
Gambar 5.9 Rancangan <i>Node</i> Pos Jaga	37
Gambar 5.10 Diagram Alir Perancangan Perangkat Lunak Pada <i>Node</i> Sensor	38
Gambar 5.11 Diagram Alir Perancangan Perangkat Lunak Pada <i>Node</i> Pos Jaga..	39

Gambar 5.12 Diagram Alir Logika <i>Fuzzy</i>	40
Gambar 5.13 Diagram Alir Proses <i>Fuzzifikasi</i>	41
Gambar 5.14 Fungsi Keanggotaan Kecepatan Air.....	42
Gambar 5.15 Fungsi Keanggotaan Ketinggian Air.....	43
Gambar 5.16 Diagram Alir Proses Inferensi.....	44
Gambar 5.17 Diagram Alir Proses Defuzzifikasi.....	46
Gambar 5.18 <i>Prototype</i> Selokan.....	47
Gambar 5.19 <i>Node</i> Sensor.....	47
Gambar 5.20 <i>Node</i> Pos Jaga.....	48
Gambar 5.21 Kode Program Baca Sensor Ultrasonik (PING).....	48
Gambar 5.22 Kode Pemrograman Baca Data Sensor Kecepatan Air G1/2.....	49
Gambar 5.23 Kode Pemrograman Derajat Keanggotaan Ketinggian Air.....	50
Gambar 5.24 Kode Pemrograman Derajat Keanggotaan Kecepatan Air.....	51
Gambar 5.25 Kode Pemrograman Inferensi.....	52
Gambar 5.26 Kode Pemrograman Inferensi.....	52
Gambar 5.27 Kode Pemrograman Defuzzifikasi.....	53
Gambar 5.28 Kode Pemrograman NRF24L01 Pada <i>Node</i> Sensor.....	54
Gambar 5.29 Kode Pemrograman NRF24L01 Pada <i>Node</i> Pos Jaga.....	54
Gambar 5.30 Kode Pemrograman Pada <i>LED</i>	55
Gambar 6.1 Hasil Pengujian NRF24L01.....	59
Gambar 6.2 Hasil Pengujian <i>LED</i>	59
Gambar 6.4 Diagram Defuzzifikasi.....	62
Gambar 6.5 Program Menghitung Waktu Pemrosesan.....	66
Gambar 6.6 Waktu Pemrosesan.....	66



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia adalah negara beriklim tropis yang hanya memiliki dua musim yaitu musim panas dan musim hujan dalam satu tahunnya. Keadaan lingkungan di Indonesia sering memprihatinkan. Ketika musim hujan datang dalam kurun waktu kurang lebih enam bulan dalam setahun, yaitu pada bulan oktober hingga bulan maret, sering kali terjadi berbagai macam bencana alam seperti terjadinya tanah longsor dan juga banjir.

Banjir merupakan peristiwa terbenamnya daratan karena volume air yang meningkat dan merupakan salah satu fenomena alam yang sering terjadi di berbagai wilayah yang sangat merusak dan merugikan masyarakat (Li yan, 2011). Banjir tidak hanya terjadi di daerah aliran sungai saja, tetapi bisa juga terjadi di daerah atau wilayah yang jauh dari aliran sungai, misalnya di daerah padat penduduk dan jalan-jalan yang tidak memiliki drainase atau serapan yang baik. Di kota-kota besar, selain menimbulkan kemacetan dan kerusakan, banjir juga dapat mengganggu aktifitas perekonomian sehingga dapat menimbulkan kerugian yang sangat besar. Banjir juga bisa terjadi karena adanya luapan air berlebihan yang terjadi di suatu tempat akibat hujan besar, yang salah satunya adalah luapan pada selokan yang kemudian menyebabkan air keluar dari batasan alaminya. Luapan air yang terjadi pada selokan pada umumnya dapat diakibatkan oleh beberapa faktor, seperti curah hujan yang tinggi dengan jangka waktu yang sangat lama, sehingga tinggi air melebihi batas maksimal selokan, atau luapan yang terjadi akibat kurang lancarnya laju aliran air karena tersumbat oleh sampah. Sehingga dengan demikian akan ditunjukkan bahwa yang berhubungan dengan luapan pada selokan diantaranya kecepatan air dan ketinggian air pada selokan tersebut. Namun sejauh ini luapan pada selokan sering dipandang sebelah mata dan tidak pernah dianggap sebagai sebuah permasalahan yang besar. Padahal seperti yang kita ketahui, hal kecil seperti selokan dapat mengakibatkan permasalahan yang lebih besar seperti banjir.

Dengan adanya kemungkinan banjir yang disebabkan oleh luapan pada selokan tersebut, diharapkan adanya sebuah sistem yang dapat mengurangi atau bahkan mencegah terjadinya banjir di daerah-daerah langganan banjir akibat luapan selokan. Banyaknya saluran air dan cakupan wilayah yang luas sangat menyulitkan jika pengecekan dilakukan secara manual di setiap selokan dan dirasa kurang efektif dan efisien. Hal ini dapat menghabiskan waktu yang cukup lama untuk penanganan pada selokan yang meluap sehingga bencana banjir tidak dapat diminimalisasi atau bahkan ditanggulangi.

Oleh karena itu dibangun sebuah sistem yang dapat monitoring selokan yang meluap dengan memanfaatkan sensor ultrasonik (PING) untuk mendeteksi ketinggian air dan sensor G1/2 untuk mendeteksi kecepatan air dalam selokan. Data dari sensor kemudian diolah oleh mikrokontroler Arduino Nano dengan menggunakan logika *Fuzzy Sugeno*. Logika *Fuzzy Sugeno* merupakan suatu proses

pengambilan keputusan berbasis aturan yang bertujuan untuk memecahkan masalah, dimana sistem tersebut sulit untuk dimodelkan atau terdapat ambiguitas dan ketidakjelasan yang berlimpah dengan *output* berupa persamaan linier (Fahmi, 2011), seperti penelitian yang akan dilakukan, dengan kecepatan dan ketinggian yang diperoleh akan menghasilkan beberapa kemungkinan yang tidak pasti apakah itu termasuk normal, waspada atau bahaya. Dengan demikian logika *Fuzzy Sugeno* dapat mengefektifkan hasil olahan data yang diperoleh. Karena jarak yang relatif jauh antara alat dan lokasi monitoring, maka hasil olahan data akan dikirimkan secara *wireless* oleh modul NRF24L01 yang ada pada sistem ke modul *wireless* NRF24L01 yang ada pada lokasi monitoring, kemudian *LED* pada lokasi monitoring akan menyala sesuai data yang diterima yaitu merah untuk bahaya, kuning untuk waspada, dan hijau untuk normal. Sehingga dapat segera di cek apakah air dalam selokan tersebut tersumbat agar dapat segera dibersihkan atau memang meluap dengan arus air yang deras agar dapat segera ditanggulangi.

1.2 Rumusan masalah

Berdasarkan penjelasan dari latar belakang di atas, didapatkan rumusan masalah yang dapat dikaji yaitu seperti yang ada dibawah ini :

1. Bagaimana desain dan juga implementasi sistem dengan pemanfaatan metode *fuzzy* agar dapat mendeteksi luapan air pada selokan menggunakan sensor ketinggian dan kecepatan air?
2. Bagaimana validasi metode *Fuzzy Sugeno* yang diterapkan pada sistem monitoring luapan air pada selokan?
3. Bagaimana performa pemrosesan pada sistem monitoring luapan air pada selokan?

1.3 Tujuan

Adapun tujuan yang didapatkan berdasarkan rumusan masalah antara lain:

1. Desain dan implementasi sistem dengan pemanfaatan metode *fuzzy* agar dapat mendeteksi luapan air pada selokan menggunakan sensor ketinggian dan kecepatan air.
2. Mengetahui metode *Fuzzy Sugeno* yang diterapkan pada sistem monitoring luapan air pada selokan berdasarkan pengujian fungsional.
3. Mengetahui performa pengiriman secara *wireless* menggunakan NRF24L01 pada sistem monitoring luapan air pada selokan.

1.4 Manfaat

Manfaat dari dilakukannya penelitian ini adalah :

Bagi Penulis

1. Dapat lebih dalam lagi memahami bagaimana implementasi sistem pendeteksi dengan memanfaatkan metode *fuzzy* terutama dalam mendeteksi luapan air pada selokan.

2. Dapat menerapkan ilmu yang telah didapat terutama tentang sistem pendeteksi dengan pemanfaatan metode *fuzzy*.

Bagi Pengguna

1. Dapat mempermudah dalam mengetahui selokan didaerah pengguna yang meluap dan segera diatasi.
2. Mendapat peringatan yang kemudian dapat membatntu dalam mencegah terjadinya banjir sebelum benar-benar terjadi.
3. Dapat memonitoring keadaan selokan dari kejauhan.

1.5 Batasan masalah

Dalam penyampaian masalah masalah yang ada diatas maka di dapat beberapa batasan dalam melakukan penelitian seperti yang ada dibawah ini :

1. Dalam penelitian ini faktor yang dipertimbangkan hanya ketinggian air dan keceptan air.
2. Menggunakan logika *fuzzy* hanya sampai menampilkan klasifikasi kondisi tanpa mempertimbangkan keakurasiannya.
3. Jarak antara *node* sensor dan *node* pos jaga sesuai dengan jangkauan NRF24L01 yaitu 230m.

1.6 Sistematika pembahasan

Sistematika dalam penulisan penelitian ditunjukkan agar dapat memberikan gambaran dan juga uraian dari penyusunan tugas akhir secara garis besar yang meliputi beberapa bab, seperti yang ada dibawah ini.

BAB 1 : PENDAHULUAN

Pada bab ini menjelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB 2 : KAJIAN PUSTAKA DAN DASAR TEORI

Pada bab ini menguraikan kajian pustaka dan juga dasar teori yang mendasari sistem monitoring luapan air pada selokan dan beberapa aspek yang dapat digunakan pada penelitian ini.

BAB 3 : METODOLOGI

Pada bab ini menguraikan tentang metode dan juga langkah kerja yang terdiri dari studi literatur, analisis kebutuhan, spesifikasi sistem, perancangan sistem, implementasi.

BAB 4 : REKAYASA KEBUTUHAN

Pada bab ini menguraikan seluruh kebutuhan yang bertujuan agar sistem monitoring luapan air pada selokan ini dapat bekerja sesuai dengan tujuan yang terdiri dari, kebutuhan komunikasi, kebutuhan fungsional sistem, dan kebutuhan lainnya yang diperlukan sistem ini.

BAB 5 : PERANCANGAN DAN IMPLEMENTASI

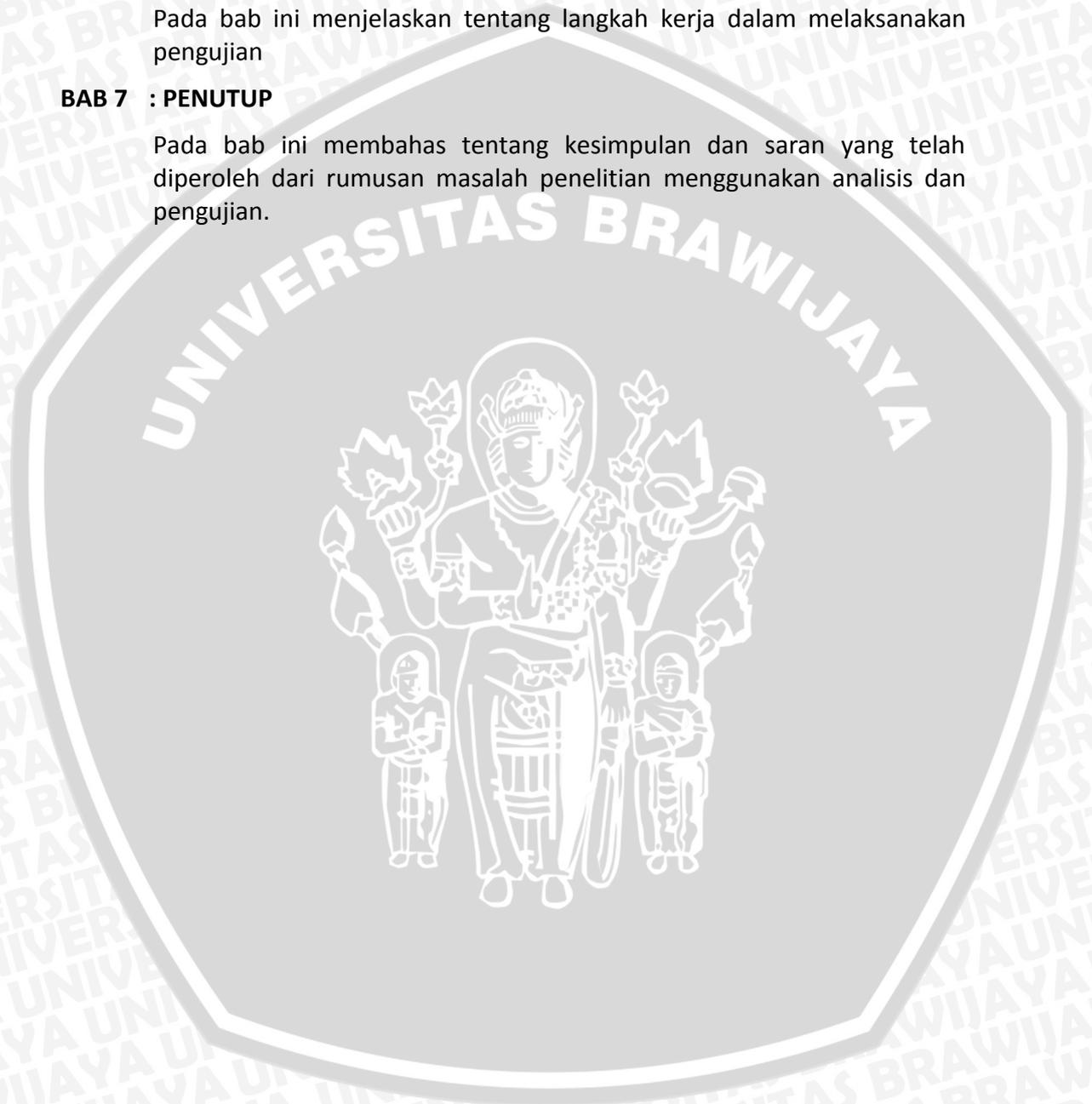
Pada bab ini menjelaskan perancangan sistem yang dibuat dan implementasi sistem yang dibuat yang berupa perangkat keras dan batasan implementasi.

BAB 6 : PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang langkah kerja dalam melaksanakan pengujian

BAB 7 : PENUTUP

Pada bab ini membahas tentang kesimpulan dan saran yang telah diperoleh dari rumusan masalah penelitian menggunakan analisis dan pengujian.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Tinjauan pustaka pada penelitian ini berisikan tentang penelitian yang telah dilakukan sebelumnya yang masih berhubungan dengan penelitian yang akan dilakukan. Tujuan diadakannya tinjauan pustaka yang telah ada sebelumnya agar bisa dijadikan referensi penentuan posisi peneliti dalam pelaksanaan penelitian.

Berikut adalah beberapa penelitian yang peneliti gunakan untuk menunjang pembuatan sistem yang akan di buat yaitu yang pertama adalah penelitian yang dilakukan oleh Amelia Syarfina Qisthi pada tahun 2012 yang berjudul "Perancangan Pendeteksi Banjir menggunakan Sistem Komunikasi 801.15.4". Penelitian ini membahas tentang sistem pendeteksi banjir yang dirancang menggunakan sensor ultrasonik sebagai *input*, mikrokontroler ATmega 8535 sebagai pengolah data, dimana *output* akan dikirimkan memanfaatkan komunikasi *wireless* Xbee-PRO dengan standar 802.15.4. Setelah itu *output* data yang berupa ketinggian akan ditampilkan di komputer menggunakan program *Visual Basic* 6.0. Penelitian ini berhasil dilakukan, namun parameter yang digunakan hanya ketinggian air saja.

Penelitian kedua dilakukan oleh P. Agustina, Agung Nugroho, Dan Unang Suryana pada tahun 2015 yang berjudul "Perancangan Dan Implementasi Perangkat Pendeteksi Banjir Dengan Sensor Pengukur Muka Level Air Menggunakan Logika Fuzzy". Penelitian ini membahas tentang sistem pendeteksi banjir yang dirancang dengan sensor kapasitif dan sensor *waterflow* yang memanfaatkan konsep kapasitor, yaitu menyimpan dan melepaskan muatan. Pada penelitian ini mikrokontroler yang digunakan adalah arduino uno dan menggunakan logika *Fuzzy Sugeno* dalam pengolahan datanya. *Output* pada penelitian ini berupa kondisi sungai dalam bentuk grafik. Penelitian kedua ini dinyatakan berhasil dengan tingkat akurasi sistem 99,65% dan tingkat akurasi sensor *waterflow* sebesar 96%. Sehingga dibutuhkan sensor *waterflow* yang mampu memberikan hasil pengukuran debit air yang lebih akurat.

Dari kajian-kajian tersebut maka ditawarkan sebuah alat monitoring luapan air pada selokan dengan sensor ultrasonik dan sensor kecepatan air yang diharapkan dapat mendeteksi luapan dan mencegah terjadinya banjir. Kelebihan dari alat monitoring luapan air pada selokan ini adalah metode *fuzzy* yang digunakan agar *output* yang diperoleh lebih efektif dan ditambahkan modul *wireless* agar dapat dilakukan monitoring dari jarak jauh.

2.2 Dasar Teori

Dasar teori dibutuhkan dalam membahas teori yang masih berkaitan dengan penelitian yang diusulkan, meliputi banjir, penyebab banjir, selokan, Arduino Nano, sensor ultrasonik (PING), sensor kecepatan air G1/2, modul *wireless* NRF24L01, LED, logika *fuzzy*, dan *wireless sensor network*.

2.2.1 Banjir

Banjir merupakan fenomena alam yang terjadi ketika adanya aliran air berlebihan dalam merendam daratan. Menurut pengarahannya Uni Eropa, banjir sebagai peredaman sementara oleh air pada daratan yang biasanya tidak terendam air, sedangkan berdasarkan SK SNI M-18-1989-F dalam Suparta 2004 menyatakan bahwa banjir merupakan aliran air yang relatif tinggi, dan sudah tidak tertampung oleh alur sungai atau saluran seperti selokan.

2.2.1.1 Penyebab Banjir

Dari macam-macam banjir yang ada terdapat beberapa penyebab (manulife-indonesia.com) seperti:

1. Luapan aliran sungai maupun selokan

Saluran air yang tidak lancar dapat mengakibatkan air tidak dapat mengalir dengan cukup baik dan akan meluap. Hal tersebut terjadi akibat pembuangan sampah sembarangan yang mengakibatkan luapan, sehingga apabila curah hujan yang tinggi dalam waktu yang cukup lama, seringkali mengakibatkan terjadinya banjir.

2. Curah hujan tinggi

Curah hujan tinggi dapat mengakibatkan penampungan air seperti sungai maupun selokan tidak dapat lagi menampung volume air yang telah melampaui kapasitas.

3. Pendirian rumah di sepanjang sungai

Pendirian rumah di sepanjang sungai dapat mengurangi lebar dari sungai. Sungai yang lebarnya berkurang dapat mengakibatkan sirkulasi air tidak dapat optimal.

4. Penggundulan hutan

Penggundulan hutan merupakan penebangan hutan dimana dalam prosesnya tidak memakai sistem tebang pilih. Sehingga tidak ada pohon yang digunakan untuk menyerap air sehingga air akan mengalir tanpa terkendali.

5. Sedikitnya daerah resapan air

Daerah resapan air membentuk suatu aliran air yang mengalir ke daerah yang lebih rendah yang apabila tidak ada akan mengakibatkan terjadinya penumpukan dan genangan besar yang mengakibatkan banjir.

2.2.2 Selokan

Selokan adalah bangunan atau saluran yang dipakai untuk membawa aliran air, baik untuk saluran yang digunakan dalam menyalurkan air pembuangan maupun air hujan yang nantinya akan dibawa ke suatu tempat sehingga tidak akan menjadi sebuah masalah bagi lingkungan dan juga kesehatan. Selokan juga dipergunakan untuk jembatan dalam ukuran kecil yang digunakan dalam mengalirkan sungai kecil maupun sebagai bagian dari drainase atau selokan

repository.ub.ac.id

jalan. Berikut adalah manfaat dan fungsi selokan bila di pergunakan dengan baik(www.greenpack.co.id):

1. Selokan berfungsi sebagai saluran pembuangan air yang sudah dipergunakan oleh manusia dari berbagai aktifitas dan kegiatan sehari-hari.
2. Selokan mampu mengantarkan air hujan yang turun menuju tempat yang lebih rendah ke penampungan air.
3. Menghindari jalan raya dari genangan air hujan.
4. Dapat mencegah banjir
5. Air serta sampah dari selokan dapat di pergunakan sebagai pupuk organik guna penyuburan tanaman.

Contoh selokan dapat dilihat pada Gambar 2.1 dibawah ini.



Gambar 2.1 Contoh selokan

Sumber: panoramio.com

2.2.3 Arduino Nano

Arduino Nano merupakan mikrokontroler yang berbasis pada ATmega328P atau ATmega168 yang diproduksi oleh perusahaan Gravitech. yang dirancang dan Arduino Nano juga memiliki fungsi yang mirip dengan Arduino Duemilanove, hanya saja jenis pakatnya yang berbeda. Arduino Nano dihubungkan ke komputer menggunakan port USB Mini-B (melalui catu daya eksternal dengan tegangan belum teregulasi antara 6-20 Volt) dan tidak menyertakan colokan DC berjenis *Barrel Jack* (www.arduino.cc). Dibawah ini adalah contoh Arduino Nano yang akan ditunjukkan pada Gambar 2.2 dan Gambar 2.3.



Gambar 2.2 Arduino Nano Bagian Depan

Sumber: www.arduino.cc



Gambar 2.3 Arduino Nano Bagian Belakang

Sumber: www.arduino.cc

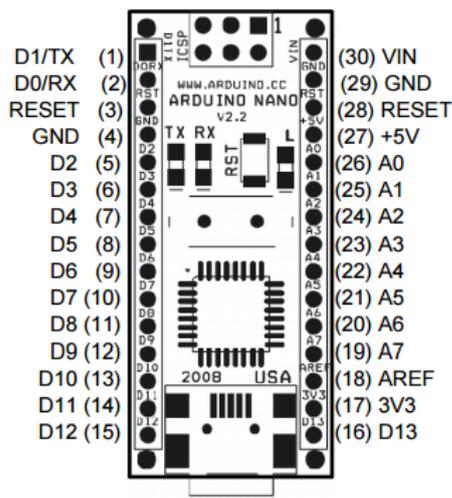
Spesifikasi pada Arduino Nano akan ditunjukkan pada Tabel 2.1

Tabel 2.1 Spesifikasi Arduino Nano

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage Limit	6-20V
Digital I/O Pins	14 termasuk PWM output
PWM Digital I/O Pins	6
Analog Input Pins	8
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16KB (ATmega168) atau 32 KB (ATmega328) 2 KB untuk bootloader
SRAM	1 KB (ATmega168) atau 2 KB (ATmega328)
EEPROM	512 byte (ATmega168) atau 1 KB (ATmega328)
Clock Speed	16 MHz
Length	1,85 cm
Width	4,3 cm

Sumber: www.arduino.cc

Arduino Nano dihubungkan menggunakan pin 30, pin VIN, maupun melalui catu daya eksternal dengan tegangan 5 volt melalui pin 27 yang merupakan pin 5V. Sumber daya pada Arduino Nano akan dipilih otomatis dari sumber tegangan yang paling tinggi. *Chip* FTDI FT232L di Arduino Nano akan aktif bila memperoleh daya dari USB, sehingga pin 3.3V dapat mengeluarkan tegangan, sedangkan LED pada TX dan RX akan berkedip bila pin digital 0 dan 1 berada dalam posisi *high*.



Gambar 2.4 Pin Mapping Arduino Nano

Sumber: www.arduino.cc

Penjelasan pada pin *input output* adalah seperti yang ada dibawah ini :

Pada Arduino Nano terdapat 14 pin digital yang masing dapat digunakan sebagai *input* maupun *output* dengan menggunakan fungsi *pinMode()*, *digitalWrite()*, dan juga *digitalRead()*. Dimana semua pin dapat beroperasi pada tegangan sebesar 5 volt. Setiap pin memberi maupun menerima arus maksimal 40 mA dan mempunyai resistor *pull-up internal* yang terputus akan secara default sebesar 20-50 KOhm. Selain pin diatas ada beberapa pin yang memiliki fungsi khusus, yaitu:

1. Serial

Terdiri dari 0 (RX) dan 1 (TX). Diamana pin untuk menerima adalah (RX) dan mengirimkan pin untuk mengirim data adalah (TX) TTL yang merupakan data serial. Pin tersebut terhubung menuju pin yang sesuai dengan *chip* FTDI USB-to-TTL Serial.

2. Interupsi *External*

Terdiri dari pin 2 dan pin 3 yang bisa dikonfigurasi untuk memicu interupsi pada nilai yang rendah, meningkat, menurun, atau pada perubahan nilai.

3. PWM

Terdiri dari pin 3, 5, 6, 9, 10, dan 11 yang memiliki *output* PWM 8-bit menggunakan fungsi *analogWrite()*.

4. SPI

Terdiri dari pin 10 (SS), 11 (MOSI), 12 (MISO), dan 13 (SCK) yang mendukung komunikasi SPI yang tersedia pada *hardware*.

5. LED

Merupakan pin 13 yang tersedia secara *built-in* di papan Arduino Nano dan terhubung ke pin digital 13. Sehingga jika pin tersebut diset dengan nilai *high*, LED akan menyala, dan jika pin tersebut diset dengan nilai *LOW*, maka LED akan mati.

Arduino Nano juga memiliki 8 pin yang digunakan sebagai *input* analog yang diberi label A0 sampai dengan A7. Masing-masing Pin menyediakan resolusi 10



bit atau 1024 nilai yang berbeda. Pin ini secara *default* dapat diukur atau diatur mulai dari *ground* sampai 5 Volt dan juga memungkinkan jika ingin mengganti titik jangkauan tertinggi maupun terendah menggunakan fungsi *analogReference()*.

2.2.3.1 Software Arduino Nano

Pemrograman Arduino Nano dapat dilakukan dengan menggunakan perangkat lunak Arduino IDE (*Integrated Development Environment*) yang merupakan sebuah *open source* agar dapat mempermudah dalam mengembangkan mikrokontroler dari membuat kode program, kompilasi, mengunggah, uji coba secara terminal serial ke semua *board* arduino termasuk Arduino Nano. Arduino IDE dapat berjalan di berbagai macam *platform* komputer seperti pada Windows, Mac OS X, dan Linux. Kode program aplikasi mikrokontroler dapat menggunakan bahasa C atau C++ yang dapat digabungkan dengan bahasa *assembly* (www.arduino.cc). Dibawah ini merupakan logo dari Arduino IDE yang dapat dilihat pada Gambar 2.5.



Gambar 2.5 Logo Arduino IDE

Sumber: www.arduino.cc

2.2.4 Sensor

Sensor merupakan perangkat yang digunakan dalam mengubah besaran fisis menjadi besaran elektrik. Sensor yang digunakan dalam implementasi sistem yang monitoring luapan air pada selokan adalah sensor ultrasonik (PING) sebagai pengukur ketinggian air pada selokan dan sensor kecepatan air G1/2 sebagai pengukur kecepatan air pada selokan.

2.2.4.1 Sensor Ultrasonik (PING)

Modul Sensor Ultrasonik (PING) dapat mengukur dengan jarak antara 2 cm hingga 300 cm. *Output* dari sensor ultrasonik (PING) adalah pulsa yang merepresentasikan jarak sesuai lebarnya. Lebar pulsa yang dihasilkan oleh sensor ultrasonik ini beragam dari 115 uS hingga 18,5 mS. Modul sensor ultrasonik memiliki *chip* pembangkit sinyal sebesar 40KHz, speaker ultrasonik yang digunakan untuk mengubah sinyal sebesar 40 KHz menjadi suara sementara, dan mikrofon ultrasonik yang digunakan untuk mendeteksi

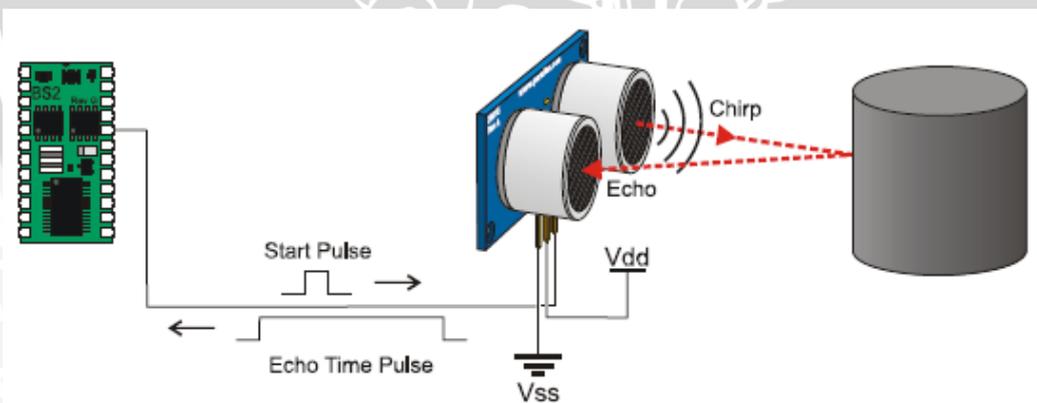
pantulan suara tersebut. Sinyal *output* pada modul sensor ultrasonik langsung dihubungkan ke mikrokontroler tanpa adanya tambahan komponen apapun. Modul sensor ultrasonik (PING) akan ditunjukkan pada Gambar 2.6.



Gambar 2.6 Sensor Ultrasonik (PING)

Sumber: www.parallax.com

Sensor Ultrasonik (PING) terdiri dari unit pemancar atau yang biasa disebut *transmitter* yang akan memancarkan gelombang ultrasonik dan unit penerima yaitu *receiver*. Suara ultrasonik sebesar 40KHz akan dipancarkan selama 200 μ S yang akan merambat di udara dengan kecepatan 344.424m/detik (1cm setiap 29.034 μ S). Pancaran tersebut akan mengenai objek yang kemudian dipantulkan kembali menuju modul sensor ultrasonik. Ketika menunggu pantulan sinyal ultrasonik yang dipancarkan oleh *transmitter*, modul sensor ultrasonik akan menghasilkan pulsa yang akan berhenti (*low*) jika suara pantulan telah terdeteksi modul sensor ultrasonik. Maka dari itulah lebar pulsa tersebut dikatakan dapat merepresentasikan jarak di antara modul Sensor Ultrasonik dan objek (www.digi-bytes.com). Gambar 2.7 merupakan pancaran gelombang ultrasonik.



Gambar 2.7 Pancaran Gelombang Ultrasonik

Sumber: www.parallax.com

Spesifikasi pada Sensor Ultrasonik (PING) akan ditunjukkan pada table 2.2

Tabel 2.2 Spesifikasi Sensor Ultrasonik (PING)

<i>Supply voltage</i>	+5 VDC
<i>Supply current</i>	30 mA typ; 35 mA max
<i>Communication</i>	<i>Positive TTL pulse</i>
<i>Package</i>	3-pin SIP, 0.1" spacing (ground, power, signal)
<i>Size</i>	22 mm H x 46 mm W x 16 mm D(0.84 in x 1.8 in x 0.6 in)
<i>Weight</i>	9 g (0.32 oz)

Sumber: www.parallax.com

2.2.4.2 Sensor Kecepatan Air G1/2

Sensor kecepatan air G1/2 tersusun dari tubuh yang berupa katup plastik, rotor air, dan juga sensor *hall effect*. Ketika air mengalir melintasi rotor, rotor akan otomatis berputar. Kecepatan putaran tersebut akan bergantung terhadap kecepatan aliran air. Sensor *hall effect* mengeluarkan *output* berupa pulsa berdasarkan besarnya aliran air. Kelebihan sensor kecepatan air ini adalah sensor hanya memerlukan 1 sinyal (*SIG*) selain 5V DC dan *ground*. Sensor kecepatan air akan ditunjukkan pada gambar 2.8.



Gambar 2.8 Sensor Kecepatan Air G1/2 1

Sumber: www.seeedstudio.com

Prinsip kerja dari sensor kecepatan air ini memanfaatkan fenomena *hall effect*. *Hall effect* ini didasarkan pada efek medan magnetik pada partikel bermuatan yang bergerak. Ketika arus listrik mengalir pada *device hall effect* yang terletak di dalam medan magnet yang memiliki arah tegak lurus dengan arus listrik, pembawa muatan akan bergerak dan berbelok ke salah satu sisi yang akan menghasilkan medan listrik (www.geraicerdas.com). Medan listrik akan semakin membesar yang membuat gaya Lorentz pada partikel menjadi nol. Potensial *hall* adalah perbedaan potensial antara kedua sisi *device* yang sebanding dengan medan magnet dan juga arus listrik yang melintasi *device*. Dibawah ini Tabel 2.3 yang merupakan spesifikasi dari sensor kecepatan air:

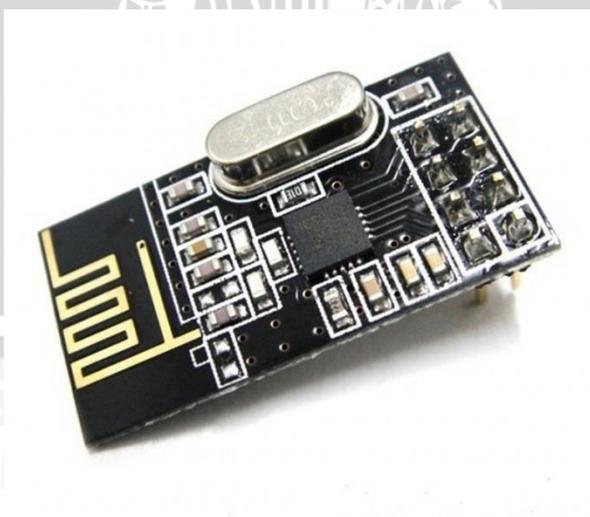
Tabel 2.3 Spesifikasi Sensor Kecepatan Air

<i>Working voltage</i>	5V-24V
<i>Maximum current</i>	15 mA (DC 5V)
<i>Weight</i>	43 g
<i>External diameters (Inflow and outflow)</i>	20mm
<i>Flow rate range</i>	1~30 L/min
<i>Operating temperature</i>	0°C~80°C
<i>Operating humidity</i>	35%~90%RH
<i>Operating pressure</i>	under 1.2Mpa
<i>Store temperature</i>	-25°C~+80°C

Sumber: www.seeeduno.com

2.2.5 Modul *Wireless NRF24L01*

Modul *wireless NRF24L01* adalah *chip* orisinil produksi *Nordic Semiconductor* dari Norwegia yang merupakan sebuah modul komunikasi nirkabel dengan jarak yang jauh menggunakan pita gelombang RF 2.4GHz ISM(*Industrial, Scientific and Medical*) yang beroperasi pada pita ISM 2,4 GHz dan juga menggunakan antarmuka SPI dalam berkomunikasi (www.geraicerdas.com). Modul NRF24L01 ini mempunyai 8 pin, yaitu VCC (3.3V DC), GND, CE, CSN, MISO, MOSI, SCK, dan IRQ. Rentang data modul NRF24L01 yang mencapai 2Mbps dengan *ultra low power* dimana penanganan paket data dan transaksi paketnya dilakukan secara otomatis. Bentuk modul NRF24L01 akan ditunjukkan pada Gambar 2.9.



Gambar 2.9 Modul NRF24L01

Sumber : www.digiwarestore.com (2015)

Dibawah ini tabel yang menunjukkan spesifikasi dari *Modul Wireless NRF24L01*.

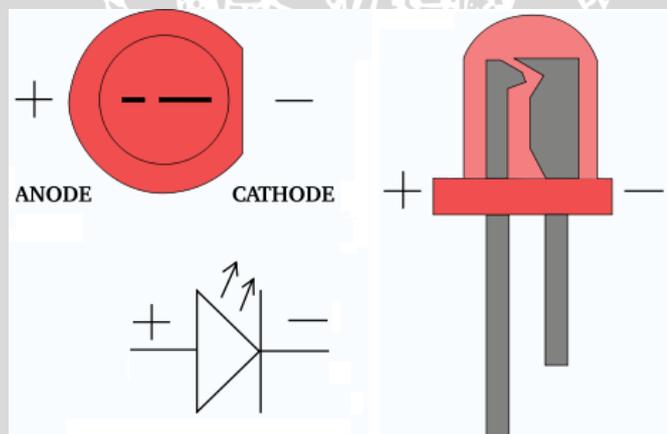
Tabel 2.4 Spesifikasi Modul *Wireless NRF24L01*

<i>Power supply</i>	1.9~3.6V
<i>IO port working voltage</i>	0~3.3 volt
<i>Transmitting rate</i>	+7dB
<i>Receiving sensitivity</i>	90dB
<i>Transmission range</i>	250m in open area
<i>Dimension</i>	15x29mm

Sumber : www.geraicerdas.com (2014)

2.2.6 LED

LED yang merupakan singkatan dari *light emitting dioda* merupakan dioda yang bisa memancarkan cahaya ketika mendapatkan arus bias maju (*forward bias*). Tidak seperti dioda pada umumnya, kemampuan *LED* dalam mengalirkan arus cukup rendah yaitu dengan nilai maksimal 20 mA, sehingga bila *LED* dialiri arus yang lebih besar dari 20 mA dapat mengakibatkan kerusakan pada *LED*, sehingga dalam rangkaian *LED* dipasang resistor untuk pembatas arus (teknikelektronika.com). Simbol dan bentuk fisik dari *LED* akan ditunjukkan pada Gambar 2.10.



Gambar 2.10 LED

Sumber: www.buildcircuit.com

2.2.7 Logika Fuzzy

Fuzzy dikenalkan oleh Dr. Lotfi Zadeh dari Universitas California, Berkeley pada 1965 yang merupakan titik awal dari konsep modern mengenai ketidakpastian. Dr. Lotfi Zadeh mengenalkan teori yang mempunyai obyek-obyek dari himpunan *fuzzy* yang memiliki batasan tidak presisi dari keanggotaan pada himpunan *fuzzy*, bukan dalam bentuk logika benar (*true*) atau salah (*false*), akan tetapi dinyatakan dalam bentuk derajat (*degree*).

Logika *fuzzy* merupakan cara yang tepat dalam memetakan ruang *input* ke dalam ruang *output* yang termasuk peningkatan dari konsep kebenaran sebagian yaitu logika Boolean. Dimana dalam logika klasik segala hal dapat diekspresikan dalam istilah Binary (0 atau 1, hitam atau putih, ya atau tidak), sedangkan logika *fuzzy* menggantikan kebenaran dari boolean menggunakan tingkat kebenaran yang memungkinkan nilai keanggotaan diantara 0 dan 1, yang dalam bentuk linguistik, konsep yang tidak pasti seperti "sedikit", "lumayan", dan "sangat", sehingga sering diterapkan dalam sistem otomatis dan juga sistem pendukung suatu keputusan.

2.2.7.1 Alasan digunakan Logika Fuzzy (Kusumadewi, 2010)

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *Fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinier yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* didasarkan pada bahasa alami.

2.2.7.2 Himpunan Fuzzy

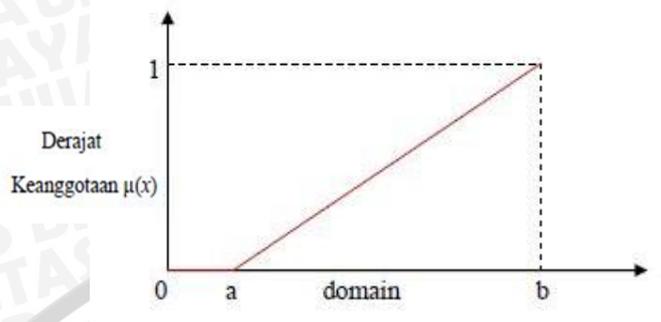
Himpunan merupakan suatu kumpulan dari objek-objek yang memiliki kesamaan sifat tertentu. Himpunan *fuzzy* adalah pengembangan lebih lanjut dari konsep himpunan dalam matematika. Himpunan *fuzzy* adalah rentang nilai yang masing-masing nilainya memiliki derajat keanggotaan diantara 0 sampai 1.

Himpunan *fuzzy* memiliki 2 atribut, (Kusumadewi, 2010) yaitu:

1. Linguistik yang merupakan penamaan grup yang mewakili suatu keadaan maupun kondisi tertentu dengan menggunakan bahasa yang alami, seperti: KERING, LEMBAP, BASAH.
2. Numeris yang merupakan nilai (dalam bentuk angka) yang mempresentasikan ukuran suatu variabel, seperti : 25, 50, 75 dsb.

Fungsi keanggotaan atau *membership function*, (Kusumadewi, 2010) adalah suatu kurva yang menampilkan pemetaan dari titik-titik *input* data ke nilai keanggotaannya atau derajat keanggotaannya yang memiliki interval antara 0 hingga 1. Ada beberapa fungsi yang bisa digunakan yaitu linier (linier naik dan linier turun), kurva segitiga, dan kurva trapesium. Fungsi tersebut akan dijelaskan seperti yang ada dibawah ini:

1. Linier naik, merupakan kenaikan himpunan yang dimulai dari nilai yang memiliki derajat keanggotaan nol ke arah kanan dan ke nilai yang memiliki derajat keanggotaan lebih tinggi yang dapat dilihat pada Gambar 2.11.



Gambar 2.11 Representasi linier naik

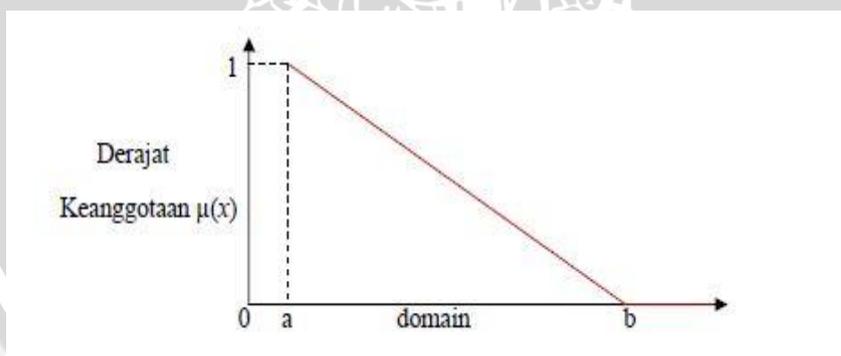
Sumber : Aplikasi Logika Fuzzy Sri Kusumadewi (2010)

Dalam menentukan derajat keanggotaan linier naik akan ditunjukkan pada persamaan (2.1) dibawah ini.

$$\mu(x) = \begin{cases} 0 & : x \leq a \\ \frac{(x-a)}{(b-a)} & : a < x \leq b \end{cases} \quad (2.1)$$

Pada persamaan (2.1) a merupakan nilai domain dengan derajat keanggotaan tendah yaitu 0, sedangkan b adalah nilai domain dengan derajat keanggotaan tertinggi yaitu 1, dan x adalah *input* yang dimasukkan.

2. Linier turunan, yang merupakan garis lurus diawali dari nilai domain dengan derajat keanggotaan tertinggi ke keanggotaan yang rendah akan ditunjukkan pada Gambar 2.12.



Gambar 2.12 Representasi linier turun

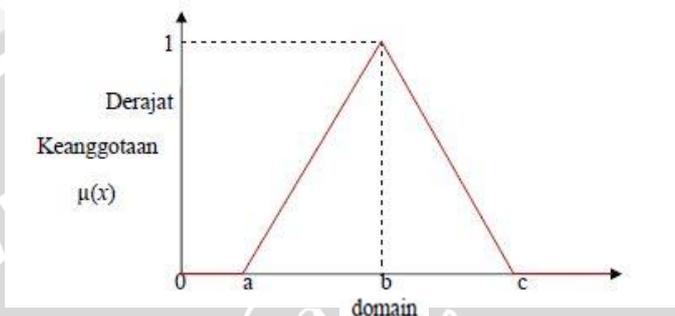
Sumber : Aplikasi Logika Fuzzy Sri Kusumadewi (2010)

Dalam menampilkan linier turun derajat keanggotaannya akan ditunjukkan pada persamaan (2.2) dibawah ini.

$$\mu(x) = \begin{cases} \frac{(x-a)}{(b-a)} & : a \leq x < b \\ 0 & : x \geq b \end{cases} \quad (2.2)$$

Pada persamaan (2.2) a merupakan nilai domain dengan derajat keanggotaan tertinggi yaitu 1, sedangkan b adalah nilai domain dengan derajat keanggotaan rendah yaitu 0, dan x adalah *input* yang dimasukkan.

3. Kurva segitiga, adalah perpaduan dari linier naik dan linier turun seperti yang dapat dilihat pada Gambar 2.13.



Gambar 2.13 Representasi Kurva Segitiga

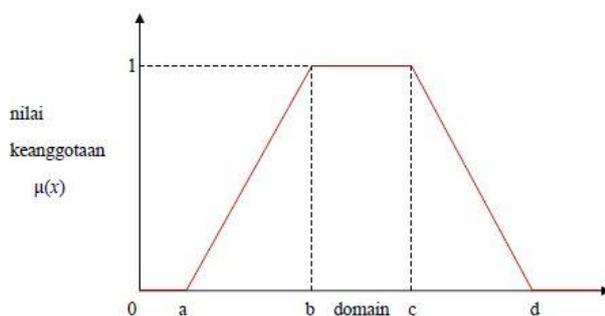
Sumber : Aplikasi Logika Fuzzy Sri Kusumadewi (2010)

Dalam menampilkan kurva segitiga, derajat keanggotaan akan ditunjukkan pada persamaan (2.3) dibawah ini.

$$\mu(x) = \begin{cases} 0 & : x \leq a \text{ dan } x \geq c \\ \frac{(x-a)}{(b-a)} & : a < x \leq b \\ \frac{(c-x)}{(c-b)} & : b < x < c \end{cases} \quad (2.3)$$

Pada persamaan (2.3) a merupakan nilai domain dengan derajat keanggotaan 0, sedangkan b adalah nilai domain dengan derajat keanggotaan tertinggi yaitu 1, c adalah domain setelah b dengan derajat keanggotaan 0, dan x adalah *input* yang dimasukkan.

4. Kurva trapesium berbentuk mirip kurva segitiga, hanya saja beberapa titik memiliki nilai keanggotaan 1 (satu) seperti yang dapat dilihat pada Gambar 2.14.



Gambar 2.14 Representasi Kurva Trapesium

Sumber : Aplikasi Logika Fuzzy Sri Kusumadewi (2010)

Dalam menampilkan kurva trapesium derajat keanggotaan akan ditunjukkan pada persamaan (2.4) dibawah ini.

$$\mu(x) = \begin{cases} 0 & : x \leq a \text{ atau } x \geq d \\ \frac{(x-a)}{(b-a)} & : a < x \leq b \\ \frac{(d-x)}{(d-c)} & : c < x < d \\ 1 & : b < x \leq c \end{cases} \quad (2.4)$$

Pada persamaan (2.3) a merupakan nilai domain dengan derajat keanggotaan 0, sedangkan b adalah nilai domain dengan derajat keanggotaan 1, c adalah nilai domain dengan derajat keanggotaan 1, dan d adalah domain dengan derajat keanggotaan 0, dan x adalah *input* yang dimasukkan.

2.2.7.3 Operasi Himpunan Fuzzy

Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu (Kusumadewi,2004):

1. Operator AND

Operator ini untuk hasil operasi dua himpunan (α predikat) diperoleh dengan mengambil nilai *minimum* antara kedua himpunan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y])$$

Misal nilai keanggotaan umur 27 pada himpunan muda adalah $\mu_{MUDA}[27] = 0,6$ dan nilai keanggotaan 2 juta pada himpunan penghasilan TINGGI adalah $\mu_{GAJITINGGI}[2\text{juta}] = 0,8$

maka α -predikat untuk usia MUDA dan berpenghasilan TINGGI adalah nilai keanggotaan minimum :

$$\begin{aligned} &\mu_{MUDA \cap GAJITINGGI} \\ &= \min(\mu_{MUDA}[27], \mu_{GAJITINGGI}[2\text{juta}]) \\ &= \min(0,6 ; 0,8) \\ &= 0,6 \end{aligned}$$

2. Operator *OR*

Operator ini berhubungan dengan operasi *union* pada himpunan, α predikat didapat dengan mengambil nilai *maximum* antara kedua himpunan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y])$$

Misal nilai keanggotaan umur 27 pada himpunan muda adalah $\mu_{MUDA}[27] = 0,6$ dan nilai keanggotaan 2 juta pada himpunan penghasilan TINGGI adalah $\mu_{GAJITINGGI}[2juta] = 0,8$

maka α -predikat untuk usia MUDA atau berpenghasilan TINGGI adalah nilai keanggotaan maksimum :

$$\begin{aligned} \mu_{MUDA \cup GAJITINGGI} &= \max(\mu_{MUDA}[27], \mu_{GAJITINGGI}[2juta]) \\ &= \max(0,6 ; 0,8) \\ &= 0,8 \end{aligned}$$

3. Operasi *NOT*

Operator ini berhubungan dengan operasi komplemen pada himpunan, α predikat diperoleh dengan mengurangi nilai keanggotaan elemen dengan nilai 1 (satu).

Misal nilai keanggotaan umur 27 pada himpunan muda adalah $\mu_{MUDA}[27] = 0,6$ maka α -predikat untuk usia TIDAK MUDA adalah :

$$\begin{aligned} \mu_{MUDA'}[27] &= 1 - \mu_{MUDA}[27] \\ &= 1 - 0,6 \\ &= 0,4 \end{aligned}$$

2.2.7.4 Metode *Fuzzy Sugeno*

Metode *Fuzzy Sugeno* dikenalkan Takagi-Sugeno Kang di tahun 1985. Penalaran metode Sugeno ini hampir sama dengan penalaran Mamdani, akan tetapi *output* sistem bukan himpunan *fuzzy*, melainkan konstanta atau persamaan linier. Terdapat 4 langkah untuk mendapatkan *output*, yaitu:

1. Fuzzifikasi

Merupakan penentuan himpunan *fuzzy* dimana variabel *input* dan *output* dibagi menjadi satu atau bahkan bisa lebih.

2. Fungsi Implikasi

Merupakan fungsi menyusun basis aturan, yaitu aturan-aturan berupa implikasi *fuzzy* yang menyatakan relasi antara variabel *input* dengan variabel *output*. Fungsi implikasi yang digunakan adalah Min (*Minimum*) yaitu solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai *minimum* aturan.

3. Komposisi Aturan

Apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan kolerasi antar aturan. Metode yang digunakan dalam melakukan inferensi sistem *fuzzy* adalah Metode Max (*Maximum*) yaitu solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan.

4. Defuzzifikasi

Input yang digunakan dalam proses penegasan adalah himpunan *fuzzy* yang didapatkan dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan adalah bilangan real yang tegas, sehingga jika diberi suatu himpunan *fuzzy* dengan *range* tertentu, maka harus dapat diambil nilai tegas tertentu yang akan menjadi *output*. Metode yang digunakan dalam skripsi ini adalah *height method* yang dikenal dengan prinsip keanggotaan maksimum dikarenakan metode ini dengan sederhana memilih nilai *crisp* yang memiliki derajat keanggotaan maksimum.

2.2.8 Wireless Sensor Network

Wireless Sensor Network (WSN) adalah kumpulan dari beberapa komponen yang saling terhubung satu sama lain sehingga mampu saling berkomunikasi maupun bertukar data, dan dapat membentuk suatu jaringan yang terhubung tanpa menggunakan kabel atau yang biasa disebut nirkabel (Hermawan, 2016). Pada *wireless sensor network* terdapat *node* yang dapat terhubung dengan *node* yang lain, sehingga dapat diolah atau dimonitoring oleh *user*. *Node* pada *wireless sensor network* mempunyai beberapa bagian, yaitu:

a. Sensing Unit

Merupakan komponen yang digunakan dalam mengambil data dari lingkungan sesuai dengan karakteristiknya. Jika data yang didapatkan merupakan data analog, maka data tersebut akan dirubah terlebih dahulu ke dalam data digital dengan menggunakan ADC (*Analog to Digital Converter*).

b. Processing unit

Merupakan komponen yang digunakan dalam pengolahan data yang diambil oleh *sensing unit*, dan kemudian meneruskannya ke *actuator* maupun ke *node* lain.

c. Transceiver

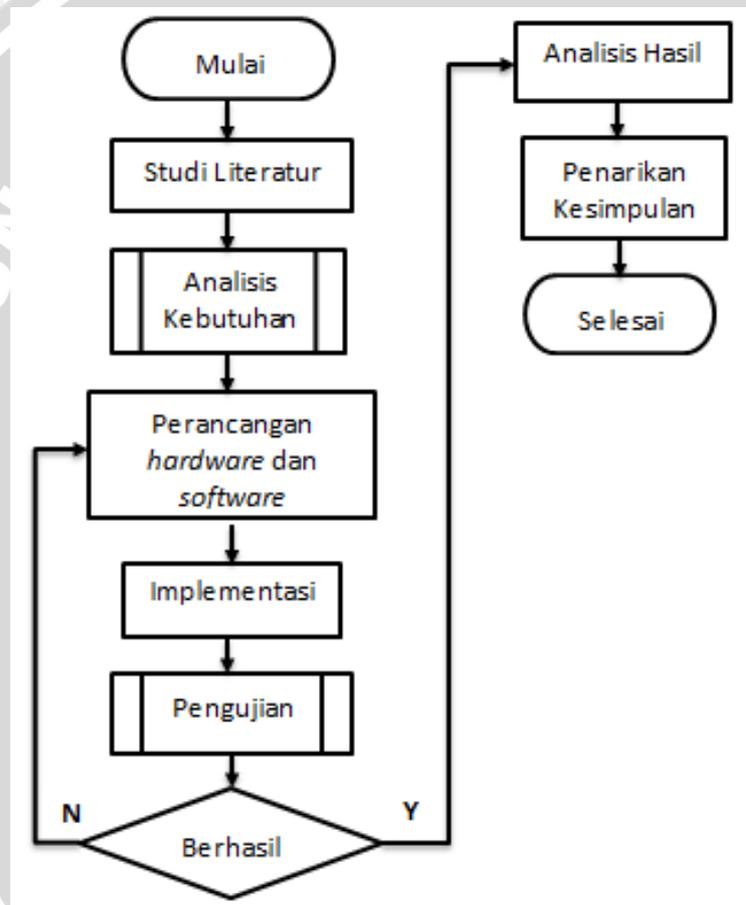
Komponen yang berfungsi untuk menerima data dari *node* lain, dan juga dapat mengirimkan data dari *processing unit* untuk dikirim ke *node* yang lain.

d. Power Unit

Komponen yang dapat mendistribusikan energi untuk digunakan oleh *sensing unit*, *processing unit*, dan juga oleh *transceiver*.

BAB 3 METODOLOGI

Pada bab ini menjelaskan bagaimana metodologi yang dilakukan agar bisa membuat sebuah sistem monitoring luapan air pada selokan menggunakan metode *fuzzy*. Tipe penelitian yang dilakukan pada penelitian ini adalah implementasi dan perancangan sistem berupa *prototype*. Sistem yang dimaksud pada penelitian ini yaitu meliputi seluruh komponen yang berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dapat berjalan sesuai dengan tujuan yaitu mendeteksi luapan air dan juga sebagai sarana monitoring luapan air pada selokan. Diagram alir dari tahap-tahap metodologi penelitian yang dapat dilihat pada Gambar 3.1 dibawah ini.



Gambar 3.1 Diagram Alir Metode Penelitian

Gambar 3.1 merupakan diagram alir metode penelitian yang dimulai dari studi literatur yang masih berhubungan dengan tinjauan pustaka dan juga dasar teori tentang riset sebelumnya yang memiliki keterkaitan dengan pendeteksi banjir dengan menggunakan sensor ketinggian dan sensor kecepatan air, analisa kebutuhan, perancangan perangkat keras dan perangkat lunak, implementasi, pengujian, analisa hasil dan penarikan kesimpulan.

3.1 Studi Literatur

Pada tahap ini dibahas mengenai studi literatur yang digunakan yang mendukung penelitian Sistem Monitoring Luapan Air Pada Selokan Menggunakan Metode *Fuzzy*. Studi literatur dilakukan dengan cara membaca dan merangkum dari jurnal maupun *via online* dimana hasil dari studi literatur ini tertuang dalam bab 2. Adapun teori-teori yang digunakan sebagai bahan studi seperti yang ada dibawah ini:

1. Arduino Nano

Mempelajari fungsi dan juga pengaplikasian Arduino Nano sebagai mikrokontroler perangkat utama dari sistem yang akan digunakan. (Sumber: www.arduino.cc)

2. Sensor Ultrasonik (PING)

Penjelasan mengenai susunan pin, fungsi sensor, serta mengetahui apa yang diperlukan agar didapat sensor yang bisa menghasilkan *output* yaitu jarak antar modul dan juga objek yang akan digunakan yaitu air. (Sumber: www.parallax.com)

3. Sensor Kecepatan Air G1/2

Mempelajari cara kerja serta konfigurasi dari sensor kecepatan air G1/2 dalam menghitung kecepatan air. (Sumber: www.seeedstudio.com)

4. Modul *Wireless NRF24L01*

Mempelajari bagaimana modul *wireless NRF24L01* mengirimkan data. (Sumber: www.sparkfun.com)

5. *LED*

Mempelajari bagaimana menjadikan *LED* sebagai hasil *output* sistem. (Sumber: www.sparkfun.com)

6. Metode *Fuzzy*

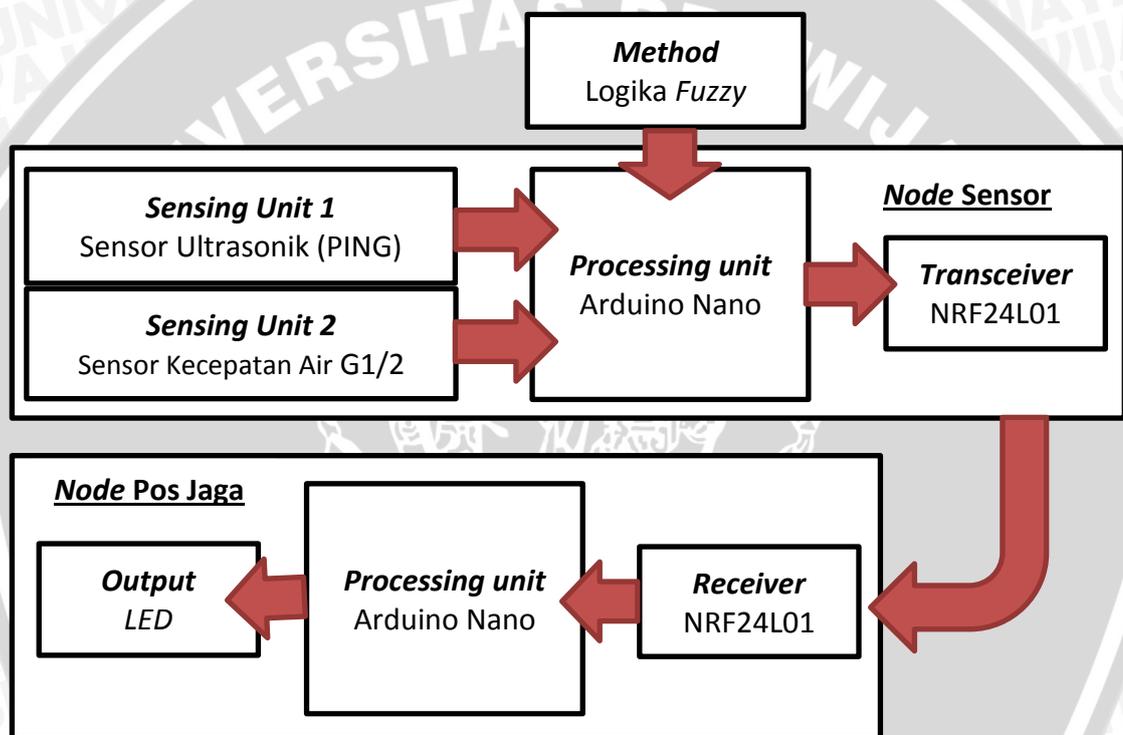
Mempelajari bagaimana mengimplementasikan logika *fuzzy* pada sistem monitoring luapan air pada selokan. (Sumber: Aplikasi Logika *Fuzzy* Sri Kusumadewi)

3.2 Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan sistem yang bertujuan agar dapat mengetahui apa yang dibutuhkan oleh sistem monitoring luapan air pada selokan agar dapat dikatakan bekerja sesuai dengan tujuan penelitian. Analisis kebutuhan akan dilakukan dengan cara mengidentifikasi masalah yang ada sehingga didapatkan proses-proses apa saja yang akan terjadi, dari proses-proses tersebut akan dianalisa sehingga akan didapatkan kebutuhan *hardware*, *software*, dan juga kebutuhan data yang didapatkan melalui uji coba.

3.3 Perancangan Sistem

Perancangan sistem merupakan tahap yang dapat dilakukan setelah melakukan studi literatur dan juga analisa kebutuhan. Pada tahap ini dilakukan perancangan yang meliputi perancangan perangkat keras dan perancangan perangkat lunak. Perancangan perangkat keras dilakukan dengan membuat 2 *node*. *Node* pertama meliputi Arduino Nano, Sensor Kecepatan air G1/2, Sensor Ultrasonik (PING), dan NRF24L01. *Node* kedua meliputi Arduino Nano, NRF24L01 dan LED. Setelah perancangan perangkat keras selesai, dilakukan perancangan perangkat lunak yang berupa perancangan program *fuzzy*, dan perancangan program pada arduino menggunakan Arduino IDE. Diagram blok perancangan sistem pendeteksi luapan air dapat dilihat pada Gambar 3.2 dibawah ini.



Gambar 3.2 Diagram Blok Sistem

Gambar 3.2 diatas adalah diagram blok perancangan sistem pada penelitian ini. Dibawah ini merupakan penjelasan dari diagram blok tersebut:

1. *Sensing Unit 1* yang digunakan dalam penelitian ini adalah sensor ultrasonik (PING) dengan *input* jarak yang terdeteksi dan menghasilkan *output* berupa ketinggian air dalam *centimeter*(cm).
2. *Sensing Unit 2* yang digunakan dalam penelitian ini adalah sensor kecepatan air G1/2 dengan *input* aliran air yang memasuki sensor dan menghasilkan *output* berupa kecepatan aliran air dalam liter per jam(L/hour).
3. *Processing Unit* yang digunakan dalam penelitian ini adalah Arduino Nano ATmega328 yang berfungsi sebagai pengolah data dari sensing maupun pengolah data yang diterima melalui NRF24L01.

4. Metode yang digunakan dalam mengolah data pada penelitian ini adalah metode *Fuzzy Sugeno*.
5. *Transceiver* yang digunakan dalam penelitian ini adalah modul NRF24L01 yang pada *node* pos jaga berfungsi sebagai *receiver*.
6. *Output* yang digunakan untuk menampilkan hasil pengolahan data adalah *LED* merah, kuning, dan hijau.

Pada gambar tersebut terdapat 2 sesi, yaitu sesi *node* sensor dan *node* pos jaga. Pada sesi *node* sensor terdapat sensor ultrasonik (PING) dan sensor kecepatan air sebagai *input* dimana data dari kedua sensor tersebut akan diolah oleh Arduino Nano menggunakan metode *fuzzy*, dan kemudian hasil dari pengolahan tersebut akan diterima oleh modul *transceiver* NRF24L01 yang selanjutnya akan dikirimkan ke modul *receiver* NRF24L01 yang ada pada *node* pos jaga.

Kemudian pada sesi *node* pos jaga juga terdapat modul NRF24L01 yang berfungsi menerima data dari modul NRF24L01 yang berada di *node* sensor, kemudian data akan menuju ke Arduino Nano yang akan mengolah data dari NRF24L01 *transceiver* sehingga *output* tersebut akan ditampilkan berupa *LED* yang akan menyala sesuai dengan data yang diterima.

3.4 Implementasi Sistem

Pada implementasi sistem akan dilakukan penerapan dari sistem monitoring luapan air pada selokan dengan cara mengacu pada tahap perancangan sistem, kemudian akan dilakukan pengujian yang bertujuan untuk mengetahui apakah sistem monitoring luapan air pada selokan tersebut sesuai yang direncanakan sebelumnya pada penelitian ini yang meliputi:

1. Implementasi *prototype* selokan.
2. Implementasi pemasangan *node* 1, yaitu *node* sensor yang dipasang di selokan.
3. Implementasi pemasangan *node* 2, yaitu *node* sensor yang dipasang di pos penjagaan.
4. Implementasi pengiriman data secara *wireless* sesuai studi literatur.
5. Implementasi dalam penggunaan logika *fuzzy*.

3.5 Pengujian dan Analisis

Pada tahap pengujian dilakukan dengan tujuan agar dapat mengetahui apakah sistem monitoring luapan air pada selokan yang telah dibuat sesuai dengan tujuan yang diinginkan atau tidak. Apabila hasil tidak sesuai dengan tujuan yang diinginkan akan dilakukan analisis kesalahan yang mengakibatkan hal tersebut terjadi yang dapat dilihat dari komponen penyusun alat maupun kode program yang telah dibuat. Pengujian dan analisis dilakukan dengan cara menguji alat sesuai dengan skenario pengujian dan dari hasil uji tersebut barulah

dilakukan sebuah analisis. Dibawah ini adalah beberapa skenario pengujian yang akan dilakukan:

1. Pengujian komponen pendeteksi ketinggian air.
2. Pengujian komponen pendeteksi kecepatan air.
3. Pengujian komponen pengolah data.
4. Pengujian pengiriman dan penerimaan data.
5. Pengujian kesesuaian data yang diterima dengan hasil yang ditampilkan komponen.
6. Pengujian validitas logika *fuzzy*.
7. Pengujian waktu pemrosesan yang dibutuhkan dalam sekali proses.

3.6 Penarikan Kesimpulan

Tahap penarikan kesimpulan dapat dilakukan apabila seluruh tahap sebelumnya yang meliputi studi literatur, analisa kebutuhan sistem, perancangan sistem, implementasi sistem dan yang terakhir adalah pengujian dan analisis telah terpenuhi. Hasil dari pengujian dan analisis dapat diambil kesimpulan yang dapat digunakan sebagai acuan apabila akan dilakukan penelitian yang terkait berikutnya. Selain kesimpulan, juga terdapat saran yang dapat digunakan untuk mengembangkan dan memperbaiki kekurangan pada sistem monitoring luapan air pada selokan ini.



BAB 4 REKAYASA KEBUTUHAN

4.1 Deskripsi Umum

Pada bab ini menguraikan rekayasa kebutuhan yang harus dipenuhi dari perancangan dan juga implementasi sistem. Sehingga diharapkan pada perancangan dan juga implementasi pada sistem monitoring luapan air pada selokan menggunakan metode *fuzzy* dapat berjalan dengan baik.

4.1.1 Perspektif Sistem

Sistem dikatakan dapat bekerja sesuai tujuan apabila sistem dapat membaca data sensor ultrasonik (PING) dan sensor kecepatan air pada alat ukur, mengolah data menggunakan logika *fuzzy* dan mengirimkan hasil olahan data tersebut secara *wireless*. Data yang dikirim dari alat ukur harus sesuai dengan data yang diterima oleh *node* yang ada pada pos jaga. Hasil olahan data yang dikeluarkan berupa nyala *LED* sesuai olahan data yang diterima yaitu hijau untuk normal, kuning untuk waspada dan merah untuk bahaya.

4.1.2 Karakteristik Pengguna

Karakteristik pengguna pada sistem ini adalah sistem ini diperuntukkan kepada masyarakat umum yang berada pada daerah yang sering mengalami peluapan air pada selokan, seistem ini diharapkan dapat memenuhi semua kebutuhan yang diperlukan oleh pengguna. Dalam sistem ini pengguna dapat melakukan monitoring air selokan dari jarak jauh dengan memanfaatkan modul *wireless* NRF24L01 dan juga memantau melalui *LED*.

4.1.3 Lingkungan Sistem Operasi

Pada penelitian ini kebutuhan lingkungan yang mendukung bekerjanya sistem yaitu :

1. Kondisi lingkungan yang kering atau tidak basah agar air tidak merusak komponen dari sistem.
2. Lingkungan di sekitar harus bersih dari interferensi gelombang yang memiliki frekuensi yang sama seperti frekuensi gelombang yang digunakan pada sistem.

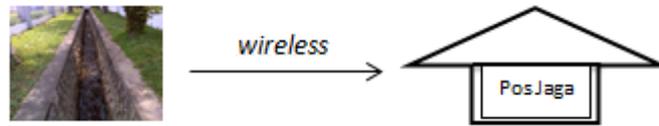
4.1.4 Asumsi dan Ketergantungan

Dibawah ini merupakan asumsi dan ketergantungan yang ada pada sistem ini diantaranya adalah:

1. Data pada sensor dapat dibaca sistem bila rangkaian yang digunakan sesuai dengan rancangan rangkaian sistem yang telah dibuat sebelumnya.
2. *Input* yang digunakan tergantung adanya air pada selokan untuk mengukur ketinggian dan kecepatan air.
3. Penempatan antara *node* sensor dan *node* yang ada di pos jaga masih dalam jangkauan modul *wireless* NRF24L01.

4.2 Rekayasa Kebutuhan

Pada sub bab rekayasa kebutuhan menguraikan semua kebutuhan yang memiliki tujuan agar sistem monitoring luapan air pada selokan ini dapat bekerja sesuai dengan tujuan yang terdiri dari kebutuhan komunikasi, kebutuhan fungsional sistem, dan kebutuhan lain yang diperlukan oleh sistem ini. Ilustrasi kebutuhan sistem dapat dilihat pada Gambar 4.1 dibawah ini.

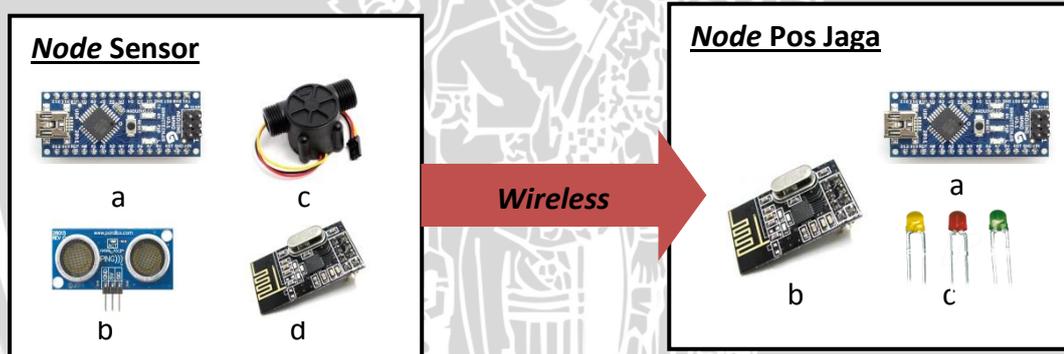


Gambar 4.1 Ilustrasi Kebutuhan Sistem

Gambar 4.1 di atas merupakan ilustrasi sistem yang digunakan untuk menentukan kebutuhan apa saja yang dibutuhkan sistem yang akan dijelaskan pada sub bab dibawah ini.

4.2.1 Kebutuhan Perangkat Keras

Gambar 4.2 dibawah ini merupakan kebutuhan perangkat keras yang dibutuhkan pada sistem ini.



Gambar 4.2 Kebutuhan Perangkat Keras

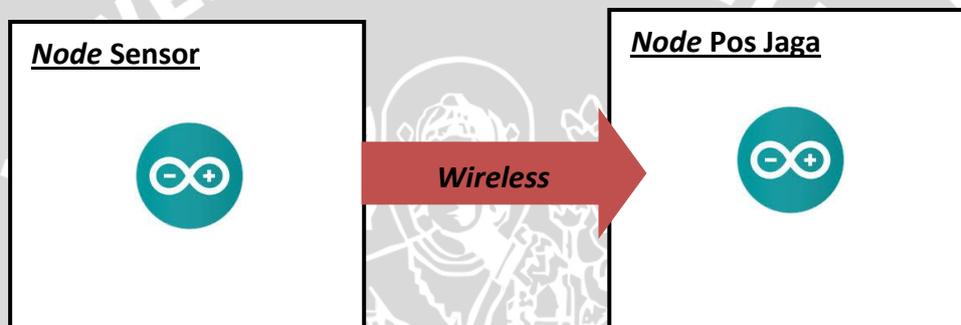
1. Pada *node sensor*:
 - a. Dibutuhkan mikrokontroler yang dapat mengolah *input* maupun *output* dari rangkaian sensor. *Input* dari sensor berupa data tinggi air pada selokan dan kecepatan air yang kemudian data tersebut diolah untuk dikirim menuju pos jaga menggunakan komunikasi *wireless* sehingga digunakan mikrokontroler Arduino Nano.
 - b. Dibutuhkan sensor yang dapat mendeteksi ketinggian air pada selokan sehingga digunakan sensor ultrasonik (PING).
 - c. Dibutuhkan sensor yang dapat mendeteksi kecepatan air pada selokan sehingga digunakan sensor kecepatan air G1/2.
 - d. Karena jarak antar selokan dan lokasi monitoring relatif jauh, dibutuhkan media komunikasi pengiriman data secara *wireless* sehingga digunakan

modul *wireless* NRF24L01 yang memanfaatkan pita gelombang RF 2.4 GHz.

2. Pada *node* pos jaga:
 - a. Dibutuhkan mikrokontroler yang dapat mengolah data yang diterima dari *node* sensor dan menampilkan data sesuai dengan data yang diterima sehingga digunakan mikrokontroler Arduino Nano.
 - b. Dibutuhkan media komunikasi *wireless* dengan gelombang radio untuk menerima data yang dikirimkan oleh *node* sensor sehingga digunakan modul *wireless* NRF24L01.
 - c. Dibutuhkan komponen yang dapat digunakan untuk menampilkan *output* sehingga digunakan *LED* dengan warna merah, kuning, dan hijau.

4.2.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan dalam membuat program pada setiap *node* pada sistem akan dijelaskan dengan diagram pada Gambar 4.3 dibawah ini.



Gambar 4.3 Kebutuhan Perangkat Lunak

1. Pada *node* sensor :

Dibutuhkan perangkat lunak yang dapat digunakan sebagai sarana membuat dan juga mengunggah program ke mikrokontroler Arduino Nano agar bisa berfungsi sebagai perangkat kontroler yang di dalamnya berisi program sesuai yang dibutuhkan *node* sensor yaitu program baca data ketinggian air, program baca data kecepatan air, program pengolahan *fuzzy*, dan program pengiriman secara *wireless*, sehingga digunakan Arduino IDE.
2. Pada *node* pos jaga :

Dibutuhkan perangkat lunak yang dapat digunakan sebagai sarana membuat dan juga mengunggah program ke mikrokontroler Arduino Nano agar dapat berfungsi sebagai perangkat kontroler yang di dalamnya berisi program yang dibutuhkan *node* pos jaga yaitu program penerimaan data dan program menhidupkan *LED*, sehingga digunakan Arduino IDE.

4.2.3 Kebutuhan Fungsional

Kebutuhan fungsional sistem merupakan kebutuhan yang harus dipenuhi agar sistem dikatakan telah bekerja sesuai dengan tujuan. Beberapa kebutuhan fungsional sistem ini akan dijelaskan pada beberapa sub bab dibawah ini:

4.2.3.1 Fungsi pembacaan data sensor ultrasonik (PING)

Fungsi ini membuat sistem harus dapat membaca data sensor yang berupa ukuran tinggi air pada selokan yang menjadi *input* pada mikrokontroler. Fungsi ini merupakan fungsi yang paling diprioritaskan karena data ketinggian air tersebut merupakan salah satu data utama yang digunakan dalam sistem ini. Nilai yang didapatkan berupa ketinggian air dalam *centimeter* (cm).

4.2.3.2 Fungsi pembacaan data sensor kecepatan air G1/2

Fungsi ini membuat sistem harus dapat membaca data sensor yang berupa ukuran kecepatan air pada selokan yang menjadi *input* pada mikrokontroler. Fungsi ini merupakan fungsi yang paling di prioritaskan karena data ketinggian tersebut merupakan salah satu data utama yang digunakan dalam sistem ini. Nilai yang didapatkan berupa kecepatan air dalam liter per jam (*L/hour*).

4.2.3.3 Fungsi pengiriman data secara *wireless*

Fungsi ini membuat sistem harus mampu mengirim data melalui komunikasi *wireless*. Fungsi ini juga termasuk yang paling diprioritaskan karena merupakan dasar lingkungan yang akan diterapkan yaitu secara *wireless* atau nirkabel dengan data yang dikirim berbentuk integer sebesar 1 *byte*.

4.2.3.4 Fungsi penerimaan data secara *wireless*

Fungsi ini mengharuskan sistem berupa *node* pos jaga mampu menerima data melalui komunikasi *wireless* dengan benar. Fungsi ini termasuk yang paling diprioritaskan karena merupakan dasar dari lingkungan yang diterapkan yaitu secara nirkabel dengan *delay* 1 detik karena pemrosesan pada *node* sensor membutuhkan waktu rata 1 detik.

4.2.3.5 Fungsi penentuan kondisi menggunakan logika *fuzzy*

Fungsi ini mengharuskan sistem dapat mengklasifikasikan kondisi sesuai dengan *input* yang diterima dan diolah pada Arduino Nano. Fungsi ini termasuk yang paling di prioritaskan karena merupakan tujuan dari sistem yaitu untuk menentukan output yang akan ditampilkan.

4.2.3.6 Fungsi menampilkan hasil olahan data

Fungsi ini mengharuskan sistem dapat menampilkan hasil olahan data sesuai dengan masukan yang diterima. Fungsi ini termasuk yang paling diprioritaskan karena merupakan hasil akhir dari sistem yaitu untuk menampilkan hasil olahan data menggunakan *LED*, yaitu hijau jika dalam kondisi normal, kuning jika dalam kondisi waspada, merah jika dalam kondisi bahaya.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini menjelaskan perancangan dan juga implementasi sistem monitoring luapan air pada selokan menggunakan metode *fuzzy* yang terdiri dari perancangan perangkat keras dan perancangan perangkat lunak yang ada pada *node* sensor dan *node* pos jaga.

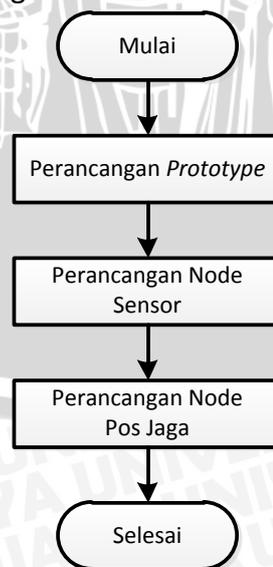
5.1 Perancangan Sistem

Tahap perancangan sistem menjelaskan langkah-langkah yang dilakukan dalam perancangan perangkat keras maupun perangkat lunak yang ada pada *node* sensor dan *node* pos jaga.

5.1.1 Perancangan Perangkat Keras

Perancangan perangkat keras di sistem pendeteksi luapan air ini dimulai dari perancangan *prototype* selokan, kemudian perancangan pada *node* sensor dimana rangkaiannya terdiri dari *sensing unit* yang berupa sensor ultrasonik (PING) dan sensor kecepatan air G1/2, dan modul *wireless* NRF24L01 sebagai *transceiver* yaitu sarana penerima sekaligus pengirim data secara nirkabel menggunakan gelombang radio. Pada *node* sensor tersebut juga terdapat mikrokontroler Arduino Nano yang digunakan sebagai *processing unit*. Mikrokontroler Arduino Nano memproses data yang masuk dari sensor dan mengirimkan data yang telah diolah secara *wireless* melalui modul NRF24L01 ke *node* pos jaga.

Sedangkan perancangan perangkat keras pada *node* pos jaga terdiri dari *processing unit* yang sama dengan yang ada pada *node* sensor yaitu mikrokontroler Arduino Nano, *transceiver* yang berupa modul *wireless* NRF24L01 dan *LED* yang digunakan sebagai *output*. Gambar 5.1 dibawah ini merupakan diagram alir perancangan perangkat keras.

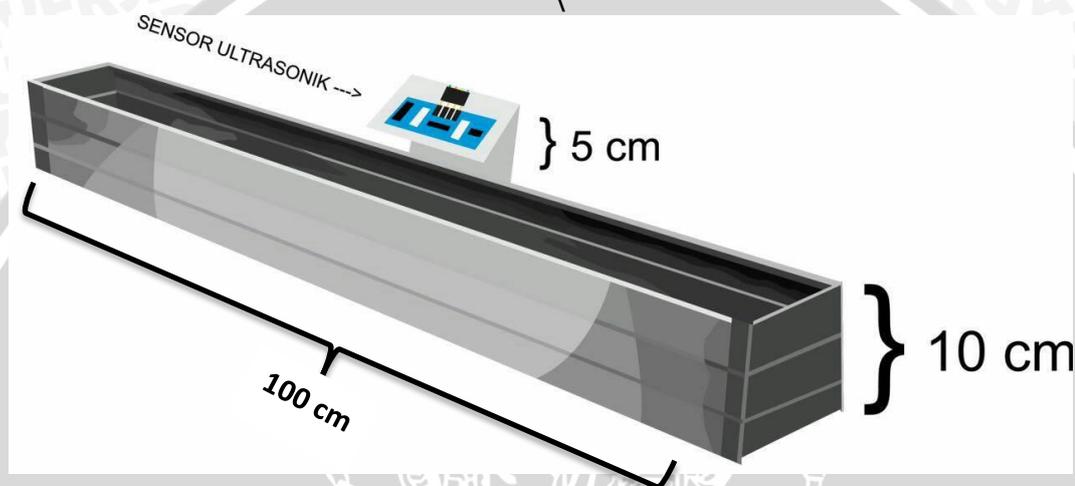


Gambar 5.1 Diagram Alir Perancangan Perangkat Keras

Gambar 5.1 diatas merupakan gambar diagram alir perancangan perangkat keras, dimana hal pertama yang harus dilakukan adalah merancang *prototype* selokan, kemudian membuat rancangan rangkaian *node* sensor, dan yang terakhir adalah membuat rancangan rangkaian *node* pos jaga.

5.1.1.1 Perancangan *Prototype*

Perancangan *prototype* yang akan dilakukan merupakan perancangan *prototype* selokan yang terbuat dari pipa pvc, akrilik batangan, dan *bracket* sensor ultrasonik (PING). Gambar perancangan *prototype* dapat dilihat pada Gambar 5.2 dibawah ini.

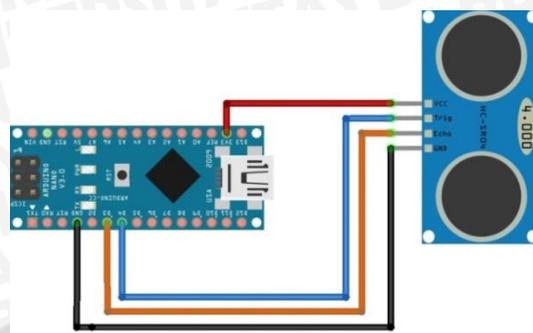


Gambar 5.2 Rancangan *Prototype* Selokan

Gambar 5.2 diatas merupakan gambar *prototype* ultrasonik dimana pvc diatur dengan tinggi 10 cm, akrilik yang digunakan sebagai penyangga panjangnya 5 cm, dan tempat sensor ultrasonik (PING) yang berupa *bracket* di lem *horizontal* terhadap penyangga akrilik.

5.1.1.2 Perancangan Perangkat Keras Pada *Node* Sensor

Perangkat keras pada *node* sensor terdiri sebuah mikrokontroler Arduino Nano, sensor ultrasonik (PING), sensor kecepatan air, dan NRF24L01. Komponen tersebut akan digabungkan hingga menjadi sebuah *node* dengan tiga sub tahap perancangan rangkaian, yaitu rangkaian sensor untrasonik (PING) yang disambungkan ke mikrokontroler, rangkaian sensor kecepatan air G1/2 yang disambungkan ke mikrokontroler dan rangkaian modul *wireless* NRF24L01 yang disambungkan ke mikrokontroler. Rangkaian mikrokontroler dengan kedua sensor dibuat agar sistem dapat membaca data sensor, sedangkan rangkaian mikrokontroler dengan modul *wireless* NRF24L01 agar dapat mengirim hasil olahan data secara nirkabel. Gambar 5.3 dibawah ini merupakan gambar rangkaian mikrokontroler dengan sensor ultrasonik (PING) dengan Tabel 5.1 sebagai tabel keterangan pin sensor ultrasonik (PING) yang terhubung dengan Arduino Nano.



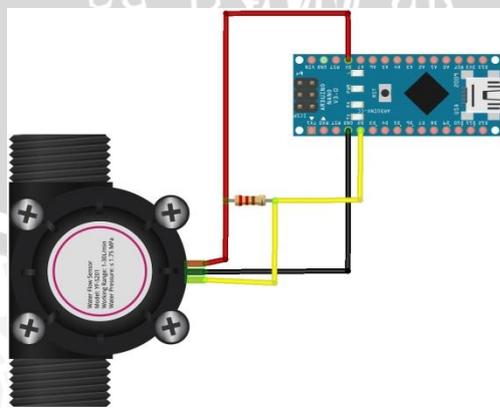
Gambar 5.3 Rancangan Sensor ultrasonik (PING) Ke Mikrokontroler

Tabel 5.1 Keterangan Pin Sensor Ultrasonik (PING)

Pin Sensor Ultrasonik (PING)	Pin Arduino Nano
VCC	3V3
Triger	D3
Echo	D4
GND	GND

Gambar 5.3 diatas merupakan rancangan Arduino Nano sebagai pemroses data hasil ukur dan sensor ultrasonik (PING) agar dapat digunakan sebagai alat pengukur ketinggian pada selokan. Dalam rangkain tersebut sensor ultrasonik memiliki empat pin yang akan tersambung pada Arduino Nano yang terdiri dari pin VCC, Triger, Echo, dan GND. Seperti yang ada pada tabel keterangan, pin VCC pada sensor terhubung menggunakan kabel jumper ke pin 3V3 yang ada pada Arduino Nano, pin Triger terhubung menggunakan kabel jumper ke pin D3 yang ada pada Arduino Nano, pin Echo terhubung menggunakan kabel jumper ke pin D4 yang ada pada Arduino Nano, dan pin GND terhubung menggunakan kabel jumper ke pin GND yang ada pada Arduino Nano.

Setelah membuat rancangan rangkaian mikrokontroler dengan modul sensor ultrasonik (PING), dibuat rancangan rangkaian mikrokontroler dengan modul sensor kecepatan air G1/2. Gambar rangkaian tersebut akan ditunjukkan Gambar 5.4.



Gambar 5.4 Rancangan Kecepatan Air G1/2 Ke Mikrokontroler

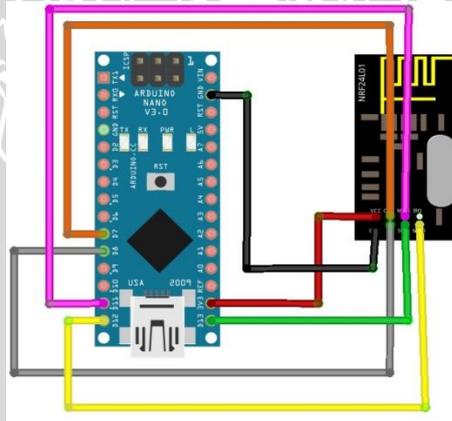
Tabel 5.2 yang merupakan keterangan pin sensor kecepatan air G1/2 yang terhubung dengan Arduino Nano.

Tabel 5.2 Keterangan Pin Sensor Kecepatan Air G1/2

Pin Sensor Kecepatan Air G1/2	Pin Arduino Nano
5V	VCC
GND	GND
D2	SIG

Gambar 5.4 diatas merupakan rancangan Arduino Nano sebagai pemroses data hasil ukur dan sensor kecepatan air G1/2 sebagai alat pengukur kecepatan air pada selokan. Dalam rangkain tersebut sensor kecepatan air G1/2 memiliki tiga pin yang akan tersambung pada Arduino Nano yang terdiri dari pin VCC, GND, dan SIG. Seperti yang ada pada tabel keterangan, pin VCC pada sensor terhubung menggunakan kabel *jumper* ke pin 5V yang ada pada Arduino Nano, pin GND terhubung menggunakan kabel *jumper* ke pin GND yang ada pada Arduino Nano, dan pin SIG terhubung menggunakan kabel *jumper* ke pin D2 yang ada pada Arduino Nano.

Setelah membuat rancangan rangkaian mikrokontroler dengan modul sensor kecepatan air G1/2, dibuat rancangan rangkaian mikrokontroler dengan modul *wireless* NRF24L01 sebagai modul *wireless*. Gambar rangkaian tersebut akan ditunjukkan Gambar 5.5.



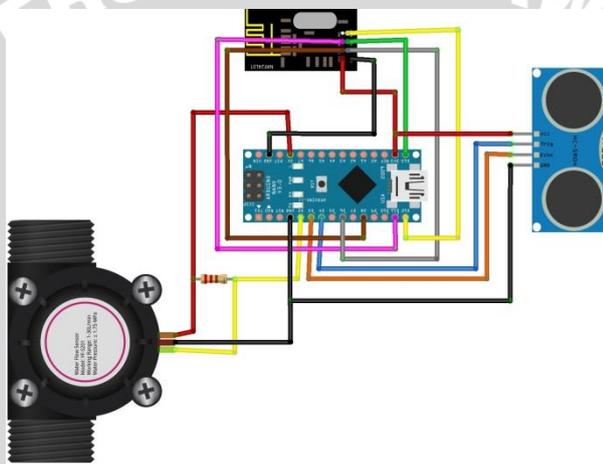
Gambar 5.5 Rancangan Modul NRF24L01 Ke Mikrokontroler

Dibawah ini adalah Tabel 5.3 yang merupakan keterangan pin NRF24L01 yang terhubung pada Arduino Nano.

Tabel 5.3 Keterangan Pin Modul NRF24L01 Node Sensor

Pin NRF24L01	Pin Arduino Nano
VCC	3V3
GND	GND
CSN	D7
CE	D8
MOSI	D11
SCK	D13
MISO	D12

Gambar 5.5 diatas merupakan rancangan Arduino Nano sebagai *processing unit* dan modul NRF24L01 sebagai sarana pengiriman data secara nirkabel. Dalam rangkain tersebut modul NRF24L01 memiliki tujuh pin yang akan tersambung pada Arduino Nano yang terdiri dari pin VCC, GND, CSN, CE, MOSI, SCK, dan MISO. Seperti yang ada pada tabel keterangan, pin VCC yang ada pada NRF24L01 terhubung menggunakan kabel *jumper* ke pin 3V3 yang ada pada Arduino Nano, pin GND terhubung menggunakan kabel *jumper* ke pin GND yang ada pada Arduino Nano, pin CSN terhubung menggunakan kabel *jumper* ke pin D7 yang ada pada Arduino Nano, pin CE terhubung menggunakan kabel *jumper* ke pin D8 yang ada pada Arduino Nano, pin MOSI terhubung menggunakan kabel *jumper* ke pin D11 yang ada pada Arduino Nano, pin SCK terhubung menggunakan kabel *jumper* ke pin D13 yang ada pada Arduino Nano, dan pin MISO terhubung menggunakan kabel *jumper* ke pin D12 yang ada pada Arduino Nano. Rangkaian *node* sensor akan ditunjukkan Gambar 5.6.

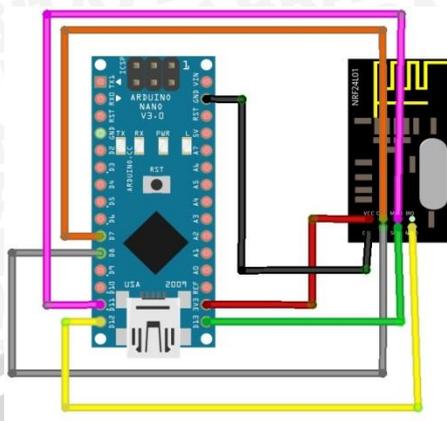


Gambar 5.6 Rancangan Node Sensor

Gambar 5.6 merupakan gambar rancangan pada *node* sensor. Rancangan *node* sensor merupakan gabungan dari rancangan antara mikrokontroler Arduino Nano, modul sensor ultrasonik (PING), modul sensor kecepatan air G1/2, dan modul *wireless* NRF24L01.

5.1.1.3 Perancangan Perangkat Keras Pada Node Pos Jaga

Perangkat keras pada *node* pos jaga terdiri dari sebuah mikrokontroler Arduino Nano, LED, dan NRF24L01. Komponen tersebut akan digabungkan hingga menjadi sebuah *node* dengan dua sub tahap perancangan rangkaian yaitu rangkaian modul *wireless* NRF24L01 yang disambungkan ke mikrokontroler dan rangkaian LED yang disambungkan ke mikrokontroler. Rangkaian mikrokontroler dengan modul *wireless* NRF24L01 dirancang agar sistem dapat menerima hasil olahan data secara nirkabel, sedangkan rangkaian mikrokontroler dengan LED dibuat agar dapat menunjukkan *output* apa yang keluar. Gambar 5.7 merupakan gambar rangkaian mikrokontroler dengan modul NRF24L01.



Gambar 5.7 Rancangan Mikrokontroler Dengan Modul NRF24L01

Tabel 5.4 yang merupakan keterangan pin NRF24L01 yang terhubung pada Arduino Nano.

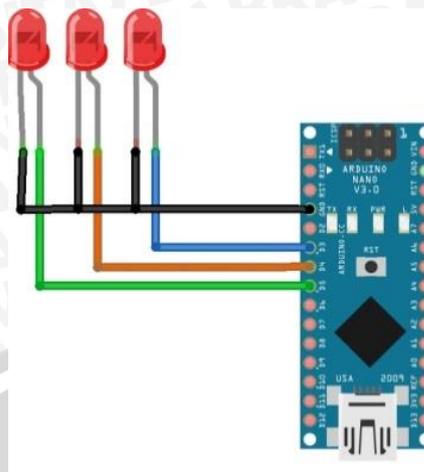
Tabel 5.4 Keterangan Pin Modul NRF24L01 Node Pos Jaga

Pin NRF24L01	Pin Arduino Nano
VCC	3V3
GND	GND
CSN	D7
CE	D8
MOSI	D11
SCK	D13
MISO	D12

Gambar 5.7 diatas merupakan rancangan Arduino Nano sebagai *processing unit* dan modul NRF24L01 sebagai sarana penerimaan data secara nirkabel. Sama halnya dengan modul NRF24L01 yang ada pada *node sensor*, dalam rangkain tersebut modul NRF24L01 memiliki tujuh pin yang akan tersambung pada Arduino Nano yang terdiri dari pin VCC, GND, CSN, CE, MOSI, SCK, dan MISO. Seperti yang ada pada tabel keterangan, pin VCC yang ada pada NRF24L01 terhubung menggunakan kabel *jumper* ke pin 3V3 yang ada pada Arduino Nano, pin GND terhubung menggunakan kabel *jumper* ke pin GND yang ada pada Arduino Nano, pin CSN terhubung menggunakan kabel *jumper* ke pin D7 yang ada pada Arduino Nano, pin CE terhubung menggunakan kabel *jumper* ke pin D8 yang ada pada Arduino Nano, pin MOSI terhubung menggunakan kabel *jumper* ke pin D11 yang ada pada Arduino Nano, pin SCK terhubung menggunakan kabel *jumper* ke pin D13 yang ada pada Arduino Nano, dan pin MISO terhubung menggunakan kabel *jumper* ke pin D12 yang ada pada Arduino Nano.

Setelah membuat rancangan rangkaian mikrokontroler dengan modul modul NRF24L01, dibuat rancangan rangkaian mikrokontroler dengan LED. Gambar rangkaian tersebut ditunjukkan Gambar 5.8.





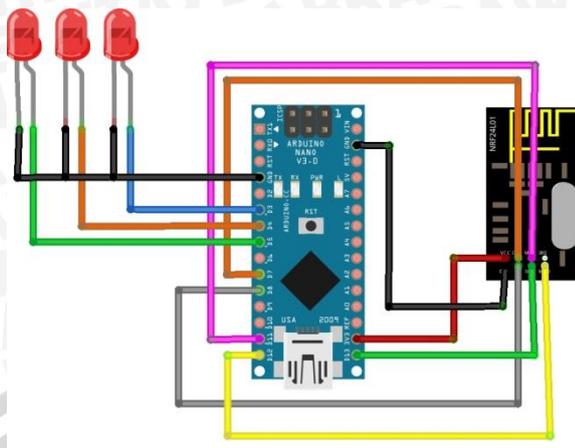
Gambar 5.8 Rancangan Mikrokontroler Dengan LED

Tabel 5.5 yang merupakan keterangan LED yang terhubung pada pin Arduino Nano.

Tabel 5.5 Keterangan LED

LED	Pin Arduino Nano
Anoda LED1	D3
Katoda LED1	GND
Anoda LED2	D4
Katoda LED2	GND
Anoda LED3	D5
Katoda LED3	GND

Dalam Gambar 5.8 diatas merupakan rancangan mikrokontroler dengan LED yang berfungsi sebagai *output* dari alat. Setiap satu LED memiliki anoda dan katoda yang akan tersambung pada mikrokontroler Arduino Nano. Seperti yang dapat dilihat pada tabel keterangan, anoda LED1 terhubung menggunakan kabel *jumper* ke pin D3 yang ada pada Arduino Nano, katoda LED1 akan terhubung menggunakan kabel *jumper* ke GND yang ada pada Arduino Nano, anoda LED2 terhubung menggunakan kabel *jumper* ke pin D4 yang ada pada Arduino Nano, katoda LED2 akan terhubung menggunakan kabel *jumper* ke GND yang ada pada Arduino Nano, anoda LED3 terhubung menggunakan kabel *jumper* ke pin D5 yang ada pada Arduino Nano, katoda LED3 akan terhubung menggunakan kabel *jumper* ke GND yang ada pada Arduino Nano. Rancangan dari *node* pos jaga akan ditunjukkan Gambar 5.9.



Gambar 5.9 Rancangan *Node* Pos Jaga

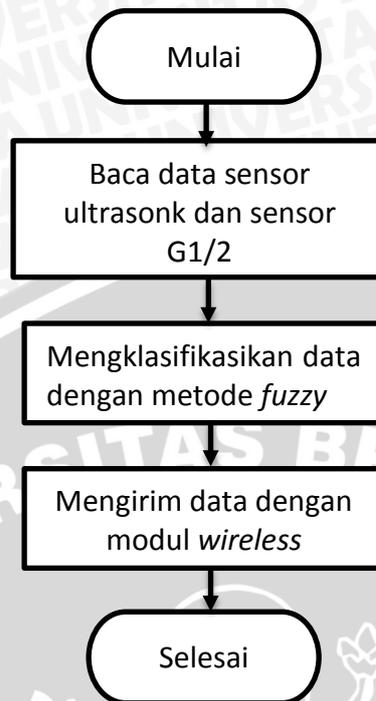
Gambar 5.9 diatas merupakan rancangan dari *node* pos jaga yang terdiri dari rancangan antara mikrokontroler Arduino Nano, modul *wireless* NRF24L01 dan LED.

5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem pendeteksi luapan air ini dilakukan agar kedua *node*, yaitu *node* sensor dan *node* pos jaga dapat bekerja dengan baik yaitu dapat membaca sensor, memproses dengan metode *fuzzy*, mengirim, dan menerima data, dimana dalam perancangannya dibutuhkan perangkat lunak yaitu Arduino IDE.

5.1.2.1 Perancangan Perangkat Lunak Pada *Node* Sensor

Perancangan perangkat lunak pada *node* sensor terdiri dari perancangan program pada Arduino IDE untuk sensor ultrasonik (PING) agar dapat memperoleh data tinggi air pada selokan dan sensor kecepatan air G1/2 agar dapat memperoleh data kecepatan air pada selokan, selanjutnya membuat program agar Arduino Nano dapat memproses data yang masuk dengan menggunakan logika *fuzzy* untuk megklasifikasikan kondisi apa yang akan menjadi *output*, kemudian membuat program agar NRF24L01 dapat mengirim data ke *node* yang ada di pos jaga. Diagram alir dalam perancangan perangkat lunak pada *node* sensor akan ditunjukkan Gambar 5.10.

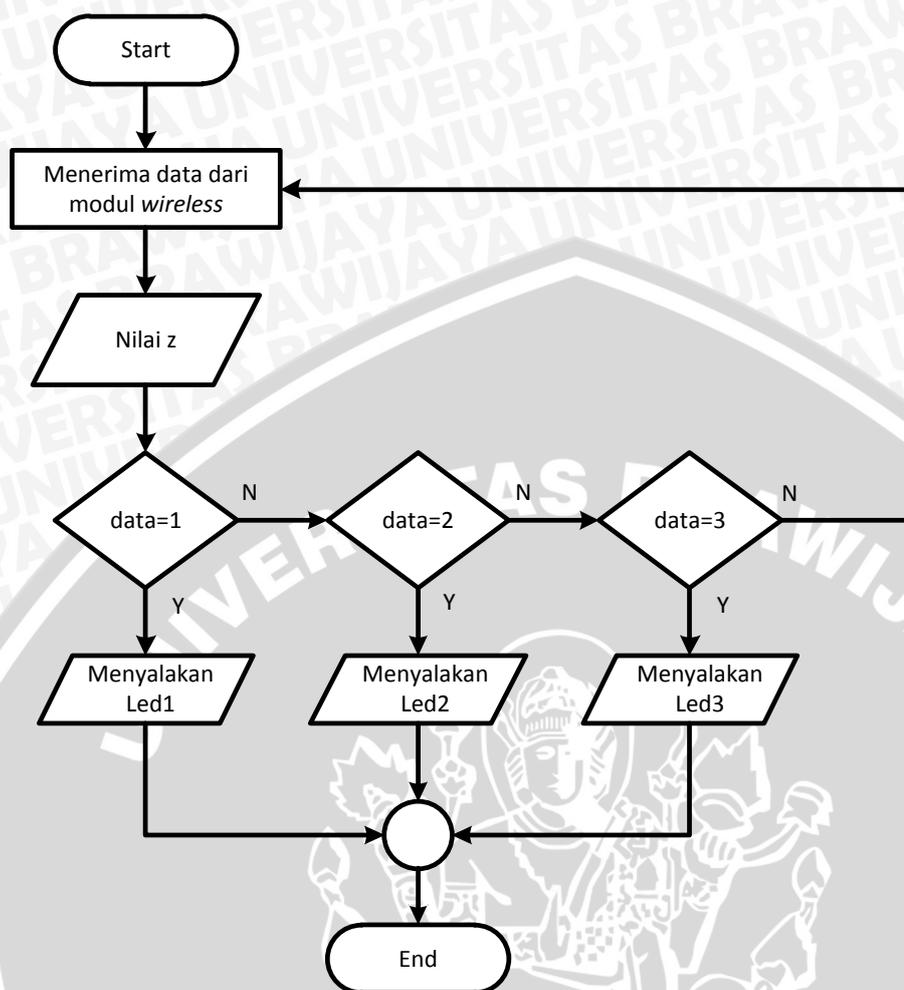


Gambar 5.10 Diagram Alir Perancangan Perangkat Lunak Pada *Node* Sensor

Gambar 5.10 diatas merupakan diagram alir perancangan perangkat lunak pada *node* sensor yang dimulai dari membaca data sensor ultrasonik dan membaca data sensor kecepatan air G1/2 yang berupa kecepatan air dan ketinggian air, setelah data diperoleh akan diolah menggunakan metode *fuzzy*, dan kemudian data hasil olahan akan dikirim menggunakan modul *wireless* ke *node* pos jaga.

5.1.2.2 Perancangan Perangkat Lunak Pada *Node* Pos Jaga

Perancangan perangkat lunak pada *node* pos jaga diperlukan agar sistem perangkat pengiriman nirkabel yang ada pada *node* pos jaga dapat menerima data dan mengeluarkan *output* yang berupa nyala dari *LED*. Agar dapat membuat perangkat lunak tersebut dibutuhkan *tools* Arduino IDE untuk memprogram Arduino Nano. Diagram alir dalam perancangan perangkat lunak pada *node* pos jaga akan ditunjukkan Gambar 5.11.

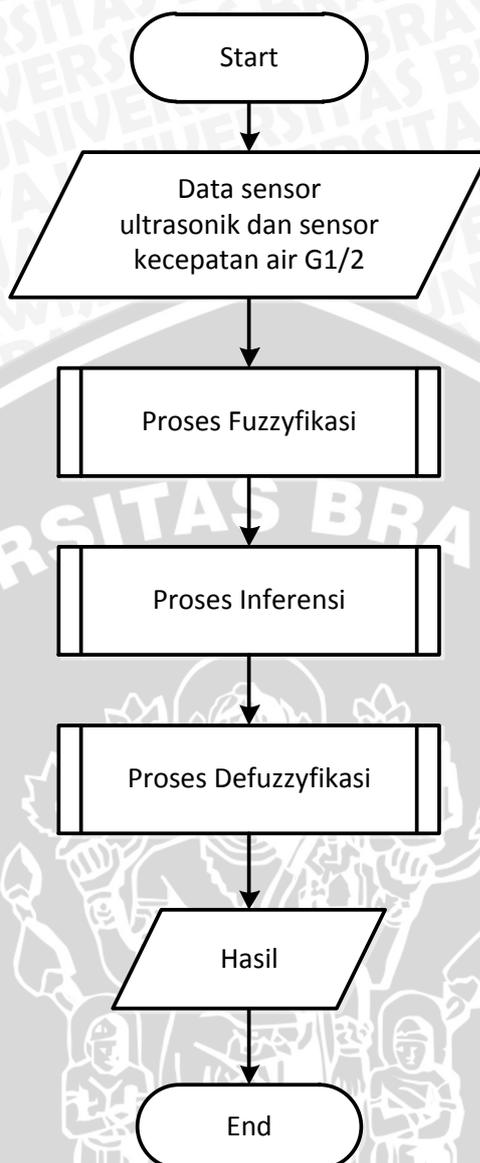


Gambar 5.11 Diagram Alir Perancangan Perangkat Lunak Pada *Node* Pos Jaga

Gambar 5.11 diatas merupakan diagram alir perancangan perangkat lunak pada *node* pos jaga yang dimulai dari menerima data dari pos jaga. Data yang diterima berupa nilai *z* yang digunakan sebagai penentu *LED* mana yang akan menyala ketika menerima data. Apabila data yang diterima adalah satu (*data=1*) *LED* yang akan menyala adalah *LED1*, apabila data yang diterima adalah dua(*data=2*) maka yang akan menyala adalah *LED2*, apabila data yang diterima adalah tiga(*data=3*), maka yang akan menyala adalah *LED3*, dan apabila data yang diterima bukan satu,dua, maupun tiga proses akan kembali mengulang dari awal lagi yaitu menerima data dari modul *wireless*.

5.1.3 Perancangan Logika *Fuzzy*

Dalam perancangan logika *fuzzy* ini akan dibahas tahap-tahap yang akan dilakukan yang meliputi proses pembentukan himpunan keanggotaan *fuzzy* atau yang biasa disebut *fuzzifikasi*, proses inferensi, dan proses defuzzifikasi yang merupakan penegasan dengan metode *height method*. Diagram alir logika *fuzzy* akan ditunjukkan Gambar 5.12.



Gambar 5.12 Diagram Alir Logika Fuzzy

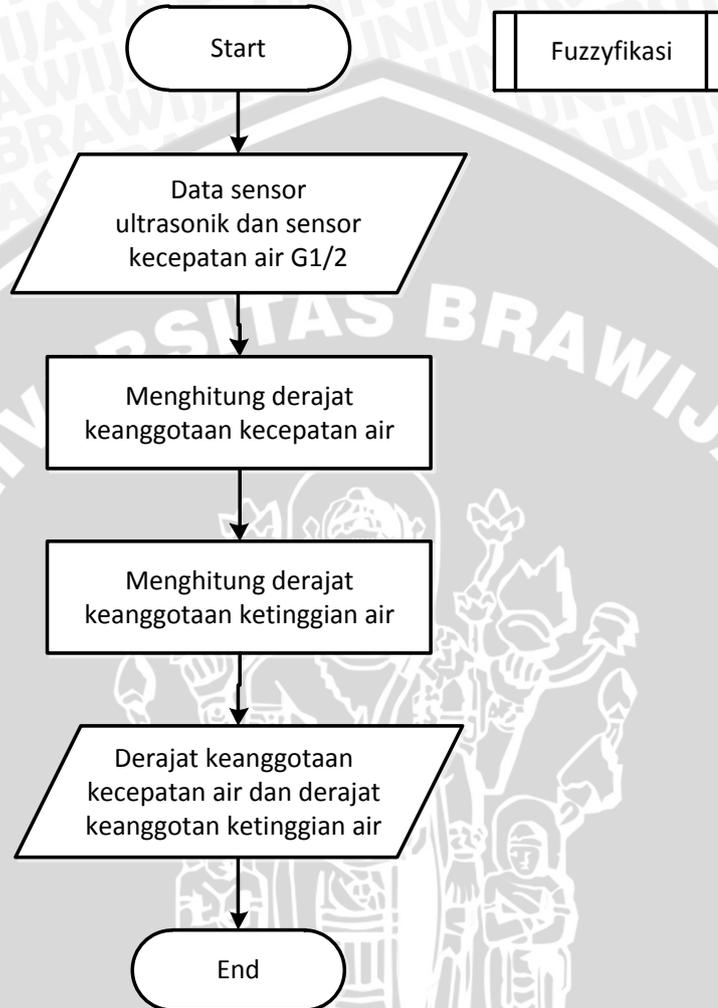
Gambar 5.12 diatas merupakan diagram alir logika yang dimulai dari menerima data sensor ultrasonik dan sensor kecepatan air G1/2 yang berupa kecepatan air dan ketinggian air, kemudian akan dimasukkan ke dalam proses *fuzzifikasi*, proses inferensi, proses defuzzifikasi, hingga mendapatkan hasil yang diinginkan.

5.1.4 Perancangan Fuzzy Sugeno

Dalam perancangan *Fuzzy Sugeno* membahas tentang tahap-tahap desain *Fuzzy Sugeno* dimulai dari memuat variabel *input*, menentukan jumlah *membership function* dan juga membuat *rule fuzzy*. Dibawah ini adalah langkah-langkah dalam pembuatan *Fuzzy Sugeno*:

5.1.4.1 Fuzzifikasi

Dalam perancangan logika *Fuzzy*, langkah utama adalah *Fuzzifikasi* yaitu menentukan himpunan keanggotaan yang terdiri dari variabel *input* dan *output*. Diagram alir proses *Fuzzifikasi* akan ditunjukkan Gambar 5.13.



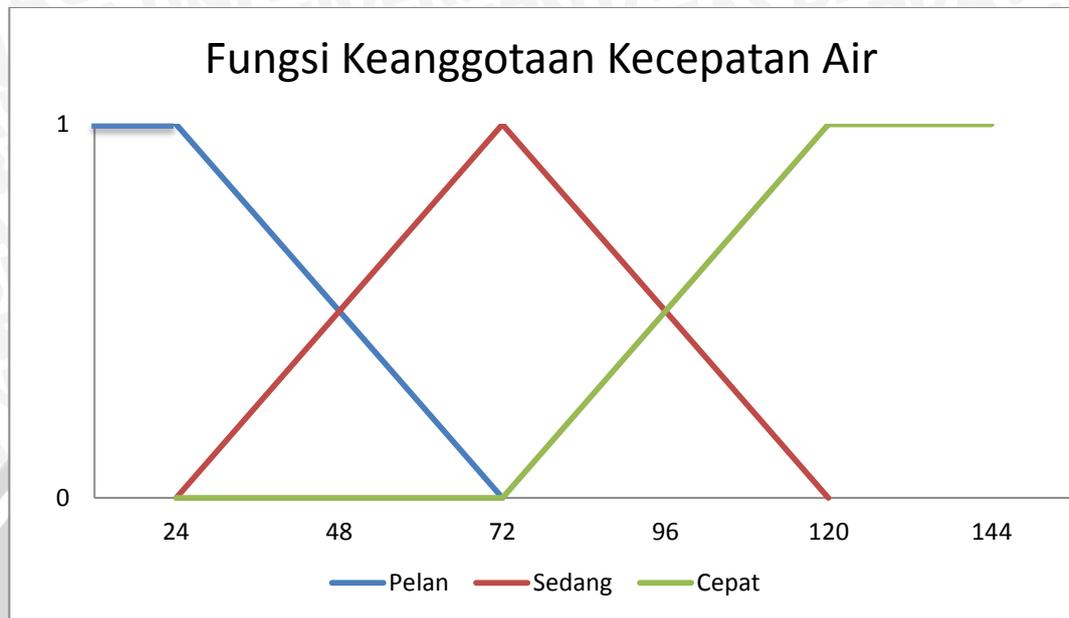
Gambar 5.13 Diagram Alir Proses Fuzzifikasi

Gambar 5.13 diatas merupakan diagram alir proses *fuzzifikasi* yang dimulai dari menerima data dari sensor ultrasonik dan sensor kecepatan air G1/2 yang berupa ketinggian air dan kecepatan air, kemudian data tersebut akan digunakan untuk menghitung derajat keanggotaan kecepatan air, lalu menghitung derajat keanggotaan ketinggian air dan hasil akhirnya akan didapat data derajat keanggotaan kecepatan air dan derajat keanggotaan ketinggian air.

Fuzzifikasi adalah proses mengubah nilai dari crisp menjadi *input fuzzy* untuk menghitung derajat keanggotaan terhadap masing-masing fungsi keanggotaan. Dibawah ini ini merupakan perhitungan derajat keanggotaan dari ketinggian air dan kecepatan air:

a. Kecepatan Air

Fungsi keanggotaan kecepatan air akan ditunjukkan Gambar 5.14.



Gambar 5.14 Fungsi Keanggotaan Kecepatan Air

Gambar 5.14 diatas merupakan fungsi keanggotaan kecepatan air yang memiliki *range* dari 0 hingga 144. Terdapat tiga derajat keanggotaan yaitu kurang dari 72 menunjukkan pelan, antara 24 hingga 120 menunjukkan sedang, dan antara 72 hingga 144 menunjukkan cepat.

Dari gambar diatas didapatkan dimasukkan ke dalam rumus untuk mencari derajat keanggotaan representasi kurva segitiga hingga didapatkan rumus dibawah ini:

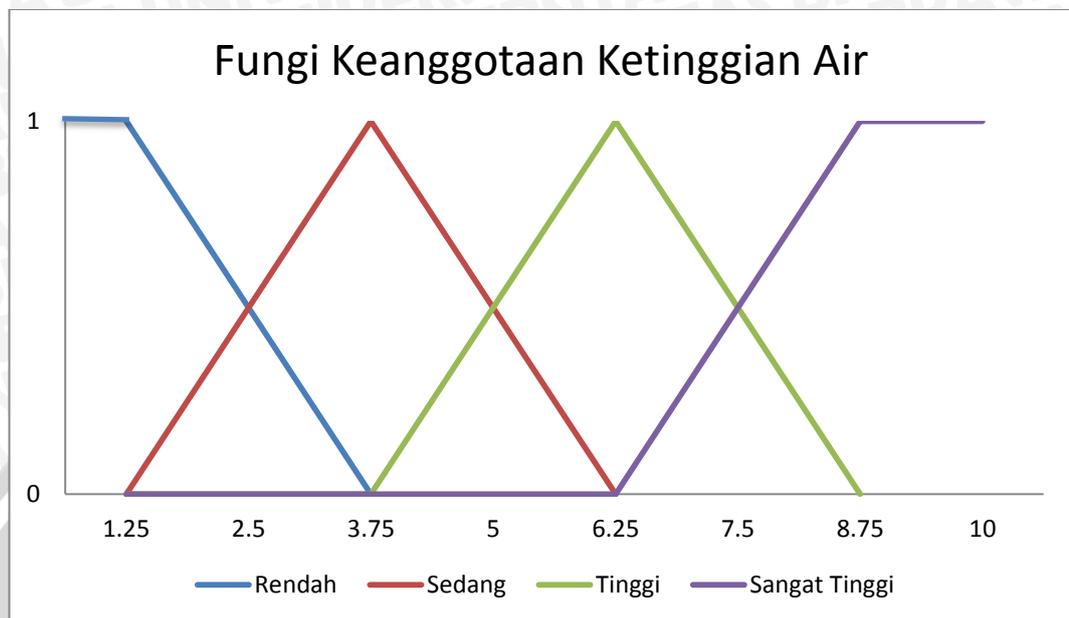
$$\mu_{Pelan}(x) = \begin{cases} 1, & x < 0 \\ \frac{72 - x}{72 - 24}, & 0 \leq x \leq 72 \\ 0, & x > 72 \end{cases}$$

$$\mu_{Sedang}(x) = \begin{cases} 0, & x < 24 \text{ atau } x > 72 \\ \frac{x - 24}{72 - 24}, & 24 \leq x \leq 72 \\ \frac{120 - x}{120 - 72}, & 72 \leq x \leq 120 \end{cases}$$

$$\mu_{Cepat}(x) = \begin{cases} 0, & x < 72 \\ \frac{x - 72}{120 - 72}, & 72 \leq x \leq 120 \\ 1, & x > 120 \end{cases}$$

b. Ketinggian Air

Fungsi keanggotaan ketinggian air akan ditunjukkan Gambar 5.15.



Gambar 5.15 Fungsi Keanggotaan Ketinggian Air

Gambar 5.15 diatas merupakan fungsi keanggotaan ketinggian air yang memiliki *range* dari 0 hingga 10. Terdapat empat derajat keanggotaan yaitu kurang dari 3,75 menunjukkan rendah, antara 1,25 hingga 6,25 menunjukkan sedang, antara 3,75 hingga 8,75 menunjukkan tinggi, dan antara 6,25 hingga 10 menunjukkan sangat tinggi.

Dari gambar diatas didapatkan dimasukkan ke rumus dalam mencari derajat keanggotaan representasi kurva segitiga hingga didapatkan rumus dibawah ini:

$$\mu_{Rendah}(x) = \begin{cases} 1, & x < 0 \\ \frac{3.75 - x}{3.75 - 1.25}, & 0 \leq x \leq 3.75 \\ 0, & x > 3.75 \end{cases}$$

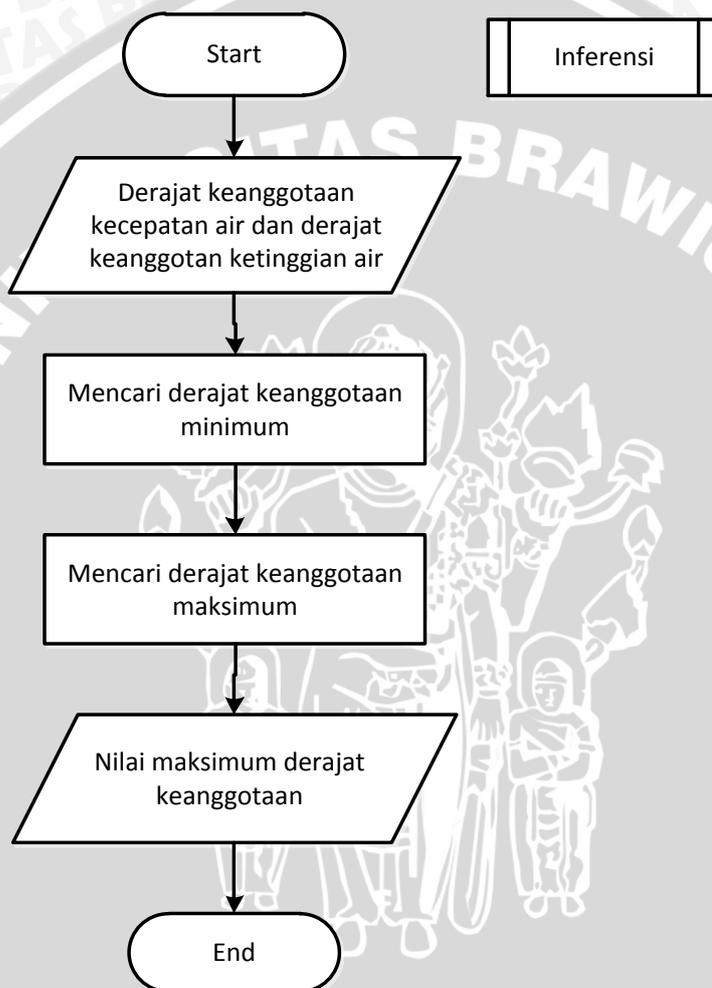
$$\mu_{Sedang}(x) = \begin{cases} 0, & x < 1.25 \text{ atau } x > 3.75 \\ \frac{x - 1.25}{3.75 - 1.25}, & 1.25 \leq x \leq 3.75 \\ \frac{6.25 - x}{6.25 - 3.75}, & 3.75 \leq x \leq 6.25 \end{cases}$$

$$\mu_{Tinggi}(x) = \begin{cases} 0, & x < 3.75 \text{ atau } x > 6.25 \\ \frac{x - 3.75}{6.25 - 3.75}, & 3.75 \leq x \leq 6.25 \\ \frac{8.75 - x}{8.75 - 6.25}, & 6.25 \leq x \leq 8.75 \end{cases}$$

$$\mu_{Sangat\ Tinggi}(x) = \begin{cases} 0, & x < 6.25 \\ \frac{x - 6.25}{8.75 - 6.25}, & 6.25 \leq x \leq 8.75 \\ 1, & x > 8.75 \end{cases}$$

5.1.4.2 Inferensi

Setelah proses *fuzzifikasi* dilakukan proses inferensi dengan diagram alir seperti yang ada Gambar 5.16.



Gambar 5.16 Diagram Alir Proses Inferensi

Gambar 5.16 diatas merupakan gambar diagram alir dari proses inferensi yang dimulai dari mengolah data hasil dari proses *fuzzifikasi*, yaitu derajat keanggotaan kecepatan air dan derajat keanggotaan ketinggian air. Dari data tersebut akan dibandingkan dan diambil nilai *minimum*nya. Setelah itu akan dimasukkan ke *rule fuzzy* dan kemudian dibandingkan lagi yang nantinya akan diperoleh nilai maksimumnya. Tabel *rule fuzzy* pada sistem ini akan ditunjukkan pada Tabel 5.6.

Tabel 5.6 Keterangan Rule Fuzzy

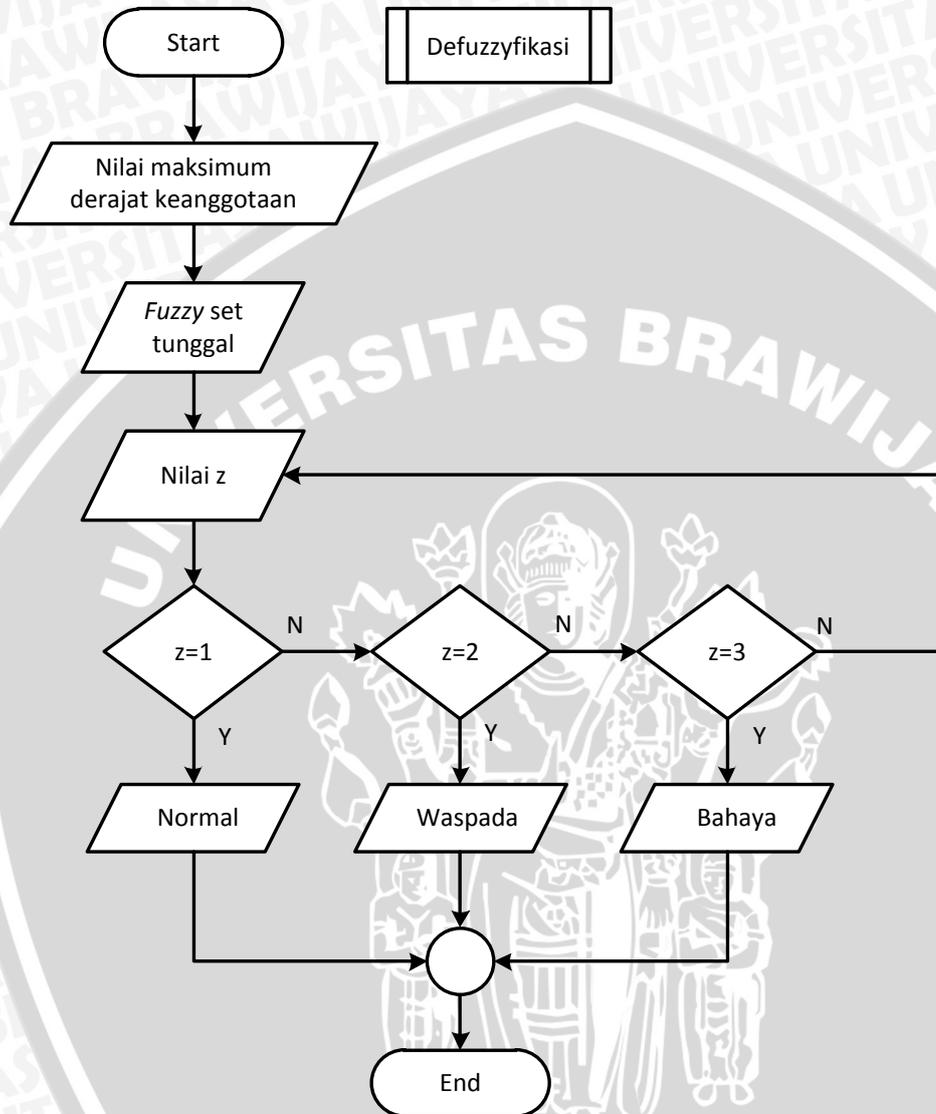
Kecepatan Air	Ketinggian Air			
	Rendah	Sedang	Tinggi	Sangat Tinggi
Pelan	Normal	Normal	Waspada	Bahaya
Sedang	Normal	Normal	Waspada	Bahaya
Cepat	Normal	Waspada	Bahaya	Bahaya

Tabel 5.6 merupakan tabel keterangan *rule fuzzy* yang telah dibuat yaitu:

1. Jika kecepatan air pelan dan ketinggian air rendah maka *outputnya* normal.
2. Jika kecepatan air pelan dan ketinggian air sedang maka *outputnya* normal.
3. Jika kecepatan air pelan dan ketinggian air tinggi maka *outputnya* waspada.
4. Jika kecepatan air pelan dan ketinggian air sangat tinggi maka *outputnya* bahaya.
5. Jika kecepatan air sedang dan ketinggian air rendah maka *outputnya* normal.
6. Jika kecepatan air sedang dan ketinggian air sedang maka *outputnya* normal.
7. Jika kecepatan air sedang dan ketinggian air tinggi maka *outputnya* waspada.
8. Jika kecepatan air sedang dan ketinggian air sangat tinggi maka *outputnya* bahaya.
9. Jika kecepatan air cepat dan ketinggian air rendah maka *outputnya* normal.
10. Jika kecepatan air cepat dan ketinggian air sedang maka *outputnya* waspada.
11. Jika kecepatan air cepat dan ketinggian air tinggi maka *outputnya* bahaya.
12. Jika kecepatan air cepat dan ketinggian air sangat tinggi maka *outputnya* bahaya.

5.1.4.3 Defuzzifikasi

Setelah proses inferensi dilakukan proses defuzzifikasi dengan diagram alir seperti yang ada Gambar 5.17.



Gambar 5.17 Diagram Alir Proses Defuzzifikasi

Gambar 5.17 diatas merupakan diagram alir proses defuzzifikasi yang dimulai dari menerima hasil akhir dari proses inferensi yaitu nilai maksimum derajat keanggotaan yang akan dijadikan *fuzzy set tunggal*. *Fuzzy set tunggal* akan menghasilkan hasil akhir yaitu nilai z . jika nilai z sama dengan satu ($z=1$) maka *output* yang dihasilkan adalah normal, jika nilai z sama dengan dua ($z=2$) maka *output* yang dihasilkan adalah waspada, jika nilai z sama dengan tiga ($z=3$) maka *output* yang dihasilkan adalah bahaya.

5.2 Implementasi Sistem

Implementasi sistem yang dibuat mengacu pada perancangan sistem yang sudah dilakukan sebelumnya, sehingga proses perancangan harus terpenuhi terlebih dahulu. Implementasi sistem yang dibuat terdiri dari implementasi pada alat, implementasi pada *node* sensor dan implementasi pada *node* pos jaga.

5.2.1 Implementasi *Prototype* Selokan

Implementasi *prototype* selokan disesuaikan dengan perancangan yang telah dilakukan sebelumnya yang dapat dilihat pada gambar 5.18.



Gambar 5.18 *Prototype* Selokan

Gambar 5.18 merupakan gambar *prototype* selokan yang dibuat menggunakan menggunakan pipa pvc kotak sebagai miniatur selokan, akrilik batangan sebagai penyangga sensor ultrasonik dan *bracket* sensor ultrasonik sebagai tempat meletakkan sensor ultrasonik.

5.2.2 Implementasi Perangkat Keras Pada *Node* Sensor

Implementasi perangkat keras pada *node* sensor disesuaikan dengan perancangan yang telah dilakukan sebelumnya yang dapat dilihat pada gambar 5.19.

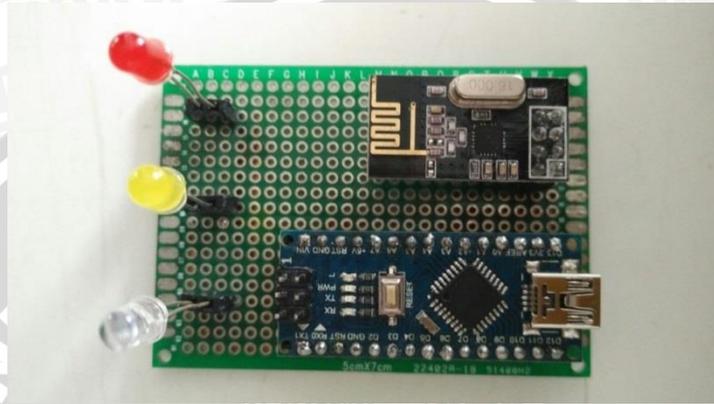


Gambar 5.19 *Node* Sensor

Gambar 5.19 merupakan gambar *node* sensor yang dibuat sesuai dengan perancangan yaitu terdiri dari mikrokontroler Arduino Nano, Sensor Ultrasonik, Sensor Kecepatan Air G1/2 dan NRF24L01.

5.2.3 Implementasi Perangkat Keras Pada *Node* Pos Jaga

Implementasi perangkat keras pada *node* pos jaga disesuaikan dengan perancangan yang telah dilakukan sebelumnya yang dapat dilihat pada gambar 5.20.



Gambar 5.20 *Node* Pos Jaga

Gambar 5.20 merupakan gambar *node* pos jaga yang dibuat sesuai dengan perancangan yaitu terdiri dari mikrokontroler arduin nano, LED dan NRF24L01.

5.2.4 Implementasi Perangkat Lunak Pada *Node* Sensor

Implementasi pada *node* sensor terdiri dari implementasi pada sensor ultrasonik, implementasi pada *node* sensor kecepatan air G1/2, implementasi pada NRF24L01 dan implementasi *Fuzzy Sugeno*.

5.2.4.1 Implementasi Pada Sensor Ultrasonik (PING)

Implementasi pada sensor ultrasonik (PING) dilakukan agar sensor dapat membaca data tinggi air pada selokan. Kode pemrograman pada sensor ultrasonik (PING) akan ditunjukkan Gambar 5.21.

```
#define TRIGGER_PIN
#define ECHO_PIN 3
#define MAX_DISTANCE 50
NewPing sonar(4, 3, 50);

void baca_ultra()
{
  unsigned int uS = sonar.ping();
  Serial.print("Ping: ");
  Serial.print(uS / US_ROUNDTRIP_CM);
  Serial.println("cm");
  num2 = 15 - (uS / US_ROUNDTRIP_CM);
  delay(50);
}
```

Gambar 5.21 Kode Program Baca Sensor Ultrasonik (PING)

Gambar 5.21 di atas merupakan kode program baca sensor ultrasonik (PING) yang dalam perancangannya dimulai dengan inialisasi pin yang digunakan dalam Arduino Nano yaitu pin D3 dan D4 dan menentukan jarak maksimalnya yaitu 50cm, kemudian memasukkan program agar dapat membaca data.

5.2.4.2 Implementasi Pada Sensor Kecepatan Air G1/2

implementasi pada sensor kecepatan air G1/2 dilakukan agar sensor dapat membaca data kecepatan air pada selokan. Kode pemrograman pada sensor kecepatan air akan ditunjukkan Gambar 5.22.

```
void baca_water()
{
  NbTopsFan = 0;
  sei();
  delay (1000);
  cli();
  Calc = (NbTopsFan * 60 / 7.5); Calc = num1;
  Serial.print (Calc, DEC);
  Serial.print (" L/hour\r\n");
}
```

Gambar 5.22 Kode Pemrograman Baca Data Sensor Kecepatan Air G1/2

Gambar 5.22 di atas merupakan kode pemrograman baca sensor kecepatan air G1/2 yang dalam perancangannya dimulai dengan inialisasi NbTopsFan yang merupakan frekuensi *pulse*, Calc untuk penghitungan dan pin yang digunakan dalam Arduino Nano yaitu pin D2, kemudian memasukkan program agar dapat membaca data.

5.2.4.3 Implementasi Fuzzy Sugeno

Pada implementasi ini menjelaskan tentang algoritma *Fuzzy Sugeno* yang digunakan untuk mengklasifikasikan kondisi sesuai dengan *input* yang diterima. Dalam metode *Fuzzy Sugeno* terdapat tiga tahap, yaitu proses *fuzzifikasi*, proses inferensi, dan proses defuzzifikasi. Dibawah ini merupakan tahap-tahap metode *Fuzzy Sugeno*:

1. Proses Fuzzifikasi

Gambar 5.23 dan Gambar 5.24 dibawah ini merupakan kode program *fuzzifikasi* dalam penentuan derajat keanggotaan tinggi selokan dan derajat keanggotaan kecepatan air pada selokan.

```
float keanggotaan_ultra (float m)
{
    if (m <= 1.25) {
        hk1[0] = 1;
        hk1[1] = 0;
        hk1[2] = 0;
        hk1[3] = 0;
    }
    else if (m > 1.25 && m < 3.75) {
        hk1[0] = ((3.76 - m) / (3.75 - 1.25));
        hk1[1] = ((m - 1.25) / (3.75 - 1.25));
        hk1[2] = 0;
        hk1[3] = 0;
    }
    else if (m >= 3.75 && m < 6.25) {
        hk1[0] = 0;
        hk1[1] = ((6.25 - m) / (6.25 - 3.75));
        hk1[2] = ((m - 5.0) / (6.25 - 3.75));
        hk1[3] = 0;
    }
    else if (m >= 6.25 && m < 8.75) {
        hk1[0] = 0;
        hk1[1] = 0;
        hk1[2] = ((8.75 - m) / (8.75 - 6.25));
        hk1[3] = ((m - 6.25) / (8.75 - 6.25));
    }
    else if (m >= 8.75 && m < 10) {
        hk1[0] = 0;
        hk1[1] = 0;
        hk1[2] = 0;
        hk1[3] = 1;
    }
    else if (m == 3.75) {
        hk1[0] = 0;
        hk1[1] = 1;
        hk1[2] = 0;
        hk1[3] = 0;
    }
    else if (m == 6.25) {
        hk1[0] = 0;
        hk1[1] = 0;
        hk1[2] = 1;
        hk1[3] = 0;
    }
    else if (m == 10) {
        hk1[0] = 0;
        hk1[1] = 0;
        hk1[2] = 0;
        hk1[3] = 1;
    }
}
```

Gambar 5.23 Kode Pemrograman Derajat Keanggotaan Ketinggian Air

Gambar 5.23 diatas merupakan penggalan kode pemrograman derajat keanggotaan ketinggian air. Dalam program diatas m adalah hasil baca ketinggian air yang menggunakan sensor ultrasoik yang kemudian dimasukkan ke dalam rumus fungsi keanggotaan menggunakan persamaan representasi kurva segitiga.

```
float keanggotaan_water (float m)
{
    if (m <= 24) {
        hk[0] = 1;
        hk[1] = 0;
        hk[2] = 0;
    }
    else if (m > 24 && m < 72) {
        hk[0] = ((72 - m) / (72 - 24));
        hk[1] = ((m - 24) / (72 - 24));
        hk[2] = 0;
    }
    else if (m >72 && m < 120) {
        hk[0] = 0;
        hk[1] = ((120 - m) / (120 - 72));
        hk[2] = ((m - 72) / (120 - 72));
    }
    else if (m == 72) {
        hk[0] = 0;
        hk[1] = 1;
        hk[2] = 0;
    }
    else if (m >= 120) {
        hk[0] = 0;
        hk[1] = 0;
        hk[2] = 1;
    }
}
```

Gambar 5.24 Kode Pemrograman Derajat Keanggotaan Kecepatan Air

Gambar 5.24 diatas merupakan penggalan kode pemrograman derajat keanggotaan kecepatan air. Dalam program diatas, sama dengan kode program perhitungan derajat keanggotaan m adalah hasil baca kecepatan air yang menggunakan sensor kecepatan air G1/2 yang kemudian dimasukkan ke dalam rumus fungsi keanggotaan menggunakan persamaan representasi kurva segitiga.

2. Proses Inferensi

Dibawah ini adalah Gambar 5.25 dan 5.26 yang merupakan kode program infrensi.

```
float rule1(float x1, float x2, float x3, float y1, float y2, float
y3, float y4){
    float z;
    // x=water y=ultra z=output rule fuzzy
    pred1 = min(x1, y1);
    pred2 = min(x1, y2);
    pred3 = min(x1, y3);
    pred4 = min(x1, y4);
    pred5 = min(x2, y1);
    pred6 = min(x2, y2);
    pred7 = min(x2, y3);
    pred8 = min(x2, y4);
    pred9 = min(x3, y1);
    pred10 = min(x3, y2);
    pred11 = min(x3, y3);
    pred12 = min(x3, y4);
}
```

Gambar 5.25 Kode Pemrograman Inferensi

Gambar 5.25 diatas merupakan kode program infrensi diamana x adalah fungsi keanggotaan dari ketinggian air, sedangkan y adalah fungsi keanggotaan dari kecepatan air dan pred (prediksi) adalah *rule fuzzy*. Dalam program tersebut menggunakan aturan *conjunction*(\wedge), sehingga *fuzzy input* dibandingkan dan akan dipilih derajat keanggotaan yang paling kecil.

```
int data;
float z;

const byte size = 15;
float rawArray[size] =
{pred1,pred2,pred3,pred4,pred5,pred6,pred7,pred8,pred9,pred10,pred11,pred
12};
Array<float> array = Array<float>(rawArray,size);

for (byte i=0; i<array.size(); i++){
}

float nmax = array.getMax();
```

Gambar 5.26 Kode Pemrograman Inferensi

Gambar 5.26 diatas merupakan kode program infrensi untuk membandingkan nilai pred berdasarkan aturan *disjunction*(\vee) untuk memilih derajat keanggotaan yang paling besar dan hasil dari perbandingan tersebut diberi nama nmax.

3. Proses Defuzzifikasi

Gambar 5.27 dibawah ini merupakan kode program defuzzifikasi.

```
if (nmax==pred1) {
  z=1;
} else {if (nmax==pred2) {
  z=1;
} else {if (nmax==pred3) {
  z=2;
} else {if (nmax==pred4) {
  z=3;
} else {if (nmax==pred5) {
  z=1;
} else {if (nmax==pred6) {
  z=1;
} else {if (nmax==pred7) {
  z=2;
} else {if (nmax==pred8) {
  z=3;
} else {if (nmax==pred9) {
  z=1;
} else {if (nmax==pred10) {
  z=1;
} else {if (nmax==pred11) {
  z=3;
} else {if (nmax==pred12) {
  z=3;
}}}}}}}}}}}}

Serial.println(z);

if (z == 1) {
  Serial.println("normal");
  data = 1;
}
if (z == 2) {
  Serial.println("waspada");
  data = 2;
}
if (z == 3) {
  Serial.println("bahaya");
  data = 3;
}
```

Gambar 5.27 Kode Pemrograman Defuzzifikasi

Gambar 5.27 diatas merupakan kode pemrograman defuzzifikasi dimana z adalah hasil akhir yang nantinya akan dikirim melalui NRF24L01. Nilai diperoleh dari hasil pred. Apabila z=1 maka menyatakan bahwa z adalah keadaan normal, apabila z=2 maka menyatakan bahwa z adalah keadaan waspada, dan apabila z=3 akan menyatakan bahwa z adalah keadaan bahaya.

5.2.4.4 Implementasi Pada NRF24L01

Implementasi pada NRF24L01 dilakukan agar *node* pada sensor dapat mengirimkan data secara nirkabel ke *node* yang ada pada pos jaga. Kode pemrograman pada NRF24L01 akan ditunjukkan Gambar 5.28.

```
Mirf.spi = &MirfHardwareSpi;
Mirf.init();
Mirf.setRADDR((byte*)"slave");
Mirf.payload = sizeof(datasend);
Mirf.channel = 123;
Mirf.config();
datasend = data;
Mirf.setTADDR((byte*)"mastr"); //Mengirimkan Ke clien
Mirf.send((byte*)&datasend); //mengirimkan data dalam bentuk byte
while (Mirf.isSending()) {
}
```

Gambar 5.28 Kode Pemrograman NRF24L01 Pada Node Sensor

Gambar 5.28 diatas merupakan gambar kode pemrograman NRF24L01 pada *node* sensor. Perancangan dimulai dengan inialisasi modul NRF24L01, inialisasi tipe data yang akan dikirim, kemudian konfigurasi program agar dapat mengirim data.

5.2.5 Imlementasi Perangkat Lunak Pada Node Pos Jaga

Implementasi pada *node* pos jaga terdiri dari implementasi pada sensor, implementasi pada NRF24L01 dan implementasi pada LED.

5.2.5.1 Implementasi Pada NRF24L01

Implementasi pada NRF24L01 dilakukan agar *node* pada pos jaga dapat menerima data secara nirkabel yang dikirim dari *node* sensor. Potongan kode pemrograman pada NRF24L01 akan ditunjukkan Gambar 5.29.

```
//const char payload_length = 3;
byte datareceive;
void setup() {
  Serial.begin(9600);
  Mirf.spi = &MirfHardwareSpi;
  Mirf.init();
  Mirf.setRADDR((byte*)"mastr");
  Mirf.payload = sizeof(datareceive);
  Mirf.channel = 123;
  Mirf.config();
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

void loop() {
  int data;
  if (Mirf.dataReady()) {
    Mirf.getData((byte*)&datareceive);
    data = datareceive;
    Serial.print("nilai : ");
    Serial.println(datareceive);
  }
}
```

Gambar 5.29 Kode Pemrograman NRF24L01 Pada Node Pos Jaga

Gambar 5.29 diatas merupakan gambar kode pemrograman NRF24L01 pada *node* pos jaga. Kode program dimulai dengan inialisasi modul NRF24L01, menyamakan *channel node* pos jaga dengan *channel* yang ada pada *node* sensor, kemudian memasukkan program agar dapat menerima data agar diolah oleh mikrokontroler Arduino Nano.

5.2.5.2 Implementasi Pada LED

Setelah membuat kode program pada NRF24L01, dibuat kode pemrograman pada LED agar dapat menyala sesuai dengan *input* yang diterima. Kode program pada LED akan ditunjukkan Gambar 5.30.

```
#include <SPI.h>
#include <Mirf.h>
#include <MirfHardwareSpiDriver.h>
#include <MirfSpiDriver.h>
#include <nRF24L01.h>
#define led1 3
#define led2 4
#define led3 5

//const char payload_length = 3;
byte datareceive;

void setup() {
  Serial.begin(9600);

  Mirf.spi = &MirfHardwareSpi;
  Mirf.init();

  Mirf.setRADDR((byte *)"mastr");
  Mirf.payload = sizeof(datareceive);

  Mirf.channel = 123;
  Mirf.config();
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
}

void loop() {
  int data;
  if (Mirf.dataReady()) {
    Mirf.getData((byte *)&datareceive);
    data = datareceive;
    Serial.print("nilai: ");
    Serial.println(datareceive);
  }
}
```

Gambar 5.30 Kode Pemrograman Pada LED

Gambar 5.30 diatas merupakan gambar kode pemrograman pada LED. Pada program diatas menjelaskan apabila data yang diterima ialah satu maka LED1 akan menyala dan LED yang lainnya mati, apabila data yang diterima ialah dua maka LED2 akan menyala dan yang lainnya mati, dan apabila data yang diterima adalah tiga maka LED yang ketiga akan menyala dan yang lainnya mati.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang pengujian dan juga analisis sistem pendeteksi luapan air menggunakan metode *fuzzy*. Pengujian meliputi pengujian alat monitoring luapan air pada selokan, kesesuaian metode *fuzzy*, dan pengiriman menggunakan NRF24L01.

6.1 Pengujian dan Analisis

Pengujian pada alat monitoring luapan air pada selokan ini bertujuan untuk mengetahui apakah alat dapat berjalan dengan baik sesuai dengan yang diharapkan yang terdiri dari pengujian sensor ultrasonik (PING), sensor kecepatan air, arduino, NRF24L01, dan *LED* dan pada sub bab analisis akan dilakukan pembahasan dari hasil pengujian yang telah dilakukan yang akan dijelaskan pada sub bab dibawah ini.

6.1.1 Pengujian Sensor Ultrasonik (PING)

Pengujian pada sensor ultrasonik (PING) ini dilakukan dengan tujuan untuk mengetahui apakah sensor ultrasonik (PING) dapat mendeteksi ketinggian air ketika dirancang pada alat monitoring luapan air pada selokan. Pengujian pada sensor ultrasonik (PING) dilakukan dengan cara mengukur tinggi manual menggunakan penggaris dan dibandingkan dengan menggunakan sensor ultrasonik (PING). Tabel pengujian dapat dilihat pada Tabel 6.1 dibawah ini.

Tabel 6.1 Hasil Pengujian Sensor Ultrasonik

Hasil Pengukuran Sensor	Hasil Pengukuran Manual
3 cm	3 cm
4 cm	4 cm
5 cm	5 cm
6 cm	6 cm
7 cm	7 cm
8 cm	8 cm
9 cm	9 cm
10 cm	10 cm
11 cm	11 cm
12 cm	12 cm
13 cm	13 cm
14 cm	14 cm
15 cm	15 cm

Tabel 6.1 diatas merupakan hasil pengujian sensor ultrasonik (PING) dimana akan ditunjukkan hasil pengukuran menggunakan sensor dan hasil pengukuran manual menggunakan penggaris.

6.1.1.1 Analisis Hasil Pengujian Sensor Ultrasonik (PING)

Seperti yang dapat dilihat pada hasil pengujian, sensor ultrasonik (PING) dapat bekerja dengan baik yaitu menghasilkan data ketinggian air pada selokan.

6.1.2 Pengujian Sensor Kecepatan Air

Pengujian pada sensor kecepatan air ini dikakukan dengan tujuan untuk mengetahui apakah sensor kecepatan air dapat mendeteksi kecepatan air pada selokan ketika dirancang pada alat monitoring luapan air pada selokan. Pengujian pada sensor kecepatan air dilakukan dengan cara mengukur kecepatan air pada *prototype* selokan sepanjang 100cm secara manual menggunakan bola ping-pong yang dihanyutkan kemudian menghitung waktu yang ditempuh dari titik awal hingga titik akhir. Setelah mendapatkan waktu tempuh digunakan rumus kecepatan = jarak/waktu ($v=s/t$). Dari kedua cara tersebut, akan didapatkan niali yang nantinya digunakan untuk menghitung keakurasiannya. Hasil pengujian tersebut dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Sensor Kecepatan Air

Hasil Pengukuran Sensor	Hasil Pengukuran Manual	Perhitungan Akurasi
8	8	100%
16	16	100%
24	23,8	99,16%
54	54	100%
40	40	100%
48	47,61	99,18%
72	71.94	99,91%
88	87.71	99,67%
104	103,62	99,63%
112	111,73	99,75%

Tabel 6.2 diatas merupakan hasil pengujian sensor kecepatan air dimana akan ditunjukkan hasil pengukuran menggunakan sensor dan hasil pengukuran manual menggunakan bola ping-pong dan stopwatch, dan penghitungan akurasi.

6.1.2.1 Analisis Hasil Pengujian Sensor Kecepatan Air

Seperti yang dapat dilihat pada hasil pengujian, sensor kecepatan air dapat bekerja dengan baik yaitu menghasilkan data berupa kecepatan arus air pada selokan dengan nilai akurasi sebesar 99. seperti yang ada dibawah ini 73%.

6.1.3 Pengujian Arduino Nano

Pengujian pada Arduino Nano ini dikakukan dengan tujuan untuk mengetahui apakah Arduino Nano dapat mengolah data ketika dirancang pada alat monitoring luapan air pada selokan. Tabel hasil uji Arduino Nano dapat dilihat pada Tabel 6.3 dibawah ini.

Tabel 6.3 Hasil Pengujian Arduino Nano

Kecepatan	Ketinggian	Output	Data yang diterima
8	4	Normal	1
32	4	Normal	1
8	5	Normal	1
32	5	Normal	1
8	7	Waspada	2
32	7	Waspada	2
8	8	Bahaya	3
32	8	Bahaya	3
56	4	Normal	1
64	4	Normal	1

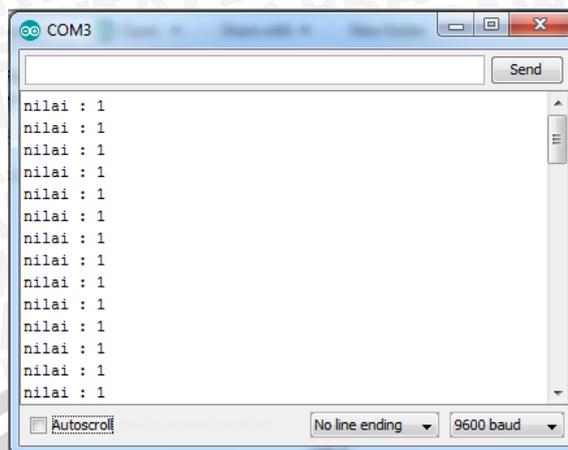
Tabel 6.3 diatas merupakan gambar hasil pengujian arduino dimana akan ditunjukkan bahwa arduino dapat menerima data dari sensor, menghitung derajat keanggotaan, dan mendapatkan data.

6.1.3.1 Analisis Hasil Pengujian Arduino Nano

Seperti yang dapat dilihat pada hasil pengujian, Arduino Nano dapat bekerja dengan baik yaitu dapat menerima data dari sensor ultrasonik (PING) dan sensor kecepatan air, mengolahnya agar dengan metode *fuzzy*, hingga dapat menghasilkan data.

6.1.4 Pengujian NRF24L01

Pengujian pada NRF24L01 ini dikakukan dengan tujuan untuk mengetahui apakah NRF24L01 dapat mengirim dan menerima data ketika dirancang pada alat monitoring luapan air pada selokan yang dapat dilihat pada Gambar 6.2 dibawah ini.



Gambar 6.1 Hasil Pengujian NRF24L01

Gambar 6.1 diatas merupakan gambar hasil pengujian NRF24L01 dimana akan ditunjukkan bahwa NRF24L01 dapat menerima data yaitu nilai sebesar 1 sehingga dapat disimpulkan juga bahwa NRF24L01 pada *node* sensor dapat mengirim data.

6.1.4.1 Analisis Hasil Pengujian NRF24L01

Seperti yang dapat dilihat pada hasil pengujian, NRF24L01 bekerja dengan baik yaitu pada *node* sensor dapat digunakan sebagai penerima dan pengirim data (*transceiver*) dan pada *node* pos jaga dapat menerima data yang dikirim oleh *node* sensor.

6.1.5 Pengujian LED

Pengujian pada LED ini dilakukan dengan tujuan untuk mengetahui apakah LED dapat menampilkan *output* ketika dirancang pada alat monitoring luapan air pada selokan yang dapat dilihat pada Gambar 6.2 dibawah ini.



Gambar 6.2 Hasil Pengujian LED

Gambar 6.2 diatas merupakan gambar hasil pengujian LED dimana akan ditunjukkan bahwa ketiga LED yaitu merah, kuning, dan hijau dapat menyala.

6.1.5.1 Analisis Hasil Pengujian LED

Seperti yang dapat dilihat pada hasil pengujian, LED dapat bekerja dengan baik yaitu menyala sesuai data yang diterima NRF24L01 yaitu hijau bila data sama dengan satu, kuning bila data sama dengan dua, dan merah bila data sama dengan tiga.

6.1.6 Pengujian Fungsional Logika Fuzzy

Pengujian pada penelitian ini membahas tentang pengujian dari penerapan logika *fuzzy* pada sistem monitoring luapan air pada selokan yang telah dibuat. Pengujian ini dilakukan dengan tujuan untuk menguji kinerja sistem setelah melakukan implementasi dengan menerapkan logika *fuzzy* menggunakan parameter berupa data sensor kecepatan air dan data dari sensor ultrasonik dengan beberapa skenario pengujian yang digunakan sehingga berdasarkan pengujian sistem yang telah dilakukan dapat diambil kesimpulan. Pengujian fungsional logika *fuzzy* dilakukan agar dapat mengetahui apakah logika *fuzzy* yang digunakan memberikan kesesuaian antara *output* pada percobaan dengan *output* yang diperoleh sistem.

Pengujian fungsionalitas merupakan pengujian yang dilakukan untuk mengetahui apakah perangkat lunak telah memenuhi kebutuhan fungsional, seperti “ketika sensor ultrasonik mendeteksi ketinggian dimana dalam klasifikasi logika *fuzzy* termasuk kategori rendah, dan sensor kecepatan air mendeteksi kecepatan dan termasuk kategori pelan, maka program harus memberikan *output* berupa normal”. Hasil dari pengujian tersebut berupa “sesuai” atau “tidak sesuai”.

Metode *fuzzy* yang digunakan pada alat ini adalah metode *Fuzzy Sugeno*. Pada proses implikasi menggunakan metode MAX sehingga operator yang digunakan adalah operator AND yang merupakan pengambilan nilai berdasarkan nilai keanggotaan terkecil antar himpunan yang bersangkutan, pada proses komposisi menggunakan metode MIN sehingga operator yang digunakan adalah operator OR yang merupakan pengambilan nilai berdasarkan nilai keanggotaan terkecil antar himpunan yang bersangkutan, dan α -predikat merupakan hasil dari operasi. Dalam pengujian ini nilai *input* yang diberikan adalah:

- Kecepatan air = 40 L/hour
- Ketinggian air = 8cm

Pertama-tama yang harus dilakukan dalam menentukan *output* adalah menghitung derajat keanggotaan masing-masing himpunan keanggotaan. Dalam menghitung derajat keanggotaan digunakan persamaan (6.1) dibawah ini.

$$\mu(x) = \begin{cases} 0 & : x \leq a \text{ dan } x \geq c \\ \frac{(x-a)}{(b-a)} & : a < x \leq b \\ \frac{(c-x)}{(c-b)} & : b < x < c \end{cases} \quad (6.1)$$

Pada persamaan (6.1) a merupakan nilai domain dengan derajat keanggotaan 0, sedangkan b adalah nilai domain dengan derajat keanggotaan tertinggi yaitu 1, c adalah domain setelah b dengan derajat keanggotaan 0, dan x adalah *input* yang dimasukkan.

Mencari nilai keanggotaan variabel kecepatan :

Kecepatan Pelan [40] : $(72-40)/(72-24) = 0.66$
 Kecepatan Sedang [40] : $(40-24)/(72-24) = 0.33$
 Kecepatan Cepat [40] : 0

Mencari nilai keanggotaan variabel ketinggian :

Ketinggian Rendah [8] : 0
 Ketinggian Sedang [8] : 0
 Ketinggian Tinggi [8] : $(8,75-8) / (8.75-6,25) = 0.3$
 Ketinggian Sangat Tinggi [8] : $(8-6,25) / (8.75-6,25) = 0.7$

Setelah diperoleh nilai keanggotaan dari kecepatan dan ketinggian, maka dilakukan proses komposisi dengan metode MIN seperti yang ada dibawah ini:

[Rule1] IF (kecepatan pelan) AND (ketinggian rendah) THEN *output* = normal
 $\alpha\text{-pred1} = \min(0, 0) = 0$

[Rule2] IF (kecepatan pelan) AND (ketinggian sedang) THEN *output* = normal
 $\alpha\text{-pred2} = \min(0, 0) = 0$

[Rule3] IF (kecepatan pelan) AND (ketinggian tinggi) THEN *output* = waspada
 $\alpha\text{-pred3} = \min(0.3, 0.4) = 0.3$

[Rule4] IF (kecepatan pelan) AND (ketinggian sangat tinggi) THEN *output* = bahaya
 $\alpha\text{-pred4} = \min(0.3, 0.4) = 0.3$

[Rule5] IF (kecepatan sedang) AND (ketinggian pelan) THEN *output* = normal
 $\alpha\text{-pred5} = \min(0, 0) = 0$

[Rule6] IF (kecepatan sedang) AND (ketinggian sedang) THEN *output* = normal
 $\alpha\text{-pred6} = \min(0, 0) = 0$

[Rule7] IF (kecepatan sedang) AND (ketinggian tinggi) THEN *output* = waspada
 $\alpha\text{-pred7} = \min(0.7, 0.66) = 0.66$

[Rule8] IF (kecepatan sedang) AND (ketinggian sangat tinggi) THEN *output* =
 bahaya
 $\alpha\text{-pred8} = \min(0.7, 0.33) = 0.33$

[Rule9] IF (kecepatan cepat) AND (ketinggian rendah) THEN *output* = normal
 $\alpha\text{-pred9} = \min(0, 0) = 0$

[Rule10] IF (kecepatan cepat) AND (ketinggian sedang) THEN *output* = normal
 $\alpha\text{-pred10} = \min(0, 0) = 0$

[Rule11] IF (kecepatan cepat) AND (ketinggian tinggi) THEN *output* = bahaya
 $\alpha\text{-pred11} = \min(0, 0) = 0$

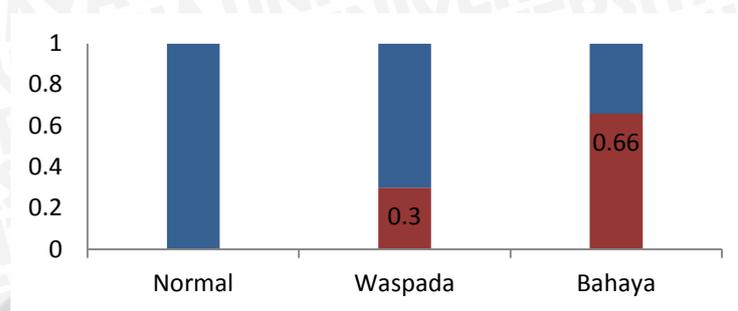
[Rule12] IF (kecepatan cepat) AND (ketinggian sangayt tinggi) THEN *output* =
 bahaya
 $\alpha\text{-pred12} = \min(0, 0) = 0$

Setelah didapatkan 4 nilai dari proses komposisi, yaitu waspada dengan nilai 0.3, waspada dengan nilai 0.3, bahaya dengan nilai 0.67, dan bahaya dengan nilai 0.34, maka dilakukan proses implikasi dengan metode MAX seperti yang ada dibawah ini:

IF (waspada 0.3) OR (waspada 0,3) THEN max = waspada 0,3

IF (bahaya 0.67) OR (bahaya 0.34) THEN max = bahaya 0.67

Dengan demikian diperoleh dua pernyataan yaitu waspada dengan nilai 0,3 dan bahaya dengan nilai 0.67. Setelah itu dilakukan proses defuzzifikasi yang dapat dilihat pada Gambar 6.4 dibawah ini:



Gambar 6.4 Diagram Defuzzifikasi

Gambar 6.4 merupakan gambar diagram defuzzifikasi di mana dari gambar tersebut akan ditunjukkan bahwa *output* yang dihasilkan adalah bahaya karena dalam proses defuzzifikasi menggunakan *height method* yang berarti mengambil nilai terbesar sebagai *output*.

Dengan cara yang sama seperti diatas dilakukan pengujian fungsional sebanyak 24 kali, menggunakan skenario pengujian sesuai dengan *rule* yang ada dengan *input* berupa data sensor kecepatan air dan data sensor ultrasonik, sehingga menghasilkan Tabel 6.4 dibawah ini.

Tabel 6.4 Hasil Pengujian Fungsional Logika Fuzzy

Kasus Uji Pada Rule?	Nilai Manual			Skenario Pengujian	Hasil Sistem	Kesesuaian
	Kecepatan	Ketinggian	Output			
1	8	4	Normal	Jika kecepatan pelan, dan ketinggian rendah akan menghasilkan <i>output</i> normal	Normal	Sesuai
1	32	4	Normal	Jika kecepatan pelan, dan ketinggian rendah akan menghasilkan <i>output</i> normal	Normal	Sesuai
2	8	5	Normal	Jika kecepatan pelan, dan ketinggian sedang akan menghasilkan <i>output</i> normal	Normal	Sesuai

2	32	5	Normal	Jika kecepatan pelan, dan ketinggian sedang akan menghasilkan <i>output</i> normal	Normal	Sesuai
3	8	7	Waspada	Jika kecepatan pelan, dan ketinggian tinggi akan menghasilkan <i>output</i> waspada	Waspada	Sesuai
3	32	7	Waspada	Jika kecepatan pelan, dan ketinggian tinggi akan menghasilkan <i>output</i> waspada	Waspada	Sesuai
4	8	8	Bahaya	Jika kecepatan pelan, dan ketinggian sangat tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
4	32	8	Bahaya	Jika kecepatan pelan, dan ketinggian sangat tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
5	56	4	Normal	Jika kecepatan sedang, dan ketinggian rendah akan menghasilkan <i>output</i> normal	Normal	Sesuai
5	64	4	Normal	Jika kecepatan sedang, dan ketinggian rendah akan menghasilkan <i>output</i> normal	Normal	Sesuai
6	56	5	Normal	Jika kecepatan sedang, dan ketinggian sedang akan menghasilkan <i>output</i> normal	Normal	Sesuai

6	64	5	Normal	Jika kecepatan sedang, dan ketinggian sedang akan menghasilkan <i>output</i> normal	Normal	Sesuai
7	56	7	Waspada	Jika kecepatan sedang, dan ketinggian tinggi akan menghasilkan <i>output</i> waspada	Waspada	Sesuai
7	64	7	Waspada	Jika kecepatan sedang, dan ketinggian tinggi akan menghasilkan <i>output</i> waspada	Waspada	Sesuai
8	56	8	Bahaya	Jika kecepatan sedang, dan ketinggian sangat tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
8	64	8	Bahaya	Jika kecepatan sedang, dan ketinggian sangat tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
9	120	4	Normal	Jika kecepatan cepat, dan ketinggian rendah akan menghasilkan <i>output</i> normal	Normal	Sesuai
9	144	4	Normal	Jika kecepatan cepat, dan ketinggian rendah akan menghasilkan <i>output</i> normal	Normal	Sesuai
10	120	5	Normal	Jika kecepatan cepat, dan ketinggian sedang akan menghasilkan <i>output</i> normal	Normal	Sesuai

10	144	5	Normal	Jika kecepatan cepat, dan ketinggian sedang akan menghasilkan <i>output</i> normal	Normal	Sesuai
11	120	7	Bahaya	Jika kecepatan cepat, dan ketinggian tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
11	144	7	Bahaya	Jika kecepatan cepat, dan ketinggian tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
12	120	8	Bahaya	Jika kecepatan cepat, dan ketinggian sangat tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai
12	144	8	Bahaya	Jika kecepatan cepat, dan ketinggian sangat tinggi akan menghasilkan <i>output</i> bahaya	Bahaya	Sesuai

6.1.6.1 Analisis Hasil Pengujian Fungsional Logika Fuzzy

Seperti yang dapat dilihat pada hasil pengujian, akan ditunjukkan penghitungan *fuzzy* yang dihasilkan berdasarkan *rule fuzzy* menghasilkan *output* yang sama dengan program *fuzzy* yang telah dibuat pada alat. Dengan demikian program *fuzzy* telah bekerja sesuai dengan perhitungan manual yang dilakukan dengan kesesuaian 100%.

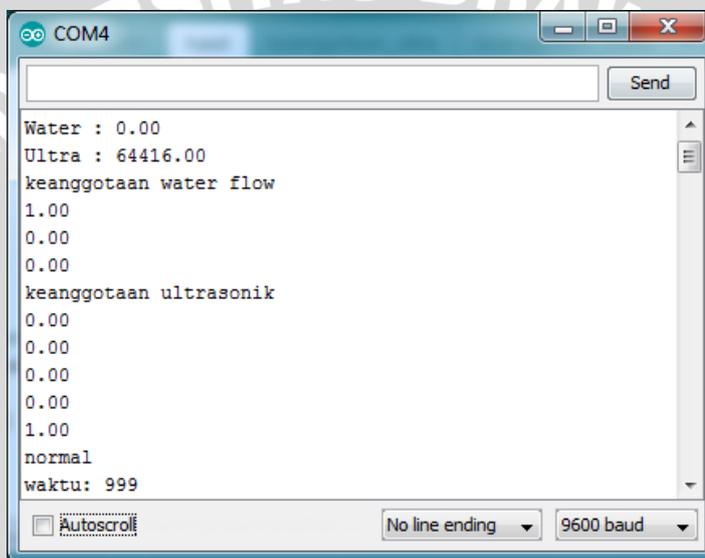
6.1.7 Pengujian Waktu Pemrosesan

Pengujian waktu pemrosesan dilakukan agar dapat mengetahui berapa lama waktu yang dibutuhkan oleh alat dari awal mendeteksi kecepatan dan ketinggian hingga mengirimkan data menggunakan NRF24L01. Untuk menghitung durasinya digunakan program seperti yang ada pada Gambar 6.5 dibawah ini.

```
Serial.print("waktu: ");
Serial.println((millis() - durasi));
delay(800);
```

Gambar 6.5 Program Menghitung Waktu Pemrosesan

Gambar 6.5 diatas merupakan gambar program menghitung waktu pemrosesan menggunakan millis(). Millis() digunakan untuk menghitung waktu dalam *milisecond* sejak arduino mulai menjalankan program, sehingga setelah inialisasi variabel durasi, nilai millis() dimasukkan ke dalam variabel durasi yang letaknya di awal program dimulai, dan kemudian nilai millis() dikurangi nilai durasi seperti gambar diatas. Hasil dari program diatas akan ditunjukkan pada gambar 6.6 dibawah ini.



Gambar 6.6 Waktu Pemrosesan

Gambar 6.6 diatas merupakan waktu pemrosesan yang ditampilkan di *serial monitor* Arduino IDE, dimana akan ditunjukkan bahwa dalam gambar tersebut waktu sama dengan 999 yang berarti waktu pemrosesan dari awal mendapatkan data kecepatan dan ketinggian hingga pengiriman data menggunakan NRF24L01 adalah 999 *millisecond*.

Dengan cara yang sama seperti diatas dilakukan pengujian sebanyak 15 kali, sehingga menghasilkan Tabel 6.5 dibawah ini.

Tabel 6.5 Hasil Pengujian Waktu Pemrosesan

No	Pengujian	Durasi
1	Ke-1	999 ms
2	Ke-2	1001 ms
3	Ke-3	1002 ms
4	Ke-4	1001 ms
5	Ke-5	1002 ms
6	Ke-6	1001 ms
7	Ke-7	1002 ms
8	Ke-8	1001 ms
9	Ke-9	1002 ms
10	Ke-10	1001 ms
11	Ke-11	1002 ms
12	Ke-12	1001 ms
13	Ke-13	1002 ms
14	Ke-14	1001 ms
15	Ke-15	1002 ms

6.1.7.1 Analisis Hasil Pengujian Waktu Pemrosesan

Seperti yang dapat dilihat pada hasil pengujian, akan ditunjukkan waktu pemrosesan dari 15 kali perulangan proses dari awal hingga mengirim data rata-ratanya adalah 1001,33 *milisecond*.

BAB 7 PENUTUPAN

7.1 Kesimpulan

Berdasarkan yang ada pada rumusan masalah, perancangan, implementasi, dan pengujian yang telah dilakukan sebelumnya, dapat diambil kesimpulan sebagai berikut:

1. Pada penelitian ini telah dibuat alat pendeteksi luapan pada air yang diaplikasikan pada selokan menggunakan metode *Fuzzy Sugeno* dalam pengambilan keputusan dimana semua komponen yang ada pada alat dapat bekerja sesuai dengan yang diharapkan.
2. Pengujian fungsional logika *fuzzy* yang dilakukan berdasarkan *rule* yang ada dengan membandingkan hasil dari perhitungan *fuzzy* yang dilakukan secara manual dengan hasil dari program *fuzzy* yang telah dibuat. Dari hasil pengujian akan ditunjukkan bahwa fungsionalitas logika *fuzzy* yang telah dibuat dalam 100% sesuai dengan yang diharapkan.
3. Berdasarkan hasil pengujian yang dilakukan, dalam satu kali pemrosesan waktu rata-rata yang dibutuhkan hingga mendapatkan *output* adalah 1001,33 *milisecond*.

7.2 Saran

Adapun beberapa saran yang dapat dilakukan dalam mengembangkan sistem monitoring luapan air pada selokan ini yaitu:

1. Sebagai pengembangan lebih lanjut, diharapkan *output* dapat ditampilkan melalui aplikasi agar dapat dilakukan pengecekan pada selokan sewaktu-waktu dimanapun berada.
2. Menambahkan fitur pada alat untuk dapat menginformasikan apakah selokan sedang dalam keadaan tersumbat atau meluap.
3. Menambahkan notifikasi yang langsung dapat diterima pengguna, seperti *buzzer* atau *sms gateway* setiap selokan dalam keadaan waspada maupun bahaya.

DAFTAR PUSTAKA

- Mulyana Eka & Kharisman Rindi . 2014 . *Perancangan Alat Peringatan Dini Bahaya Banjir dengan Mikrokontroler Arduino Nano R3* . STMIK. Tasikmalaya
- Prihantoro, Tegar Bhakti & Husni, Rizky Charli Wijaya . 2010. *Alat Pendeteksi Tinggi Permukaan Air Secara Otomatis Pada Bak Penampungan Air Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler*. AMIK GI MDP
- Li Yan, Li Manchun, 2011. Application and Research on Flood Risk Assessment Decision Support Sistem in the Lower Yellow River. IEEE 978-1-61284-848-8/11.
- Bakornas, 2007. Pedoman Penanggulangan Bencana Banjir.
"Arduino," [Online]. Available: <http://www.arduino.cc/>. [Diakses 10 Februari 2016]
- P. Parallax, "Parallax Inc.," [Online]. Available: <https://www.parallax.com/sites/default/files/downloads/28015-PING-Sensor-Product-Guide-v2.0.pdf>. [Diakses 10 Februari 2016].
- Arduino, "Water Flow Sensor," 2013. [Online]. Available: <http://forum.arduino.cc/index.php/topic,8548.0.html>. [Diakses 10 Februari 2016].
- "Seeedstudio," [Online]. Available: <http://www.seeedstudio.com/depot/datasheet/water%20flow%20sensor%20datasheet.pdf>. [Diakses 10 Februari 2016].
- Digiware, "NRF24L01 Module" [Online]. Available: <http://digiwarestore.com/en/transceiver/NRF24L01-module-432169.html>. [diakses 12 Februari 2016]