

***PROTOTYPE MONITORING KETINGGIAN AIR BERBASIS
INTERNET OF THINGS MENGGUNAKAN BLYNK DAN
NODEMCU ESP8266 PADA TANGKI***

SKRIPSI

Karya Tulis sebagai syarat memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung

Disusun oleh:

ANGGA TRIATNA
NIM. C1A150017



PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG
BANDUNG

2019

LEMBAR PERSETUJUAN PEMBIMBING

*PROTOTYPE MONITORING KETINGGIAN AIR BERBASIS
INTERNET OF THINGS MENGGUNAKAN BLYNK DAN
NODEMCU ESP8266 PADA TANGKI*

Disusun oleh:

ANGGA TRIATNA

NIM. C1A150017

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Pembimbing 1

Pembimbing 2

Yudi Herdiana, S.T., M.T
NIDN. 0428027501

Zen Munawar, S.T., M.Kom
NIDN. 0422037002

LEMBAR PERSETUJUAN PENGUJI

*PROTOTYPE MONITORING KETINGGIAN AIR BERBASIS
INTERNET OF THINGS MENGGUNAKAN BLYNK DAN
NODEMCU ESP8266 PADA TANGKI*

Disusun oleh:

ANGGA TRIATNA

NIM. C1A150017

Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Disetujui oleh:

Penguji 1

Penguji 2

Denny Rusdianto, S.T., M.Kom.
NIK. 04104808094

Yaya Suharya, S.Kom., M.T.
NIK. 01043170007

LEMBAR PENGESAHAN PROGRAM STUDI

*PROTOTYPE MONITORING KETINGGIAN AIR BERBASIS
INTERNET OF THINGS MENGGUNAKAN BLYNK DAN
NODEMCU ESP8266 PADA TANGKI*

Disusun oleh:

ANGGA TRIATNA

NIM.C1A150017

SKRIPSI ini telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2019

Mengetahui,
Dekan

Mengesahkan,
Ketua Program Studi

Yudi Herdiana, S.T., M.T
NIK. 04104808008

Yaya Suharya, S.Kom., M.T
NIK. 01043170007

LEMBAR PERNYATAAN KEASLIAN

Saya yang bertanda tangan dibawah ini:

Nama : ANGGA TRIATNA

NIM : C1A150017

Judul Skripsi : ***PROTOTYPE MONITORING KETINGGIAN AIR BERBASIS INTERNET OF THINGS MENGGUNAKAN BLYNK DAN NODEMCU ESP8266 PADA TANGKI***

Menyatakan dengan sebenarnya bahwa penulisan skripsi ini berdasarkan hasil penelitian, pemikiran dan pemaparan asli dari saya sendiri, baik untuk naskah laporan maupun kegiatan *programming* yang tercantum sebagai bagian dari skripsi ini. Jika terdapat karya orang lain, saya mencantumkan sumber yang jelas.

Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini dan sanksi lain sesuai dengan peraturan yang berlaku di FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS BALE BANDUNG.

Demikian surat pernyataan ini saya buat dalam keadaan sadar tanpa paksaan dari pihak manapun.

Baleendah, Agustus 2019

Yang membuat pernyataan,

ANGGA TRIATNA

NIM. C1A150017

ABSTRACT

The water level is measuring water from the bottom to the surface. The water level in the water reservoir is measured by using a tool that is inserted into the reservoir so that it will get a water level value.

Many are still measuring or monitoring water levels in containers that use conventional or traditional methods where this method is carried out by several people alternately, although this conventional method is easy to do and does not require a lot of costs but will require a lot of human resources. to check also will spend a lot of time, besides checking with conventional methods will be constrained if the weather is raining, it is not possible to check the water level especially the condition of the water reservoir is at a high place. To overcome this, using an internet of things technology that will be very helpful. This technology uses a NodeMCU microcontroller board and ping sensor (ultrasonic) as a proximity sensor that will be connected to Android via the Blynk application.

The Method used is experiment that is tailored to the needs. Starting from observation is by testing each of the modules used, followed by interviews as validation of the literature review. Next is hardware and software development, implementation and testing. This study aims to make an internet of things water level meter.

Keywords: *Internet of things, Water Level, NodeMCU, Ultrasonic, Monitoring, experiment.*

ABSTRAK

Ketinggian air yaitu mengukur air dari dasar sampai permukaan. Ketinggian air pada penampung air diukur dengan cara menggunakan alat yang di masukan kedalam penampung tersebut sehingga akan mendapatkan nilai ketinggian air. Banyak yang masih melakukan pengukuran atau *monitoring* ketinggian air pada penampung yang menggunakan cara-cara konvensional atau tradisional dimana cara ini dilakukan oleh beberapa orang secara bergantian, meskipun cara konvensional ini mudah dilakukan dan tidak memerlukan banyak biaya namun akan membutuhkan banyak SDM (Sumber Daya Manusia) untuk melakukan pengecekan juga akan menghabiskan banyak waktu, selain itu pengecekan dengan cara konvensional akan terkendala jika cuaca sedang terjadi hujan, sangat tidak memungkinkan melakukan pengecekan ketinggian air apalagi kondisi penampungan air berada pada tempat yang tinggi.

Untuk menanggulangnya maka menggunakan sebuah teknologi *internet of things* yang akan sangat membantu. Teknologi ini menggunakan sebuah *board* mikrokontroler NodeMCU dan sensor ping (ultrasonik) sebagai sensor jarak yang akan terhubung ke android melalui aplikasi Blynk.

Metode yang digunakan merupakan metode percobaan atau eksperimen yang disesuaikan dengan kebutuhan, mulai dari observasi yaitu dengan melakukan uji coba terhadap setiap modul yang akan digunakan, dilanjutkan dengan wawancara sebagai validasi terhadap studi pustaka, selanjutnya adalah pengembangan *hardware* dan *software*, implementasi dan terakhir adalah pengujian. Penelitian ini bertujuan untuk membuat sebuah alat pengukur ketinggian air berbasis *Internet of Things*.

Kata Kunci: *Internet of things*, Ketinggian Air, NodeMCU, Ultrasonik, *Monitoring*, Eksperimen.

KATA PENGANTAR

Puji dan syukur penulis panjatkan atas kehadiran Allah SWT yang mana berkat Rahmat dan Karunia-NYA penulis dapat menyelesaikan Laporan Penelitian ini dengan judul ***“PROTOTYPE MONITORING KETINGGIAN AIR BERBASIS INTERNET OF THINGS MENGGUNAKAN BLYNK DAN NODEMCU ESP8266 PADA TANGKI”*** dengan baik dan tepat pada waktunya. Penelitian ini disusun untuk menyelesaikan tugas akhir guna memperoleh gelar Sarjana Komputer.

Laporan ini dibuat dengan berbagai observasi, wawancara dan beberapa bantuan dari berbagai pihak yang membantu menyelesaikan proses pengerjaan Penelitian ini. Oleh karena itu, penulis ucapkan terima kasih sebesar-besarnya kepada:

1. Yudi Herdiana, S.T., M.T, selaku Dekan dan Dosen Pembimbing Satu di Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Yaya Suharya, S.Kom., M.T, selaku Ketua Prodi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
3. Zen Munawar, S.T., M.Kom., selaku Dosen Pembimbing Dua di Prodi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung.
4. Keluarga yang memberikan dukungan secara moril dan materil serta do'a yang tak pernah putus dalam proses pengerjaan laporan.
5. Erlin Novita Sari yang memberikan dukungan serta semangat dalam proses pengerjaan laporan penelitian.
6. Rekan-rekan seperjuangan yang saling membantu dalam proses pengerjaan laporan penelitian.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran dari semua pihak yang bersifat membangun selalu penulis harapkan demi kesempurnaan laporan ini.

Bandung, Agustus 2019

Penulis

DAFTAR ISI

<i>ABSTRACT</i>	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Batasan Masalah	3
1.3 Rumusan Masalah	3
1.4 Tujuan.....	3
1.5 Metodologi Penelitian	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA.....	6
2.1 Landasan Teori	6
2.2 Dasar Teori	8
BAB III METODOLOGI PENELITIAN.....	32
3.1 Kerangka Pikir.....	32
3.2 Penjelasan Skema Penelitian	33
3.2.1 Pengumpulan Data.....	33
3.2.2 Pengembangan Sistem	33
3.2.3 Implementasi.....	38
3.2.4 Pembuatan Laporan	39
BAB IV ANALISIS DAN PERANCANGAN	41
4.1 Analisis	41
4.1.1 Instrumen Penelitian	41
4.1.2 Analisis Sistem	42
4.1.3 Analisis Kebutuhan.....	44
4.1.4 <i>User Interface</i>	50

4.1.6	Fitur-fitur	51
4.1.7	Hasil Analisis.....	52
4.2	Perancangan.....	52
4.2.1	Diagram Arus Data (DAD) – UML.....	52
4.2.2	Perancangan Perangkat Keras (<i>Hardware</i>)	59
4.2.3	Perancangan Perangkat Lunak (<i>Software</i>).....	72
BAB V IMPLEMENTASI DAN PENGUJIAN		73
5.1	Implementasi	73
5.1.1	Implementasi Arduino IDE.....	73
5.1.2	Implementasi Modul Elektronika	79
5.2	Pengujian	86
5.2.1	Pengujian Modul Elektronika	86
5.2.2	Kesimpulan Hasil Pengujian.....	102
BAB VI KESIMPULAN DAN SARAN		103
6.1	Kesimpulan.....	103
6.2	Saran	103
DAFTAR PUSTAKA		105
LAMPIRAN.....		107

DAFTAR TABEL

Tabel 2.1 Tabel Konfigurasi Pin NodeMCU ESP8266 v3	14
Tabel 2.2 Fungsi Pin Pada Sensor Ultrasonik	18
Tabel 2.3 Pin Pada LCD 16x2.....	21
Tabel 2.4 Simbol dan Keterangan Flowmap	29
Tabel 4.1 Instrumen Penelitian Perangkat Keras <i>Monitoring</i> Ketinggian Air.....	42
Tabel 4.2 Kebutuhan Internet.....	44
Tabel 4.3 Kebutuhan Alat dan Bahan	49
Tabel 4.4 Tabel Deskripsi Aktor	53
Tabel 4.5 Deskripsi <i>Usecase Monitoring</i> Ketinggian Air.....	54
Tabel 5.1 Pengujian IP Address pada board NodeMCU	87
Tabel 5.2 Pengujian Sensor Ultrasonik	89
Tabel 5.3 Pengujian LCD.....	90
Tabel 5.4 Pengujian LCD dan Sensor Ultrasonik	93
Tabel 5.5 Pengujian Mengukur Jarak Pada Virtual LCD	97
Tabel 5.6 Pengujian Keseluruhan Alat.....	98
Tabel 5.7 Pengujian Pengukuran Ketinggian Air	102

DAFTAR GAMBAR

Gambar 2.1 Tampilan <i>interface</i> dan <i>Widget Box</i> aplikasi Blynk.....	12
Gambar 2.2 Board NodeMCU ESP8266 v3 Lolin.....	13
Gambar 2.3 Arduino IDE (Sumber: Arduino, inc)	17
Gambar 2.4 Konfigurasi PIN Sensor Ultrasonik	18
Gambar 2.5 Cara Kerja Sensor Ultrasonik.....	19
Gambar 2.6 <i>Relay Single Channel 5V</i>	21
Gambar 2.7 LCD (<i>Liquid Crystal Display</i>) 16x2	22
Gambar 2.8 Gambar Breadboard	23
Gambar 2.9 Notasi Pada <i>Usecase Diagram</i>	25
Gambar 2.10 Simbol-simbol pada <i>Activity Diagram</i>	26
Gambar 2.11 Simbol-simbol dalam <i>flowchart</i>	27
Gambar 3.1 Kerangka Pikir.....	32
Gambar 3.2 Alur Kerja Sistem.....	36
Gambar 3.3 Diagram Blok Alur <i>Hardware</i>	38
Gambar 4.1 <i>Flowmap</i> Sistem Monitoring Konvensional	44
Gambar 4.2 <i>User Interface</i> Aplikasi Blynk	51
Gambar 4.3 <i>Usecase Diagram</i> Monitoring Ketinggian Air.....	53
Gambar 4.4 <i>Acivity</i> Melakukan SetUp.....	56
Gambar 4.5 <i>Activity</i> Smartphone Android dan Blynk	57
Gambar 4.6 <i>Activity</i> Menyalakan Buzzer.....	57
Gambar 4.7 <i>Activity</i> Menampilkan Status Tangki	58
Gambar 4.8 <i>Activity</i> Menampilkan Ketinggian Air	59
Gambar 4.9 Diagram Blok Sistem <i>Monitoring</i> Ketinggian Air.....	59
Gambar 4.10 Skema Keseluruhan Sistem <i>Monitoring</i> Ketinggian Air.....	61
Gambar 4.11 Sketsa Tangki Air Bagian Luar.....	62
Gambar 4.12 Sketsa Tangki Air Bagian Dalam.....	63
Gambar 4.13 Sketsa Alat Penampung Air Keseluruhan	64
Gambar 4.14 Skema Rangkaian Sensor Ultrasonik	65
Gambar 4.15 Rangkaian Sensor Ultrasonik dengan NodeMCU.....	66
Gambar 4.16 Skema Rangkaian LCD.....	67

Gambar 4.17 Rangkaian LCD dengan NodeMCU ESP8266.....	67
Gambar 4.18 Skema Rangkaian Buzzer.....	68
Gambar 4.19 Rangkaian Buzzer	69
Gambar 4.20 Gambar Skema Rangkaian Alat Keseluruhan.....	70
Gambar 4.21 Rangkaian Alat Secara Keseluruhan	70
Gambar 4.22 Flowchart Monitoring Ketinggian Air	71
Gambar 5.1 Arduino <i>Setup License Agreement</i>	73
Gambar 5.2 Arduiono <i>Setup Installation Options</i>	74
Gambar 5.3 Arduino <i>Setup Installation Folder</i>	74
Gambar 5.4 Proses <i>Installing</i> Arduino IDE	75
Gambar 5.5 Tampilan Jika Belum Instal USB Driver	75
Gambar 5.6 Proses Instal Arduino Selesai	75
Gambar 5.7 Tampilan Mulai Dari Arduino IDE.....	76
Gambar 5.8 Tampilan Arduino IDE Tab File <i>Preferences</i>	76
Gambar 5.9 Tampilan Preferences Arduino IDE.....	77
Gambar 5.10 Memasukan Link Library NodeMCU	77
Gambar 5.11 Tampilan Tab Tools Pada Arduino IDE	78
Gambar 5.12 Tampilan Tab Tools Memilih Board.....	79
Gambar 5.13 Tampilan LCD yang terhubung dengan NodeMCU	79
Gambar 5.14 Tampilan buzzer yang terhubung dengan NodeMCU.....	80
Gambar 5.15 Tampilan Sensor Ultrasonik dengan Board NodeMCU.....	80
Gambar 5.16 NodeMCU dengan <i>Relay Single Channel</i>	81
Gambar 5.17 LCD dan Sensor Ultrasonik dengan NodeMCU	81
Gambar 5.18 Tampilan LCD dan Buzzer dengan NodeMCU buzzer menyala....	82
Gambar 5.19 Tampilan LCD dan Buzzer dengan NodeMCU buzzer mati	82
Gambar 5.20 Tampilan <i>Interface</i> Virtual LCD Pada Aplikasi Blynk.....	83
Gambar 5.21 Tampilan <i>Interface</i> Virtual LED pada aplikasi blynk.....	84
Gambar 5.22 Tampilan Keseluruhan Modul Elektronika	84
Gambar 5.23 Tampilan Alat Secara Keseluruhan.....	85
Gambar 5.24 Kode Untuk Pengujian IP Address pada Board NodeMCU	86
Gambar 5.25 Hasil Uji Coba untuk mengetahui IP Address	87
Gambar 5.26 Kode Program pengujian Sensor Ultrasonik.....	88

Gambar 5.27 Tampilan Hasil Baca Sensor Ultrasonik	88
Gambar 5.28 Kode Program Pengujian LCD	89
Gambar 5.29 Tampilan Hasil Pengujian LCD	90
Gambar 5.30 Kode Program LCD dengan Buzzer.....	91
Gambar 5.31 Tampilan hasil LCD dengan Buzzer	91
Gambar 5.32 Kode Program Pengujian LCD dengan Sensor	92
Gambar 5.33 Kode Program untuk LCD virtual Blynk	94
Gambar 5.34 Tampilan Virtual LCD pada Aplikasi Blynk	94
Gambar 5.35 Tampilan Kode Program Pengujian LED	95
Gambar 5.36 Tampilan Virtual LED pada Aplikasi Blynk.....	95
Gambar 5.37 Kode Program Mengukur Jarak Pada Virtual LCD dan LED.....	96
Gambar 5.38 Tampilan Hasil Pengukuran Jarak Pada Virtual LCD dan LED.....	96
Gambar 5.39 Kode Program Keseluruhan Alat	98
Gambar 5.40 Kode Program Keseluruhan Alat Monitoring	99
Gambar 5.41 Hasil Pengujian Kondisi Cukup pada LCD.....	100
Gambar 5.42 Hasil Pengujian Kondisi Kosong pada LCD.....	100
Gambar 5.43 Tampilan Hasil Pengujian Pada Blynk.....	101

DAFTAR LAMPIRAN

Lampiran 1 Sketsa Alat <i>Monitoring</i> Ketinggian Air.....	107
Lampiran 2 Konsep Alat <i>Monitoring</i> Ketinggian Air.....	108
Lampiran 3 Uji Coba Alat Monitoring (Awal Percobaan).....	109
Lampiran 4 Form Wawancara.....	110
Lampiran 5 Kumpulan Sketch Program Alat <i>Monitoring</i>	112
Riwayat Hidup Penulis.....	124

BAB I

PENDAHULUAN

1.1 Latar Belakang

Ketinggian air yaitu mengukur air dari dasar sampai permukaan. Ketinggian air pada penampung air diukur dengan cara menggunakan alat yang di masukan kedalam penampung tersebut sehingga akan mendapatkan nilai ketinggian air. Hal ini menyebabkan penampung air menjadi penting dan diperlukan sebuah mekanisme untuk melakukan pengukuran demi mengetahui ketersediaan air pada penampung tersebut. Seringkali pengukuran dilakukan dengan cara-cara manual misalnya dengan melakukan pengecekan dengan mendatangi, melihat dan melakukan pengukuran ke tempat penampungan secara langsung. Cara konvensional ini merupakan cara mudah untuk dilakukan, tetapi akan sulit jika penampung air tersebut jauh dan sulit dijangkau, misalnya berada diatas sebuah bangunan.

Air sudah menjadi kebutuhan pokok untuk disetiap rumah yang digunakan untuk kebutuhan sehari-hari seperti mandi, mencuci pakaian dan lainnya. Disetiap rumah sekarang sudah mempunyai penampungan air berupa tangki air atau tendon sebagai upaya penghematan dari sebuah perusahaan PDAM karena dari hasil wawancara hampir 500liter perhari untuk memenuhi kebutuhan dari setiap rumah. Namun di setiap rumah tersebut pengecekan ketinggian air masih bersifat manual.

Masih banyak masyarakat atau pemilik rumah yang melakukan pengukuran ketinggian air pada tangki yang menggunakan cara-cara konvensional atau tradisional dimana cara ini dilakukan oleh beberapa orang secara bergantian, meskipun cara konvensional ini mudah dilakukan dan tidak memerlukan banyak biaya namun akan membutuhkan banyak SDM (Sumber Daya Manusia) untuk melakukan pengecekan juga akan menghabiskan banyak waktu dan tenaga, selain itu pengecekan dengan cara konvensional atau tradisional akan terkendala jika cuaca sedang terjadi hujan, sangat tidak memungkinkan melakukan pengecekan ketinggian air apalagi kondisi penampungan air berada pada tempat yang tinggi.

Untuk mengatasi hambatan atau permasalahan tersebut dapat menggunakan sebuah teknologi yang akan sangat membantu, dimana dengan menggunakan teknologi ini dapat menghemat energi, tenaga, dan waktu. Ada beberapa teknologi yang dapat digunakan diantaranya menggunakan *switch* pompa air dan menggunakan sebuah teknologi *monitoring* berbasis *Internet of Things* (IoT). *Switch* pompa air lebih dikenal dengan pelampung otomatis, seperti namanya cara kerja alat ini yaitu jika air penuh maka pompa akan otomatis mati, dan jika air habis maka otomatis pompa akan aktif. Kelebihan alat ini otomatis dalam segala hal maka tidak perlu lagi melakukan pengecekan lainnya, tapi masih memiliki kelemahan yaitu tidak akan diketahui berapa ketinggiannya dan harus disesuaikan dengan pompa yang digunakan. Selanjutnya menggunakan IoT yaitu dengan memanfaatkan sebuah mikrokontroler dan sensor Ultrasonik dengan media komunikasi Blynk. Kelebihan alat ini yaitu dapat memantau ketinggian air secara *realtime* dan tidak terpengaruh jenis pompa karena alat disimpan pada tangki secara langsung. Karena penggunaan teknologi IoT sedang berkembang maka penulis memutuskan untuk memilih IoT untuk *monitoring* ketinggian air pada tangki. Beberapa penelitian sebelumnya yang menggunakan teknologi IoT telah dilakukan sebelumnya yaitu Sistem Kendali Pemantauan Ketinggian Air Pada Tangki Berbasis Sensor Ultrasonik, Pembuatan Sistem *Monitoring* Ketinggian Air Dengan Sensor Ultrasonik Berbasis Mikrokontroler ATMEGA8535, dan Prototipe Sistem *Monitoring* Air Pada Tangki Berbasis *Internet of Things* Menggunakan NodeMCU Esp8266 Dan Sensor Ultrasonik.

Oleh karena itu, dengan adanya alat ini dapat memudahkan dalam mengetahui ketinggian air dalam tangki penampung yang diharapkan dapat sangat membantu dalam pengefisiensi dari segi biaya dan pengurangan SDM dalam melakukan *monitoring*. Maka, dari paparan paragraf diatas pada penulisan karya ilmiah ini, diambil judul “*Prototype Monitoring Ketinggian Air Berbasis Internet of Things Menggunakan Blynk dan Nodemcu Esp8266 Pada Tangki*”.

1.2 Batasan Masalah

Dalam perancangan dan pembuatan sistem ini terdapat beberapa batasan masalah antara lain:

1. Penelitian dilakukan dalam bentuk *prototype*.
2. Menggunakan mikrokontroler yang sudah terintegrasi Wi-Fi.
3. Menggunakan sensor jarak yang sudah ada *transmitter* dan *receiver*.
4. Mengabaikan selisih pada alat dengan ukuran sebenarnya.
5. Jarak maksimal sensor dengan permukaan air adalah 28cm.
6. Jarak minimal sensor dengan permukaan air adalah 15cm.
7. Menghitung volume air pada tangki.
8. Diasumsikan batas minimal air 5cm.
9. Hanya berfokus pada ketinggian air.
10. Diasumsikan hanya berjalan pada satu tangki saja.
11. Desain yang dibahas hanya tentang menghubungkan antar komponen.
12. *Output* berupa tampilan ketinggian air dan bunyi pada buzzer.
13. Menggunakan Aplikasi Blynk sebagai perantara komunikasi antara alat dan Android.
14. Hanya menginformasikan ketinggian secara jarak jauh dan hanya pada kondisi saat ini saja.

1.3 Rumusan Masalah

Dari latar belakang masalah diatas dapat dibuat rumusan masalah yaitu:

1. Bagaimana melakukan efisiensi dari penggunaan tenaga SDM?
2. Bagaimana mendapatkan informasi yang lebih baik tentang *monitoring* pada permukaan air?
3. Bagaimana peran komunikasi yang semula konvensional sekarang menggunakan aplikasi blynk yang dipadukan dengan NodeMCU ESP8266?

1.4 Tujuan

Tujuan dari perancangan dan pembuatan sistem ini merancang sebuah alat yang dapat digunakan untuk mengetahui ketinggian air pada tangki penampung air

sebagai upaya efisiensi dari SDM sebagai pengguna alat, mendapatkan informasi ketinggian air serta memahami cara kerja alat komunikasi blynk dengan NodeMCU ESP8266.

1.5 Metodologi Penelitian

Metode Penelitian yang dilakukan yaitu dengan melakukan pengumpulan data, dimulai dari Observasi dengan melakukan beberapa percobaan pada alat yang akan digunakan, dilanjutkan dengan wawancara kepada narasumber yang ahli pada bidang *monitoring* air, memilih narasumber ini karena berkaitan dengan rencana penelitian. Selanjutnya melakukan verifikasi untuk hasil observasi dan wawancara yang nantinya akan menjadi daftar pustaka yang dilengkapi dengan studi pustaka. Serta penelitian ini merupakan sebuah uji coba atau eksperimen dalam pengembangan teknologi baru, metode penelitian ini tidak mengikuti metode yang telah ada sebelumnya dan merupakan pembuktian apakah dengan menggunakan teknologi ini dapat mempermudah dalam sebuah proses pengerjaan yang sebelumnya membutuhkan banyak tenaga dan waktu menjadi lebih efisien.

1.6 Sistematika Penulisan

Adapun sistematika penulisan Karya Tulis Ilmiah ini sebagai berikut:

BAB I: PENDAHULUAN

Pada bab ini membahas latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian yang digunakan serta sistematika penulisan pada penelitian.

BAB II: TINJAUAN PUSTAKA

Pada bab ini berisi beberapa landasan teori dari beberapa jurnal ilmiah yang dijadikan sebagai referensi penelitian, dan dasar teori yang merupakan teori penunjang dalam penelitian.

BAB III: METODOLOGI PENELITIAN

Pada Bab ini berisikan kerangka pikir serta metode penelitian yang digunakan, kebutuhan yang dipakai dalam penelitian, pengembangan *software* dan *hardware* serta penjelasan dari kerangka pikir dan disusun berdasarkan kajian teoritik yang sesuai dengan sistematika penulisan.

BAB IV: ANALISIS DAN PERANCANGAN

Pada Bab ini berisi analisis dan perancangan, bab ini menjelaskan tentang melakukan sebuah analisa terhadap masalah dan mulai melakukan beberapa rancangan terkait pemecahan masalahnya serta melaporkan hasil yang didapatkan.

BAB V: IMPLEMENTASI DAN PENGUJIAN

Pada Bab berisi impementasi dan pengujian dari hasil pemecahan masalah dan perancangan yang didapat secara teoritis serta implementasi secara menyeluruh pada alat yang dibangun.

BAB VI: KESIMPULAN DAN SARAN

Pada Bab ini berisi kesimpulan yang diperoleh dari bab-bab sebelumnya dan saran-saran dari hasil pengamatan.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan Teori berisi referensi dari jurnal yang berkaitan dengan judul dan objek penelitian, berikut beberapa judul jurnal yang digunakan dalam proses penelitian ini:

2.1.1 Sistem Kendali Pemantauan Ketinggian Air Pada Tangki Berbasis Sensor Ultrasonik. (Amalia Awaliah dan Adnan Rafi al Tahtawi, KOPERTIP: Jurnal Ilmiah Manajemen Informatik dan Komputer, 2017)

“Pada penelitian ini mengemukakan bagaimana *me-monitoring* ketinggian air agar dapat mengontrol kualitas dan kuantitas air pada tangki, karena kesalahan pada sistem sebelumnya seringkali menyebabkan kerugian pada suatu industri. Penelitian ini menggunakan mikrokontroler Arduino UNO dan Sensor Ultrasonik. Perancangan *hardware* dirancang sedemikian rupa menggunakan 2 unit tangki dengan diameter 21 cm dan tinggi 28 cm. Tangki pertama sebagai tangki kontrol dan yang kedua merupakan tangki sistem. Cara kerja alat ini sensor akan mendeteksi ketinggian air dan mengirimkan data kepada mikrokontroler. Untuk merancang programnya penelitian ini menggunakan *Software* Arduino IDE yang nantinya akan ditanam pada mikrokontrolernya, ketinggian maksimum yang ditetapkan adalah 25cm. Dalam pengujian dengan ketinggian maksimum 25 cm sensor dapat mengukur jarak dari 5 cm hingga 25 dengan tingkat kesalahan sebesar 4.9%. Pada ketinggian 0-4 cm sensor tidak dapat mendeteksi karena permukaan tidak stabil. Pada penelitian ini terdapat beberapa pengujian diantaranya adalah pengujian sistem kendali histeris Skema kendali histeris yang telah dirancang akan diuji untuk mengetahui respon sistem ketika sensor mendeteksi nilai *setpoint*. Pengujian dilakukan dengan cara menampilkan data grafik pada perangkat antar muka yang dirancang.”.

2.1.2 Rancang Bangun Sistem *Monitoring Volume* Dan Pengisian Air Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler AVR Atmega8. (Adhitya Permana, Dedi Triyanto, dan Tedy Rismawan, Jurnal Coding Sistem Komputer Untan, 2015)

“Penelitian ini lebih menitik beratkan pada *monitoring* persediaan air pada bak karena aliran PDAM yang sering tidak stabil. Penelitian ini mengandalkan sebuah mikrokontroler ATmega8, Sensor Ultrasonik untuk sensor jarak, dan modul *Bluetooth* HC-06 sebagai media komunikasi. Penelitian *Monitoring* ketinggian air menggunakan sensor ultrasonik berbasis ATmega8 dilaksanakan secara bertahap, mulai dari analisa masalah, studi pustaka, perancangan *hardware*, sampai perancangan aplikasi. Alur kerja sistem pada mikrokontroler ini dimulai dari konfigurasi, dilanjutkan dengan membaca sensor jarak, melakukan konversi ke data aktual air, lalu dibandingkan dengan level minimum, jika level aktual lebih kecil dari level minimum maka *buzzer* dan pompa akan aktif, jika tidak maka akan membaca apakah level aktual lebih besar dari level minimum maka *buzzer* dan pompa akan mati, jika tidak maka akan kembali membaca sensor jarak. Jika selesai maka akan mengirimkan data aktual ke handphone via *Bluetooth*. Setelah semua selesai penelitian dilakukan dengan pengujian, pengujian yang dilakukan yaitu mengukur ketinggian air pada bak dengan ukuran 65x45cm dengan ketinggian bak sekitar 38cm, sebelumnya telah ditentukan pada aplikasi yang mencakup peletakan sensor pada ketinggian 65cm, batas bak air 20cm dan batas minimum 15%. Hasil ketinggian aplikasi dibandingkan dengan pengukuran aktual dengan menggunakan meteran, setelah itu dilakukan maka di dapatkan hasil bahwa tinggi permukaan air dalam bak sama dengan ketinggian air pada *display* aplikasi.”

2.1.3 Prototipe Sistem *Monitoring Air* Pada Tangki Berbasis *Internet of Things* Menggunakan NodeMCU Esp8266 Dan Sensor Ultrasonik. (Ulumuddin, dkk, SENTER (Seminar Nasional Teknik Elektro) 2017, 2017)

“Penelitian ini membahas tentang bagaimana pengefisiensian penggunaan air tawar dimana persediaan air tawar di bumi hanya 3%. Pengefisiensian ini

diantaranya memantau kondisi ketersediaan pada bak penyimpanan, yang diharapkan dapat memberikan peran dalam upaya penghematan air yang akan berguna untuk mencegah terjadinya krisis air. Alat ini terdiri dari perangkat keras yang telah memiliki wifi, dan memiliki server yang berguna sebagai media komunikasi. Tahapan penelitian ini studi literatur, pembuatan *prototype* dan pengujian aplikasi. Skema kerjanya ketika *jet pump* diaktifkan maka *rain sensor* akan mendeteksi apakah itu air atau angin, selanjutnya sensor ultrasonik akan mendeteksi ketinggian air secara *real-time* yang nantinya pengguna bisa mengakses pada *browser* dengan berbagai *device*.”

2.2 Dasar Teori

Dasar teori ini berisikan teori-teori penunjang pada penelitian. Berikut ini adalah beberapa teori yang digunakan pada penelitian *monitoring* ketinggian air:

2.2.1 *Prototype*

Prototype adalah alat yang memberikan ide bagi pembuat maupun bagi pemakai tentang cara sistem berfungsi dalam bentuk lengkapnya, proses untuk menghasilkan sebuah *prototype* disebut *prototyping*. *Prototyping* adalah proses pembuatan model sederhana yang memungkinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototyping* memberikan fasilitas bagi pengembang dan pemakai untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat yang akan dibuat. *Prototyping* merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. (Raymond McLeod, 2008).

Kunci agar *prototype* ini berhasil dengan baik adalah dengan mendefinisikan aturan-aturan main pada saat awal, yaitu pelanggan dan pengembang harus setuju bahwa *prototype* dibangun untuk mendefinisikan kebutuhan. *Prototype* akan di hilangkan sebagian atau seluruhnya dan perangkat akan dirancang dengan kualitas dan implementasi yang telah ditentukan.

Penjelasan pengertian *prototype* berdasarkan Ilmu Interaksi Manusia dan Komputer. *Prototype* merupakan alat yang digunakan untuk mensimulasikan beberapa atau tidak semua fitur dari sistem yang akan dibuat. Terdapat 3 pendekatan utama *prototyping* yaitu:

- 1) ***Throw-Away***: *Prototype* dibuat dan dites. Pengalaman yang diperoleh dari pembuatan *prototype* digunakan untuk membuat produk akhir, kemudian *prototype* tersebut akan dibuang (tidak digunakan).
- 2) ***Incremental***: Produk akhirnya dibuat sebagai komponen-komponen yang terpisah. Desain produk akhirnya secara keseluruhan hanya ada 1 tetapi dibagi dalam komponen-komponen lebih kecil dan terpisah (*independent*).
- 3) ***Evolutionary***: Pada metode ini, *prototype* nya tidak dibuang tetapi digunakan untuk iterasi desain berikutnya. Dalam hal ini, sistem atau produk yang sebelumnya dipandang sebagai evolusi dari versi awal yang sangat terbatas menuju produk final atau produk akhir.

2.2.2 *Monitoring*

Monitoring adalah aktifitas yang ditunjukan untuk memberikan informasi tentang sebab dan akibat dari suatu kebijakan yang sedang dilaksanakan. *Monitoring* dilakukan ketika sebuah kebijakan sedang di implemantasikan. *Monitoring* diperlukan agar kesalahan awal dapat segera diketahui dan dapat dilakukan tindakan perbaikan, sehingga mengurangi resiko yang lebih besar. (Sekretariat Kabinet Republik Indonesia, 2015). Adapun beberapa pengertian *Monitoring* menurut para ahli:

- 1) (Clayton dan Petry 1983) *Monitoring* sebagai suatu proses mengukur, mencatat, mengumpulkan, memproses dan mengkomunikasikan informasi untuk membantu pengambilan keputusan manajemen atau proyek.
- 2) (Frances Rubin 1995:15) “*Monitoring is a continuous, methodical process of data collection and information-gathering throughout the life of a project. The information collected can be used for regular evaluation of progress, so that adjustments can be made while the work is going on.*”

- 3) (Handoko 1995) *Monitoring* didefinisikan sebagai proses untuk “menjamin” bahwa tujuan-tujuan organisasi dan manajemen tercapai. Ini berkenaan dengan cara-cara membuat kegiatan-kegiatan sesuai dengan yang direncanakan.

Tujuan *Monitoring* sebagai berikut:

- 1) Menjaga agar kebijakan yang sedang diimplementasikan sesuai dengan tujuan dan sasaran.
- 2) Menemukan kesalahan sedini mungkin sehingga mengurangi resiko yang lebih besar.
- 3) Melakukan tindakan modifikasi terhadap kebijakan apabila hasil *monitoring* mengharuskan untuk itu.

2.2.3 Ketinggian Air

Ketinggian air adalah suatu parameter yang digunakan untuk berbagai keperluan, misalnya untuk pengisian air pada penampung air atau tangki agar dapat tepat sasaran. Tinggi muka air biasanya dinyatakan dalam satuan meter (m) atau centimeter (cm). Pengukuran tinggi air dapat dilakukan dengan cara manual menggunakan alat duga air biasa (*non-recording gauges*) dan atau cara otomatis menggunakan alat duga air otomatis (*recording gauges*).

Dengan menggunakan alat duga air otomatis bisa didapatkan informasi yang dikonversi dari tekanan air yang diterima oleh perangkat uji menjadi data pengukur ketinggian air, dari alat yang dimasukan ke kedalaman yang diinginkan sampai ke permukaan. Terdapat beberapa opsi dari penggunaan alat otomatis diantaranya menggunakan alat duga air otomatis atau menggunakan teknologi IoT.

2.2.4 Internet of Things

Internet of Things atau dikenal juga dengan singkatan IoT, merupakan sebuah konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus-menerus. Adapun kemampuan seperti berbagi data, *remote control*, dan sebagainya, termasuk juga pada benda di dunia nyata.

Contohnya bahan pangan, elektronik, koleksi, peralatan apa saja, termasuk benda hidup yang semuanya tersambung ke jaringan lokal dan global melalui sensor yang tertanam dan selalu aktif. *Internet of Things* (IoT) adalah sebuah konsep/skenario dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia (Yudhanto, Yudha 2015).

Istilah *Internet of Things* (IoT) mulai dikenal pada tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi Kevin Ashton, *cofounder and executive director of the Auto-ID Center* di MIT.

“The world-changing promise of Internet of Things technology cannot be realized without a focus on security as the bedrock. The whole point of the Internet of Things is that a combination of human and automated systems will make decisions about the allocation of time and resources based on inputs from sensors and systems distributed throughout our environment and our lives. If those sensor inputs are not trustworthy, human and automated systems will either disregard Internet of Things sensors and systems, or worse, the human and automated systems will make bad decisions. In extreme cases, such bad decisions could cause financial losses or even injury to people and our environment.” (Ashton, K. 2017).

Internet of Things merupakan sebuah teknologi yang telah berkembang, teknologi IoT sudah diterapkan di berbagai bidang misalkan di bidang pertanian, dimana memungkinkan pemilik memberikan makan pada ternak dengan cara mengontrol dari jarak jauh dengan memanfaatkan internet. IoT juga merupakan alat komunikasi berbasis internet yang sangat memungkinkan untuk melakukan komunikasi antara mesin dengan mesin.

2.2.5 Android

Android merupakan sebuah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Sistem operasi ini resmi rilis

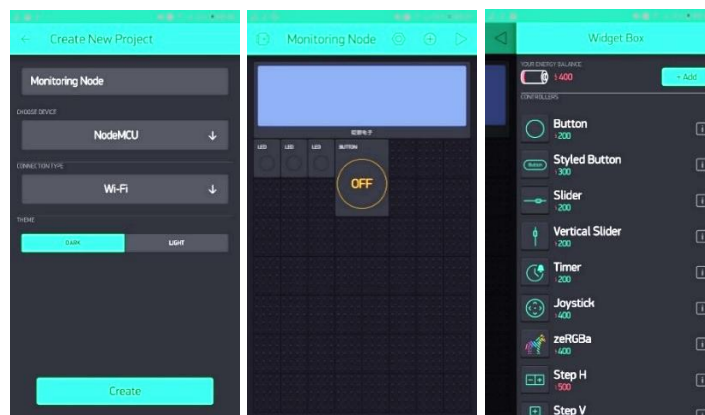
pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance. Ponsel Android pertama dijual pada tahun 2008.

Android adalah sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karena *source code* (kode sumber) dari sistem operasi Android dapat dilihat, di-download, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi Android karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi.

2.2.6 Blynk

Blynk adalah aplikasi untuk iOS dan Android OS untuk mengontrol Arduino, NodeMCU, Raspberry Pi dan sejenisnya melalui Internet. Aplikasi ini dapat digunakan untuk mengendalikan perangkat *hardware*, menampilkan data sensor, menyimpan data, visualisasi, dan lain-lain.

Aplikasi Blynk memiliki 3 komponen utama. yaitu Aplikasi, *Server*, dan *Libraries*. Blynk *server* berfungsi untuk menangani semua komunikasi diantara smartphone dan *hardware*. *Widget* yang tersedia pada Blynk diantaranya adalah *Button*, *Value Display*, *History Graph*, Twitter, dan Email. Aplikasi Blynk memiliki beberapa *hardware* mikrokontroler yang dapat melakukan komunikasi sesuai dengan *hardware* yang dipilih. NodeMCU dikontrol dengan Internet melalui WiFi, chip ESP8266, Blynk akan dibuat online dan siap untuk *Internet of Things*. Contoh *user interface* dari aplikasi Blynk sebagai berikut:



Gambar 2.1 Tampilan *interface* dan *Widget Box* aplikasi Blynk

2.2.7 NodeMCU ESP8266

ESP 8266 adalah sebuah modul WiFi yang akhir-akhir ini semakin digemari para *hardware developer*. NodeMCU merupakan sebuah *platform* IoT yang bersifat *OpenSource* dan yang bersifat SOC (*System on Chip*), sehingga dapat melakukan pemrograman langsung ke NodeMCU ESP8266 tanpa memerlukan mikrokontroler tambahan. Kelebihan lainnya, ESP8266 ini dapat menjalankan peran sebagai *ad hoc* akses poin maupun klien sekaligus. (Mannan Mehta, 2015).

NodeMCU pada dasarnya adalah pengembangan dari ESP8266 dengan *firmware* berbasis e-Lua. Pada NodeMcu dilengkapi dengan *micro usb port* yang berfungsi untuk pemrograman maupun *power supply*. Selain itu juga pada NodeMCU di lengkapi dengan tombol *push button* yaitu tombol *reset* dan *flash*. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan *package* dari ESP8266. Jika menggunakan bahasa Lua maka dapat menggunakan *tool* Lua loader maupun Lua uploader. Selain dengan bahasa Lua NodeMCU juga support dengan software Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE.



Gambar 2.2 Board NodeMCU ESP8266 v3 Lolin

Dibawah ini tabel spesifikasi dan konfigurasi pin dari mikrokontroler NodeMCU ESP8266 v3:

Tabel 2.1 Tabel Konfigurasi Pin NodeMCU ESP8266 v3

SPESIFIKASI	NODEMCU V3
Mikrokontroler	ESP8266
Ukuran Board	57 mmx 30 mm
Tegangan Input	3.3 ~ 5V
GPIO	13 PIN
Kanal PWM	10 Kanal
10 bit ADC Pin	1 Pin
Flash Memory	4 MB
Clock Speed	40/26/24 MHz
WiFi	IEEE 802.11 b/g/n
Frekuensi	2.4 GHz – 22.5 Ghz
USB Port	Micro USB
Card Reader	Tidak Ada
USB to Serial Converter	CH340G

2.2.8 Tangki Air

Tangki air adalah wadah untuk menyimpan air. Tangki air digunakan untuk menyimpan persediaan air, mulai dari persediaan air minum, kebutuhan rumah tangga, tanaman dan ternak hingga digunakan sebagai media penyimpanan bahan kimia oleh perusahaan yang bergerak dibidang kimia. Dalam perkembangannya tangki air mengalami banyak perubahan desain dan bahan pembuatannya. Untuk saat ini bahan yang paling banyak digunakan untuk membuat tangki air adalah plastik dan stainless steel.

2.2.9 Pompa Air

Water Pump atau pompa air merupakan elemen yang berfungsi untuk menyerap sekaligus mendorong air yang terdapat pada sistem pendinginan sehingga dapat bersirkulasi pada mesin.

Pompa air secara umum adalah alat yang digunakan untuk memindahkan cairan atau *fluida* dari suatu tempat ke tempat lainnya melalui saluran (pipa) dengan menggunakan tenaga motor listrik yang bersumber dari listrik PLN yang berfungsi untuk mendorong air secara terus menerus. Disaat pengoperasian pompa air beroperasi dengan prinsip membuat perbedaan di sisi tekanan dan sisi bagian hisap, perbedaan tekanan tersebut dihasilkan dari sebuah mekanisme yang terjadi pada roda implek yang membuat keadaan sisi hisap menjadi tidak bergerak. Perbedaan inilah yang menghisap air sehingga dapat berpindah dari suatu tempat ke tempat lain.

2.2.10 Logika Fuzzy

Logika Fuzzy atau *Fuzzy Logic* pertama dikenalkan oleh Prof Lotfi A. Zadeh pada tahun 1965. *Fuzzy Logic* merupakan suatu metode pengambilan keputusan berbasis aturan yang digunakan untuk memecahkan keabu-abuan masalah pada sistem yang sulit dimodelkan atau ambiguitas. Dasar logika fuzzy adalah teori himpunan fuzzy.

Adapun beberapa alasan mengapa menggunakan logika fuzzy yaitu sebagai berikut:

- 1) Konsep logika fuzzy mudah dimengerti.
- 2) Penggunaan logika fuzzy yang fleksibel.
- 3) Logika fuzzy mampu memodelkan fungsi-fungsi *non-linier* yang sangat kompleks.
- 4) Tidak perlu adanya proses pelatihan untuk memodelkan pengetahuan yang dimiliki pakar.
- 5) Logika fuzzy didasari pada Bahasa sehari-hari sehingga mudah dimengerti.

Himpunan fuzzy disebut himpunan tegas (*crisp*), nilai keanggotaan suatu *item* x dalam suatu himpunan A yang dituliskan dengan $\mu_A [x]$, dimana memiliki dua buah kemungkinan nilai yaitu:

- a) Satu (1), yang memiliki arti bahwa suatu item menjadi anggota dalam suatu himpunan tertentu.
- b) Nol (0), yang memiliki arti bahwa suatu item tidak menjadi anggota dalam suatu himpunan tertentu.

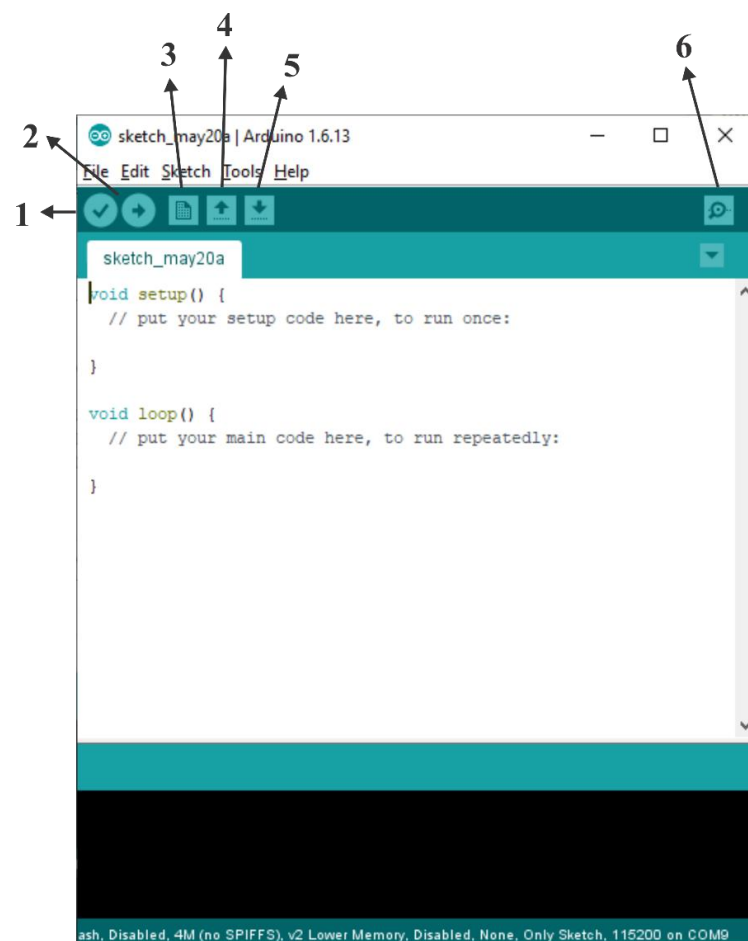
2.2.11 Software Arduino IDE (*Integrated Development Environment*)

Arduino *Integrated Development Environment* atau disingkat Arduino IDE ini terdiri dari editor teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah toolbar dengan *tools* untuk fungsi yang umum dan beberapa menu. Arduino IDE terhubung ke Arduino Board untuk mengunggah program dan juga untuk berkomunikasi dengan Arduino (Simanjuntak, Matur. 2012).

Software yang ditulis menggunakan Arduino IDE disebut *sketch*. *Sketch* ditulis pada editor teks, disimpan dengan berekstensi *.ino*. Area pesan memberikan informasi dan pesan error ketika penyimpanan atau pada saat membuka *sketch*. Pada fungsi konsol menampilkan *output* teks dari Arduino IDE dan juga menampilkan pesan error ketika akan mengkompilasi *sketch*. Pada sudut kanan bawah dari jendela Arduino IDE menunjukkan jenis *board* dan port serial yang sedang digunakan. Tombol *toolbar* pada Arduino IDE untuk mengecek dan mengunggah *sketch*, membuat, membuka dan menyimpan *sketch*, dan mengunggah *sketch*, membuat, membuka dan menyimpan *sketch*, dan menampilkan serial monitor. Arduino IDE menggunakan bahasa pemrograman C++ dengan tingkat dan versi yang disederhanakan, sehingga lebih mudah dalam pengerjaan. Agar bisa melakukan *upload sketch* ke board NodeMCU hanya perlu menambahkan library board NodeMCU ESP8266 yang bisa ditambahkan dengan cara pergi ke Preferences yang ada di menu File dan bisa di download disana. Selanjutnya lakukan perubahan pada menu *tools* lalu pilih board dan pilih NodeMCU 1.0 karena itu merupakan versi yang paling baru atau bisa disesuaikan dengan board yang digunakan.

Pada Gambar 2.3 merupakan bagian-bagian dari Arduino IDE. Berikut ini adalah *tools* serta fungsi dari masing-masing:

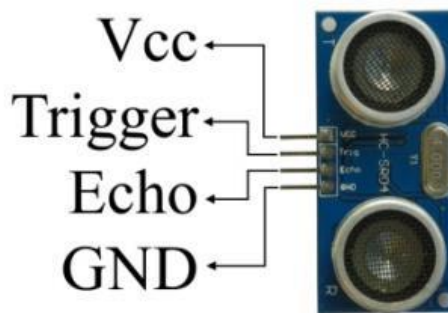
- 1) *Verify* berfungsi sebagai mengecek error pada code program
- 2) *Upload* berfungsi sebagai Mengumpulkan dan mengunggah program Arduino board.
- 3) *New* berfungsi sebagai membuat sketch yang baru.
- 4) *Open* berfungsi sebagai menampilkan sebuah menu dari seluruh sketch yang berada didalam sketch.
- 5) *Save* berfungsi sebagai Menyimpan sketch.
- 6) *Serial Monitor* berfungsi sebagaia membuka *serial monitor* (Simanjuntak, Matur. 2012).



Gambar 2.3 Arduino IDE (Sumber: Arduino, inc)

2.2.12 Modul Ultrasonik

Sensor Ping merupakan jenis sensor ultrasonik yang bekerja melalui pemancaran gelombang bunyi dengan frekuensi 40kHz dengan kecepatan 344 m/s. Cara kerja sensor ini berdasarkan pada prinsip dari pantulan suatu gelombang sehingga dapat dipakai untuk menafsirkan jarak suatu benda dengan frekuensi tertentu. Disebut sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik.



Gambar 2.4 Konfigurasi PIN Sensor Ultrasonik

Fungsi Pin pada Sensor Ultrasonik:

Tabel 2.2 Fungsi Pin Pada Sensor Ultrasonik

No	PIN	Fungsi
1	VCC	Sumber Tegangan 5V
2	Trigger	Pemicu Sinyal Sonar dari Sensor
3	Echo	Penangkap Pantulan Sinyal Sonar
4	GND	Ground

Sensor ultrasonik HC-SR04 memiliki 4 pin, yaitu VCC, GND, *Trigger*, dan *Echo*. VCC digunakan untuk arus positif, pada pin *Trigger* digunakan *receiver* atau

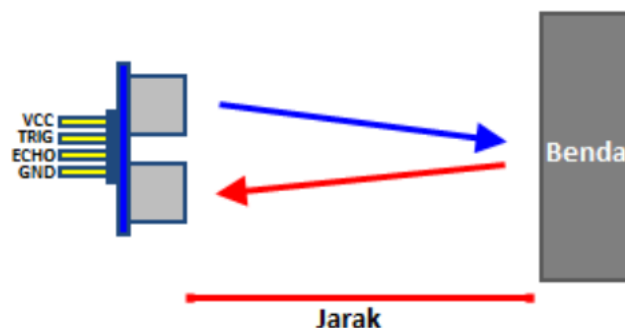
penerima frekuensi, *Echo* digunakan untuk *Tranducer* atau pemancar suara dan GND digunakan untuk arus negatif.

Karakteristik Sensor Ultrasonik HC-SR04:

1. Tegangan sumber operasi tunggal 5V.
2. Konsumsi arus berjumlah 15mA.
3. Frekuensi operasi 40KHz. Minimum pendeteksi jarak 2cm.
4. Maksimum jarak 4m.
5. Sudut pantul gelombang 15 drajat.
6. Minimum waktu ukur 10 *microseconds* dengan pulsa.
7. Dimensi 45 x 20 x 15 mm.

Cara kerja dari Sensor Ultrasonik:

Modul sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan *piezoelektrik* dengan frekuensi tertentu. Pada dasarnya, alat ini akan menembakkan gelombang ultrasonik yang menjadi targetnya. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu pantulan yang diterima dapat dilihat pada Gambar 2.5



Gambar 2.5 Cara Kerja Sensor Ultrasonik

2.2.13 Buzzer

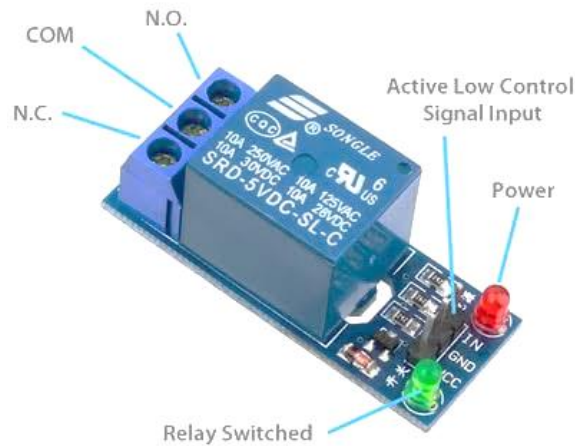
Buzzer adalah sebuah komponen elektronika yang dapat mengubah sinyal listrik menjadi gelombang suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loudspeaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik kedalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer ini biasa dipakai pada sistem alarm juga bisa dipakai sebagai indikasi suara. Buzzer merupakan komponen elektronika yang tergolong transduser yaitu komponen yang memiliki 2 kaki yaitu positif dan negatif untuk menggunakannya bisa memberikan tegangan 3-12V. Dalam penelitian ini buzzer berperan sebagai indikator jika ketinggian air telah mencapai titik bahaya.

2.2.14 Relay

Relay merupakan salah satu komponen elektronika yang berfungsi sebagai saklar mekanik. Fungsi *relay* yaitu memisahkan rangkaian listrik tegangan tinggi dengan rangkaian tegangan rendah. *Relay* memiliki 5 (lima) buah kaki. 2 (dua) kaki untuk mengaktifkan koil. Kedua kaki ini tidak bertanda, artinya boleh terbalik dalam pemasangannya. Tiga kaki lainnya berfungsi sebagai saklar yang terdiri dari kaki *Common* (COM), kaki *Normally Open* (NO), dan kaki *Normali Closed* (NC). Dalam keadaan koil tidak dialiri arus listrik, kaki COM akan terhubung ke kaki NC. Dalam keadaan koil dialiri arus listrik, kaki COM akan terhubung dengan kaki NO.

Relay yang digunakan yaitu *single channel relay 5v* yang artinya hanya memiliki 1 (satu) COM, NO, dan NC yang hanya bisa mengaktifkan atau mematikan 1 (satu) alat saja. Dimana *relay* ini akan memiliki sebuah pin signal input yang berfungsi untuk melakukan kontrol pada board NodeMCU ESP8266 dan di dalam kasus ini alat yang digunakan adalah pompa air yang memiliki peranan penting.

Berikut adalah contoh gambar dari *relay single channel 5v*:



Gambar 2.6 *Relay Single Channel 5V*

2.2.15 LCD (*Liquid Crystal Display*)

LCD atau *Liquid Crystal Display* adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. Terdapat banyak jenis LCD yang beredar di pasaran. Namun ada standarisasi yang cukup populer digunakan yaitu modul LCD dengan tampilan 16x2 (16 kolom x 2 baris) dengan konsumsi daya yang rendah. LCD dengan jenis seperti ini memungkinkan melakukan pemrogram untuk mengoperasikan komunikasi data secara 8 bit atau 4 bit. Berikut spesifikasi pada kaki LCD 16x2 dan tampilan gambar LCD yang digunakan:

Tabel 2.3 Pin Pada LCD 16x2

Pin	Keterangan
1	Ground
2	VCC
3	V0 (Tegangan Kontras)

Pin	Keterangan
4	RS (Register Select)
5	RW (Read/Write LCD Register)
6	EN (Enable)
7-14	Data I/O pin
15	Backlight +
16	Backlight -

Tabel 2.3 Pin Pada LCD 16x2 (lanjutan)

Berikut merupakan gambar dari tampilan LCD 16x2 yang digunakan:

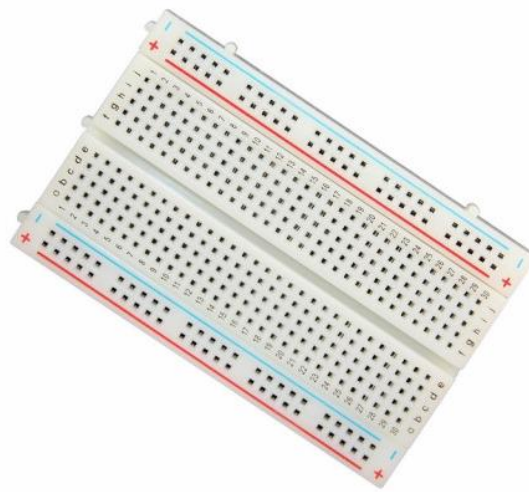
Gambar 2.7 LCD (*Liquid Crystal Display*) 16x2

2.2.16 Breadboard

Breadboard atau yang biasa disebut *project board* adalah dasar konstruksi sebuah sirkuit elektronik dan merupakan *prototype* dari suatu rangkaian elektronik. Breadboard banyak digunakan untuk merangkai komponen karena dengan menggunakan breadboard pembuatan *prototype* tidak memerlukan proses menyolder. Breadboard memiliki sifat solderless alias tidak memerlukan solder sehingga dapat digunakan kembali dengan demikian breadboard sangat cocok digunakan pada tahapan proses pembuatan *prototype* serta membantu dalam

berkreasi dalam desain sirkuit elektronika. Berbagai sistem elektronik dapat dimodelkan dengan menggunakan breadboard, mulai dari sirkuit adalog dan digital kecil sampai dengan pembuatan *Central Processing Unit* (CPU). (Satria Agung, 2015).

Berikut merupakan gambar dari Breadboard yang digunakan dan memudahkan dalam penyambungan kabel:



Gambar 2.8 Gambar Breadboard

2.2.17 UML

Menurut (Henderi, 2010) “*Unified Modeling Language* (UML) adalah bahasa yang telah menjadi standar untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”.

Menurut Nugroho (2010:6), “UML (*Unified Modeling Language*) adalah Bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma (berorientasi objek).” Pemodelan sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

UML adalah sebuah *tools* yang biasa digunakan untuk membuat atau merancang sebuah perangkat lunak. UML juga merupakan metode pengembangan OOP yang digunakan untuk penyederhanaan dari sebuah permasalahan yang ada. UML juga dapat digunakan sebagai media *transfer* antar *developer* yang memungkinkan *developer* saling mengembangkan aplikasi atau sistem tersebut.

Ada 3 diagram yang terdapat pada UML yaitu *Usecase Diagram*, *Class Diagram*, dan *Activity Diagram*. Namun yang digunakan hanya 2, berikut penjelasan dari diagram yang digunakan:




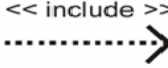
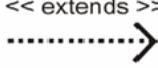

a. *Usecase Diagram*

Usecase menurut Fowler (2005, p141) adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Usecase Diagram* menampilkan aktor mana yang menggunakan *use case* mana, *use case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case*.

Usecase diagram menggambarkan interaksi antara sistem dengan sistem eksternal dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem.

Diagram ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

Berikut ini adalah simbol-simbol yang ada pada *Use Case Diagram*:

Nama Simbol	Simbol
Aktor	
Use Case	
Association Relationship	
Include Relationship	
Extend Relationship	
Generalisasi Relationship	

Gambar 2.9 Notasi Pada *Usecase Diagram*

b. *Activity diagram*




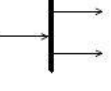


Activity diagram menggambarkan berbagai alur aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Decision digunakan untuk menggambarkan *behavior* pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa objek *swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktifitas tertentu.

Diagram ini memperlihatkan aliaran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini penting terutama

dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Berikut ini adalah simbol-simbol pada *activity diagram*:

Simbol	Keterangan
	Start Point
	End Point
	Activities
	Fork (Percabangan)
	Join (Penggabungan)
	Decision
Swimlane	Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)

Gambar 2.10 Simbol-simbol pada *Activity Diagram*

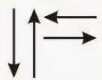















2.2.18 Flowchart

Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (intruksi) dengan proses lainnya dalam suatu proyek.

Dalam perancangan *flowchart* sebenarnya tidak ada rumus atau patokan yang bersifat mutlak (pasti). Hal ini didasari oleh *flowchart* adalah sebuah gambaran dari hasil pemikiran dalam menganalisa suatu permasalahan. Karena setiap analisa akan menghasilkan hasil yang bervariasi antara satu dan yang lainnya. Kendati begitu secara garis besar setiap perancangan *flowchart* selalu terdiri dari tiga bagian yaitu: input, proses, output. Dalam penggunaannya *flowchart* bertujuan untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, teratur, rapi, dan jelas.

Dalam penulisan *flowchart* dikenal 2 (dua) model yaitu Sistem *Flowchart* dan Program *Flowchart*. Sistem *flowchart* yaitu bagan yang memperlihatkan urutan prosedur dan proses dari beberapa file dalam media tertentu. Melalui *flowchart* ini terlihat jenis media penyimpanan yang dipakai dalam pengolahan data. Sedangkan Program *Flowchart* yaitu bagan yang memperlihatkan urutan dan hubungan proses dalam suatu program. Terdapat 2 (dua) jenis penggambaran program *flowchart* yaitu *conceptual flowchart* dan *Detail flowchart*.

Berikut adalah simbol-simbol yang digunakan dalam menggambar sebuah *Flowchart*:

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2.11 Simbol-simbol dalam *flowchart*

2.2.19 Flowmap

Menurut Al Bahra Bin Ladjamudin (2006) *Flowmap* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu

masalah. *Flowmap* merupakan cara penyajian dari suatu algoritma. Ada dua macam Flow map yang menggambarkan proses dengan komputer yaitu:

- a. System Flow map Bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.
- b. Program Flow map Bagan yang memperlihatkan urutan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program.




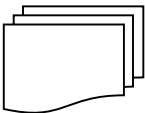
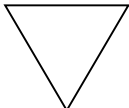

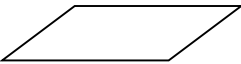
Flowmap adalah bagan yang menunjukan aliran di dalam program atau prosedur sistem secara logika. *Flowmap* ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan flowmap ini harus dapat memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi. Adapun pedoman-pedoman dalam pembuatan flowmap adalah sebagai berikut:

1. *Flowmap* sebaiknya digambarkan dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam *flowmap* harus ditunjukan dengan jelas.
3. Harus ditunjukan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan didalam *flowmap* sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan didalam flow map harus didalam urutan yang semestinya.
6. Kegiatan yang terpotong dan akan disambung ditempat lain harus ditunjukan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol simbol *flowmap* yang standar.

Flowmap memiliki beberapa symbol khusus yang digunakan didalam setiap perancangannya. Ini memudahkan pengguna lain dalam melakukan pembacaan pada setiap perancangan yang dibuat. Dibawah ini merupakan simbol-simbol dalam flowmap.

Adapun simbol-simbol yang sering digunakan dalam *flowmap* dapat dilihat pada tabel 2.4 berikut ini:

Tabel 2.4 Simbol dan Keterangan Flowmap

Simbol	Deskripsi
	Simbol yang digunakan untuk menunjukkan awal atau akhir dari suatu proses
	Menunjukkan dokumen input dan output baik untuk proses manual mekanik atau komputer
	Menunjukkan pekerjaan manual
	Menunjukkan multi dokumen
	Pengarsipan Data
	Menunjukkan Proses
	Simbol input/output digunakan untuk mewakili data input/output

2.2.20 Metode Penelitian

Metode Penelitian adalah langkah-langkah yang akan dilakukan oleh peneliti dalam sebuah penelitian dalam rangka untuk mengumpulkan informasi atau data serta melakukan investigasi pada data yang telah di dapatkan tersebut. Metode

penelitian memberikan gambaran rancangan penelitian yang meliputi prosedur penelitian, langkah-langkah yang harus ditempuh, waktu penelitian, sumber data dan bagaimana data itu diproses. Metode penelitian yang digunakan yaitu metode eksperimen yang disesuaikan dengan beberapa kebutuhan penelitian saat ini.

Metode Eksperimen (Sunarti, 2009) merupakan metode penelitian yang menguji hipotesis berbentuk hubungan sebab-akibat melalui pemanipulasian variabel independen dan menguji perubahan yang diakibatkan oleh pemanipulasian tersebut. Maka metode eksperimen ini digunakan untuk mengukur perubahan yang terjadi setelah dilakukannya pemanipulasian. Oleh karena itu, penelitian eksperimen erat kaitannya dalam menguji suatu hipotesis dalam rangka mencari pengaruh, hubungan, maupun perbedaan perubahan terhadap objek yang dikenakan perlakuan.

Beberapa tahapan metode penelitian kali ini diawali dengan pengumpulan data, yang dimulai dari melakukan beberapa percobaan pada alat dan modul-modul yang digunakan sebagai observasi awal, yang dilanjutkan dengan wawancara kepada narasumber yang kompeten pada bidangnya. Hasil dari observasi dan wawancara dilakukan validasi dengan studi pustaka agar data menjadi valid dan dapat di pertanggung jawabkan. Waktu yang dibutuhkan untuk pengumpulan data adalah 4 (empat) minggu.

Setelah selesai dengan pengumpulan data. Maka, selanjutnya dilakukan pengembangan *hardware* dan pengembangan *software* sebagai inti dari penelitian ini. Yang dilakukan pertama adalah analisis masalah pada pengembangan *software* dimana disini memunculkan masalah-masalah yang ada setelah dilakukan wawancara dan mencari solusi yang tepat. Tahapan selanjutnya adalah analisis kebutuhan, analisis kebutuhan *software* dan *hardware* dilakukan bersamaan dengan tujuan penghematan waktu penelitian. Waktu yang dibutuhkan untuk analisis masalah dan analisis kebutuhan adalah 2 (dua) minggu.

Setelah selesai dengan analisis masalah dan analisis kebutuhan maka selanjutnya melakukan desain pada pengembangan *software* dan *hardware* yang akan menunjang keberlangsungan penelitian ini, jika pada *software* melakukan

desain tentang bagaimana nanti sistem akan berjalan sedangkan pada *hardware* menentukan bagaimana bentuk dari alat yang akan dibuat. Waktu yang dibutuhkan dalam proses ini adalah 2 (dua) minggu, waktu ini sampai benar-benar tidak ada kendala.

Setelah selesai dengan desain dan mengetahui bagaimana alat akan dibuat, maka dilanjutkan dengan konstruksi, dimana pada pengembangan *hardware* mulai membuat miniatur dari tangki air dan penampung sedangkan pada pengembangan *software* memulai dengan melakukan koding atau pengkodean. Proses ini berjalan selama 2 (dua) minggu hingga konstruksi benar-benar sesuai.

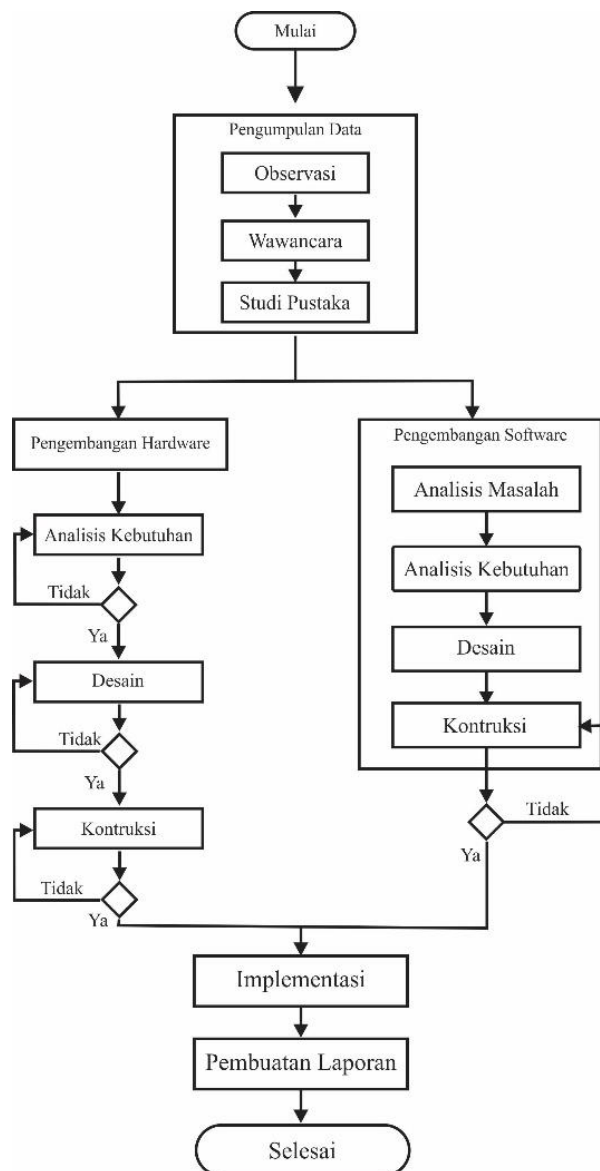
Tahapan selanjutnya adalah implementasi, pada tahapan ini mulai dengan melakukan berbagai uji coba apakah alat sudah berjalan sesuai dengan desain dan bisa mengatasi masalah yang sebelumnya telah dianalisis. Jika masih terdapat kesalahan maka akan kembali ke tahapan konstruksi.

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Pikir

Pada pembuatan *prototype monitoring* ketinggian air ini dibutuhkan beberapa tahapan yang harus dilakukan untuk dapat sebuah alat yang berguna dalam memonitor ketinggian air secara *realtime*. Untuk penjelasan skema penelitian terdapat pada halaman selanjutnya.



Gambar 3.1 Kerangka Pikir

3.2 Penjelasan Skema Penelitian

3.2.1 Pengumpulan Data

Pengumpulan data dimulai dari observasi, pada penelitian ini observasi dilakukan dengan melakukan berbagai percobaan pada alat yang akan digunakan sebagai penunjang penelitian. Percobaan pertama yang dilakukan yaitu pada sensor ultrasonik untuk memastikan bahwa sensor berfungsi dengan baik, percobaan berikutnya dilakukan pada LCD (*Liquid Crystal Display*) yang akan berguna untuk menampilkan hasil baca dari sensor ultrasonik, dilanjutkan dengan penambahan *buzzer* yang nantinya akan berguna sebagai alarm peringatan jika sensor ultrasonik membaca dan menunjukan pada ketinggian tertentu. Dari hasil observasi yang berupa hasil uji coba alat maka penulis mencoba untuk melakukan wawancara kepada narasumber terkait *monitoring* ketinggian air. Wawancara dilakukan kepada seorang teman yang mengerti tentang air dan kondisi air, *form* wawancara dapat dilihat dibagian lampiran.

Setelah dilakukan wawancara penulis mendapatkan kesimpulan sementara mengenai *monitoring* ketinggian air pada tangki. Penulis melakukan verifikasi dengan cara studi pustaka agar dapat mempermudah dalam pemecahan masalah dan mendapatkan referensi. Studi pustaka yang dilakukan yaitu mencari beberapa jurnal dan buku dari berbagai media yang masih berkaitan dengan *monitoring* ketinggian air. Beberapa jurnal dari penelitian sebelumnya yang digunakan sebagai referensi dalam penelitian ini tercantum di bab II.

3.2.2 Pengembangan Sistem

Pengembangan sistem pada penelitian ini dibagi menjadi 2 (dua), yaitu pengembangan perangkat lunak (*software*) dan pengembangan perangkat keras (*hardware*).

1. Pengembangan Perangkat Lunak

Pengembangan perangkat lunak ini dilakukan beberapa tahapan, diantaranya adalah analisis masalah, analisis kebutuhan, desain, dan konstruksi atau

pengkodean. Berikut penjelasan dari tahapan-tahapan pengembangan perangkat lunak:

a. Analisis Masalah

Pengembangan perangkat lunak dimulai dari analisis masalah, dimana pada penelitian ini masalahnya adalah:

1. Membutuhkan 2-3 orang tenaga SDM (Sumber Daya Manusia) yang digunakan untuk melakukan *monitoring* secara manual.
2. Masih belum akuratnya informasi yang didapatkan saat melakukan *monitoring*.
3. Membutuhkan waktu beberapa menit untuk mendapatkan hasil *monitoring* ketinggian air.

Melihat dari beberapa masalah yang terjadi pada *monitoring* ketinggian air, maka penulis melakukan beberapa batasan masalah diantaranya adalah:

1. Ketinggian air minimal 5cm dan ketinggian maksimal adalah 30cm.
2. Hanya berfokus pada *monitoring* ketinggian air.
3. Menggunakan NodeMCU ESP8266 sebagai *board* mikrokontroler.
4. Alat yang dibuat dalam bentuk miniatur atau *prototype*.
5. Melakukan komunikasi melalui Wi-Fi antara aplikasi Android dengan alat.
6. Hanya menginformasikan ketinggian secara jarak jauh.

Jika pada tahap ini masih memiliki kekurangan atau terdapat kesalahan maka akan kembali ke tahap sebelumnya hingga mendapatkan hasil yang tepat, kemudian baru melanjutkan ke tahapan selanjutnya. Setelah selesai dengan analisis masalah dilanjutkan dengan analisis kebutuhan yang akan dibangun pada alat *monitoring* ketinggian air ini.

b. Analisis Kebutuhan

Analisis kebutuhan digunakan sebagai penunjang dalam penelitian *monitoring* ketinggian air.

Berikut beberapa analisis kebutuhan yang digunakan untuk membangun alat *monitoring* ketinggian air pada tangki berbasis *internet of things*:

1. Alat harus bisa menunjukkan ketinggian permukaan air dengan batasan yang sebelumnya telah ditetapkan.
2. Harus bisa menginformasikan kepada *user* jika permukaan air telah mencapai titik tertentu.
3. Harus bisa membaca ketinggian air lalu menampilkan ketinggian air pada layar LCD dan menghidupkan *buzzer* sebagai *warning system*.

Kebutuhan lainnya yang akan menunjang dalam pengembangan perangkat lunak untuk alat *monitoring* ketinggian air adalah:

1. Laptop atau komputer.
2. Arduino IDE.
3. *Library* yang dibutuhkan.
4. Board NodeMCU ESP8266 v3

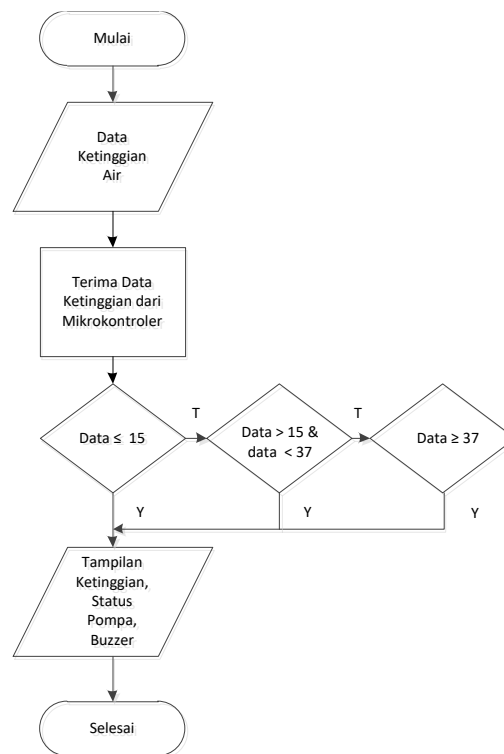
c. Alur Kerja Sistem

Setelah selesai dengan tahap analisis masalah dan analisis kebutuhan dilanjutkan dengan membuat sebuah alur kerja sistem. Alur kerja sistem pada pengembangan *software* sangat penting dilakukan karena akan menunjang bagaimana perangkat keras (*hardware*) bekerja. Perancangan *software* meliputi alur kerja sistem dan *interface* aplikasi android pada blynk dimana ini terdiri dari virtual monitor atau lcd dan virtual led. Dapat dilihat pada gambar 3.2 itu adalah gambar dari alur kerja sistem pada penelitian *monitoring* ketinggian air.

d. Kontruksi Software

Setelah tahap desain *software* selesai dilanjutkan dengan membuat kontruksi atau melakukan pengkodean (*coding*) dari pengembangan *software* yaitu dengan menggunakan Arduino IDE (*Integrated Development*

Environment) sebagai *tools* untuk melakukan pengkodean agar dapat di *upload* ke board NodeMCU ESP8266. Pengkodean yang dibuat telah sesuai dengan desain yang sebelumnya telah di sesuaikan dengan permasalahan dan kebutuhan.



Gambar 3.2 Alur Kerja Sistem

2. Pengembangan Perangkat Keras (*Hardware*)

Pengembangan perangkat keras ini dilakukan beberapa tahapan, diantaranya adalah analisis kebutuhan, desain, dan konstruksi. Berikut penjelasan dari beberapa tahapan pengembangan perangkat keras:

a. Analisis Kebutuhan

Yang dilakukan pertama untuk pengembangan perangkat keras yaitu dengan melakukan analisis kebutuhan, apa saja yang dibutuhkan dalam

melakukan pembuatan alat *monitoring* ketinggian air ini. Berikut beberapa kebutuhan perangkat keras yang menunjang penelitian ini:

1. Laptop / PC
2. Board NodeMCU ESP8266 v3
3. Modul Sensor Ultrasonik HC-SR04
4. Modul *Relay*
5. LCD (*Liquid Crystal Display*)
6. Buzzer
7. Kabel Jumper
8. Breadboard
9. Pompa Air
10. Ember sebagai tangki
11. Besi siku yang digunakan membuat penampang (tiang)
12. Adaptor DC 5V 2A

Setelah selesai dengan melakukan analisis kebutuhan jika *user* telah setuju dengan beberapa kebutuhan perangkat keras yang akan digunakan maka selanjutnya dilanjutkan dengan melakukan desain, tapi jika masih terdapat beberapa kebutuhan lainnya setelah dilakukan desain maka dapat kembali ke proses analisis kebutuhan.

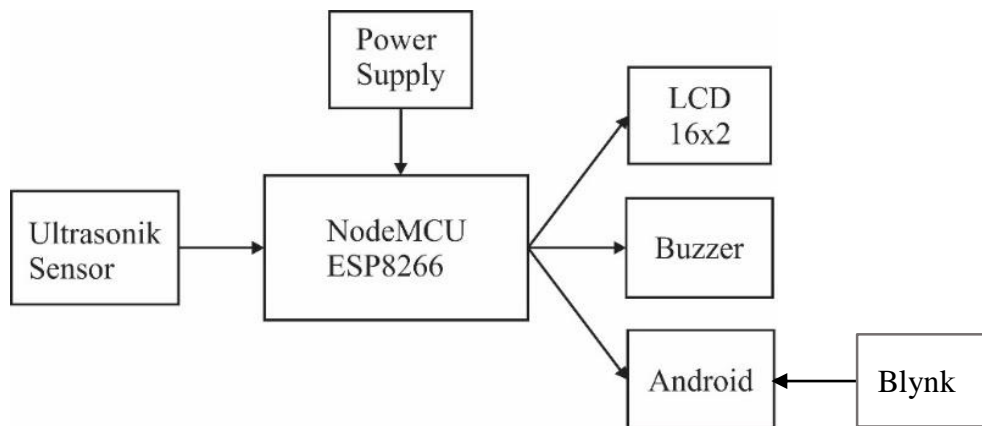
b. Desain

Setelah selesai dengan analisis kebutuhan dan sudah disetujui oleh *user* atau pengguna, selanjutnya melakukan tahap desain perangkat keras. Dimana pembuatan desain pada penelitian ini menggunakan sebuah *tools* menggambar yaitu CorelDraw. Desain digunakan untuk mengetahui tentang bagaimana gambaran awal dari konstruksi alat yang nantinya akan dibangun, dan merupakan penggunaan dari beberapa kebutuhan yang sebelumnya telah disebutkan.

Apabila pada proses ini masih terdapat kekurangan dan belum tepat maka akan kembali ke proses sebelumnya untuk diperbaiki sampai mendapatkan hasil yang tepat sebelum melanjutkan ke tahap berikutnya. Penjelasan selengkapnya terkait desain pada perangkat keras akan dibahas di bab IV.

c. Kontruksi

Selanjutnya adalah kontruksi dimana tahapan ini bisa dikerjakan setelah memiliki persetujuan dari *user* tentang desain yang sebelumnya di usungkan. Kontruksi pembuatan alat ini dapat digambarkan melalui diagram alur sebagai berikut:



Gambar 3.3 Diagram Blok Alur *Hardware*

Selain diagram alur, pada tahap kontruksi juga melakukan pembuatan perangkat pendukung seperti membuat penampang dan membuat miniatur tangki air sehingga menyerupai aslinya. Setelah selesai semua pada tahap kontruksi maka selanjutnya yaitu implementasi tetapi jika masih ada proses yang kurang dan belum tepat maka akan kembali ke tahapan desain untuk diperbaiki.

3.2.3 Implementasi

Tahapan selanjutnya adalah implementasi, tahapan ini dilakukan setelah proses pengembangan selesai, baik itu pengembangan *software* maupun pengembangan *hardware*. Pada tahap ini dilakukan beberapa uji coba untuk memastikan bahwa alat benar-benar bekerja sesuai dengan *code* atau kontruksi yang sebelumnya telah dibuat.

Beberapa kesalahan muncul pada saat proses pengujian:

1. Baca sensor yang salah karena permukaan tidak stabil (timbul selisih).
2. Dengan menggunakan adaptor 5v untuk catu daya PLN, pin VCC dan GND dari sensor ultrasonik harus dilepas.

Jika pada tahap ini masih terdapat kekurangan atau kesalahan maka akan kembali ke tahap kontruksi sampai mendapatkan hasil yang tepat. Kemudian dilanjutkan ke tahap selanjutnya. Tahapan berikutnya yaitu pelaporan.

3.2.4 Pembuatan Laporan

Tahapan terakhir adalah pembuatan laporan sebagai salah satu persyaratan kelulusan, pembuatan laporan ini terdiri dari hasil uji coba dan implementasi dalam bentuk rekapitulasi dan dalam bentuk laporan tertulis yang terdiri dari 6 bab dengan sistematika penulisan sebagai berikut:

BAB I: PENDAHULUAN

Pada bab ini membahas latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metode penelitian yang digunakan serta sistematika penulisan pada penelitian yang sedang di teliti.

BAB II: TINJAUAN PUSTAKA

Pada bab ini berisi beberapa landasan teori dari beberapa jurnal ilmiah yang dijadikan sebagai referensi penelitian yang berguna bagi penelitian *prototype monitoring* ketinggian air berbasis *Internet of Things* menggunakan blynk dan NodeMCU ESP8266 pada tangki, dan dasar teori yang merupakan teori penunjang dalam penelitian yaitu *prototype*, *monitoring*, ketinggian air, *Internet of Things*, NodeMCU ESP8266, tangki air, Arduino IDE dan beberapa teori tentang modul yang digunakan.

BAB III: METODOLOGI PENELITIAN

Pada bab ini berisikan kerangka pikir yang menyangkut metode penelitian yang digunakan, kebutuhan yang dipakai dalam penelitian, pengembangan *software*

dan *hardware* serta penjelasan dari skema penelitian dan disusun berdasarkan kajian teoritik yang sesuai dengan sistematika penulisan.

BAB IV: ANALISIS DAN PERANCANGAN

Pada bab ini berisi analisis dan perancangan, bab ini menjelaskan tentang telah melakukan sebuah analisa terhadap masalah dan mulai melakukan beberapa rancangan terkait pemecahan masalahnya serta melaporkan hasil analisis yang didapatkan.

BAB V: IMPLEMENTASI DAN PENGUJIAN

Pada bab berisi impementasi dan pengujian dari hasil pemecahan masalah dan perancangan yang didapat secara teoritis serta implementasi secara menyeluruh pada alat yang dibangun dimana pada bab ini menampilkan gambar hasil dari uji coba.

BAB VI: KESIMPULAN DAN SARAN

Pada Bab ini berisi kesimpulan yang diperoleh dari bab-bab sebelumnya dan saran-saran dari hasil pengamatan. Kesimpulan berisi dari jawaban permasalahan yang sebelumnya terjadi serta memberikan beberapa saran bagi peneliti selanjutnya jika ingin melanjutkan penelitian ini.

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Analisis kegiatan awal pada pembangunan dan pengembangan penelitian ini menggunakan metode uji coba atau ekperimental terhadap penggunaan teknologi IoT (*Internet of Things*) pada alat *monitoring*. Dalam analisis ini penulis mempunyai instrumen penelitian dan melakukan beberapa tahapan yaitu, analisis kebutuhan, analisis pengguna, *user interface*, fitur-fitur dan hasil analisis.

4.1.1 Instrumen Penelitian

Instrumen penelitian pada penelitian kali ini terdiri dari perangkat keras, perangkat lunak dan hasil ulasan saat melakukan wawancara. Berikut penjabaran yang menjadi instrumen penelitian:

a. Perangkat Lunak

Penulis menggunakan beberapa perangkat lunak yang menunjang dalam penelitian ini:

1. Menggunakan Sistem Operasi Windows 10 Profesional 64bit.
2. *Code Editor* menggunakan Arduino IDE yang bisa melakukan *Edit*, *Compile*, dan *Upload* pada Board Mikrokontroler yaitu NodeMCU ESP8266.
3. Menggunakan Fritzing sebagai penggambaran skema rangkaian elektronika.

b. Perangkat Keras

Berikut adalah beberapa perangkat keras yang menjadi penunjang dalam instrumen penelitian pada penelitian ini:

Tabel 4.1 Instrumen Penelitian Perangkat Keras *Monitoring*
Ketinggian Air

Perangkat Keras Laptop

Tipe Laptop	Asus X44H Series
Prosesor	Intel Pentium CB950 2.1Ghz
RAM	2GB
HDD	500GB

Perangkat Keras Smartphone

<i>Smartphone</i>	Samsung J5 Prime
OS	Android Oreo
RAM	2GB
Penyimpanan	16GB

c. Wawancara

Wawancara dilakukan kepada salah satu rekan yang bekerja di perusahaan PDAM dan mengetahui dengan jelas bagaimana *monitoring* ketinggian air dilakukan.

Menurut pemaparannya *monitoring* ketinggian air belum ada yang menggunakan *smartphone* atau menggunakan teknologi IoT dan belum dapat diketahui ketinggiannya karena saat ini baik dikantor maupun warga atau masyarakat lebih memilih menggunakan bandul (*switch*) karena dianggap lebih mudah dan simpel dan dengan menggunakan bandul maka air dalam tangki akan tetap penuh.

4.1.2 Analisis Sistem

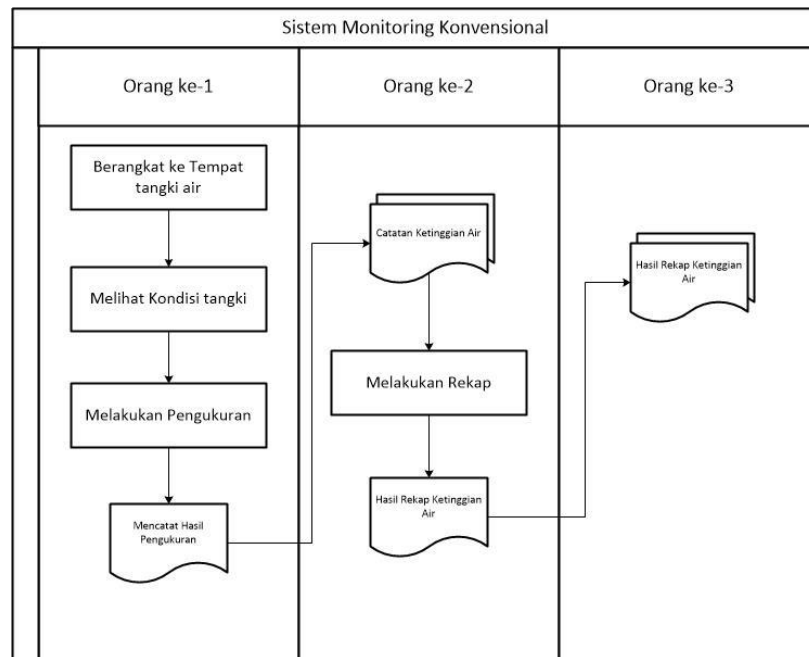
Sistem yang berjalan saat ini dalam kendali otomatisasi kontrol ketinggian air masih bersifat konvensional yaitu dengan mendatangi dan melihat tempat penyimpanan tangki air secara langsung atau bisa juga dengan menggunakan bandul (*switch*) pompa otomatis. Cara konvensional

dengan mendatangi dan melihat secara langsung memang mudah dilakukan dan sangat ringan pada segi biaya namun hal ini akan terkendala jika cuaca buruk atau tempat penyimpanan tangki berada di tempat yang tinggi serta akan membutuhkan waktu yang cukup lama juga membutuhkan tenaga yang banyak.

Sedangkan bandul ini merupakan alat yang saat ini banyak digunakan oleh masyarakat karena hanya dengan menggunakan bandul ini saja tidak perlu khawatir air akan tumpah karena jika air sudah menyentuh salah satu permukaan bandul maka pompa air akan otomatis mati. Namun bandul ini masih memiliki kelemahan yaitu diantaranya adalah listrik menjadi boros karena jika air sedikit saja kosong maka pompa akan menyala, masa aktif bandul relatif singkat karena kurangnya pengontrolan dan akan berpengaruh ke masa aktif pompa karena jika aliran air macet pompa akan terus menyala tapi air tidak mengalir, jika terjadi kendala masih harus melakukan pengecekan secara manual serta tidak dapat diketahui berapa tinggi air yang ada didalam tangki dan tidak bersifat *realtime* serta tidak memiliki alarm.

Berdasarkan dari pemaparan permasalahan yang telah dipaparkan diatas, maka penulis berupaya melakukan uji coba dengan menggunakan teknologi IoT sebagai solusi untuk permasalahan tersebut. Dimana jika menggunakan teknologi ini memungkinkan untuk mengetahui kondisi air secara *realtime*, dapat diakses secara jarak jauh dan bisa diakses menggunakan Internet dan memiliki alarm dari bunyi yang dihasilkan oleh buzzer serta memungkinkan mengetahui nilai volume air pada tangki tersebut.

Berikut merupakan sistem *monitoring* konvensional yang sebelumnya berjalan sistem ini merupakan sistem pertama sebelum adanya sistem bandul dan sistem yang menggunakan teknologi IoT:



Gambar 4.1 *Flowmap* Sistem Monitoring Konvensional

4.1.3 Analisis Kebutuhan

Analisis kebutuhan pada penelitian kali ini dibagi menjadi 2 (dua) yaitu kebutuhan *software* (perangkat lunak) dan kebutuhan *hardware* (perangkat keras). Dimana kebutuhan perangkat keras berisikan beberapa alat dan bahan yang digunakan untuk menunjang penelitian.

1. Kebutuhan Internet

Analisis kebutuhan internet dilakukan untuk mengetahui berapa total penggunaan yang dihabiskan selama penelitian ini berlangsung:

Tabel 4.2 Kebutuhan Internet

Data	5GB
Kecepatan Akses	20Kbps
Lama Penggunaan	30 hari
Biaya	Rp. 36.000

Kebutuhan penggunaan internet pada penelitian ini menggunakan data dari salah satu provider yang menawarkan harga yang cukup terjangkau dan memiliki data yang cukup untuk memenuhi kebutuhan internet alat selama satu bulan. Data yang ditawarkan sebanyak 5GB dengan kecepatan sampai dengan 2Mbps dengan data sebanyak ini akan cukup selama satu bulan karena penggunaan internet pada alat *monitoring* tidak akan memakan terlalu banyak.

2. Kebutuhan *Software* (Perangkat Lunak)

Analisis kebutuhan *software* ini dilakukan untuk mengetahui *software* apa aja yang digunakan. Berikut *software* yang akan digunakan dalam penelitian ini antara lain:

1. Arduino IDE digunakan sebagai *compiler* yang berguna untuk menulis, mengedit dan meng-upload sketch program yang berguna sebagai fungsi utama dari pembuatan *monitoring* ketinggian air ini.
2. Library board NodeMCU ESP8266 disini digunakan agar board NodeMCU ESP8266 bisa terbaca dan digunakan di Arduino IDE, karena pada dasarnya board *default* Arduino IDE adalah board Arduino Uno.
3. Library Sensor Ultrasonik digunakan agar sensor ultrasonik dapat berjalan dengan baik ketika sensor terhubung dengan board NodeMCU ESP8266. Library ini dapat berjalan ketika dimasukan kedalam Arduino IDE dan dituliskan dalam sketch program.
4. Library LCD (*Liquid Crystal Display*) digunakan agar LCD dapat berfungsi dengan baik ketika terhubung dengan board NodeMCU ESP8266. Library ini di-*input*-kan kedalam Arduino IDE dan dimasukan kedalam sketch program yang sedang dibangun.

5. Library Aplikasi Blynk digunakan supaya aplikasi blynk saat terhubung dengan board NodeMCU ESP8266 dapat berjalan dengan baik. Library nya sendiri didapatkan di situs github yang menyediakan library yang dapat berjalan di Arduino IDE.
6. Aplikasi *smartphone* Blynk digunakan sebagai media komunikasi dengan Board NodeMCU ESP8266. Aplikasi ini akan menjadi *output* jarak jauh yang terhubung melalui internet. Sumber internet saat ini menggunakan *hotspot* dari *smartphone*.
7. Fritzing digunakan untuk membuat skema rangkaian elektronika pada tiap modul yang digunakan pada penelitian ini. Fritzing sangat membantu dalam memberikan gambaran dalam menghubungkan rangkaian elektronika.

3. Kebutuhan *Hardware* (Perangkat Keras)

Analisis kebutuhan *hardware* yang dilakukan pada penelitian *monitoring* ketinggian air. Berikut beberapa *hardware* yang akan digunakan dalam penelitian ini:

1. Board NodeMCU ESP8266 v3 digunakan karena merupakan board yang jauh lebih baik dari versi sebelumnya serta memiliki kapasitas memori yang besar. Board NodeMCU ini memiliki beberapa pin yang terhubung dengan berbagai komponen elektronika lainnya yaitu Sensor Ultrasonik, LCD dan buzzer serta Wi-Fi yang terhubung dengan *Smartphone*. Adapun pin yang terhubung dengan sensor ultrasonik yaitu pin D3, D4, VU sebagai input VCC dan GND sebagai ground. NodeMCU yang terhubung dengan LCD yaitu pin VU, GND dan D1, D2 selanjutnya adalah NodeMCU yang terhubung dengan buzzer adalah pin GND dan pin D5.

Sedangkan Wi-Fi berguna untuk komunikasi antara board NodeMCU dan *Smartphone*.

2. Modul Sensor Ultrasonik HC-SR04 digunakan kerana jarak sensor ini memiliki *range* yang bisa mencapai lebih dari 2 meter serta sudah memiliki *transmitter* dan *receiver* yang berfungsi untuk mendapatkan hasil yang lebih baik dibandingkan dengan sensor-sensor jarak lainnya. Sensor ini terhubung dengan NodeMCU melalui pin VU akan terhubung dengan pin VCC pada sensor, GND akan terhubung dengan GND sebagai sumber arus dan D3, D4 sebagai pin trigger dan pin echo pada sensor ultrasonik.
3. LCD (*Liquid Crystal Display*) digunakan untuk menampilkan hasil baca dari sensor ultrasonik dan bagaimana kondisi tangki air. LCD yang digunakan adalah LCD I2C 16x2 yaitu LCD yang memiliki 16 kolom 2 baris, penggunaan LCD jenis ini karena disesuaikan dengan kebutuhan *output* yang akan menampilkan hasil baca sensor dan kondisi tangki air. LCD sendiri terhubung dengan board NodeMCU melalui pin VU dan GND sebagai sumber arus, dan pin D3 dan D4 dihubungkan dengan SDA dan SCL pada LCD.
4. Buzzer disini digunakan sebagai alarm. Penggunaan buzzer akan berguna karena akan memberitahukan kondisi tangki air terhadap ketinggian air. Karena jika kondisi ketinggian air terhadap sensor menunjukan kondisi air penuh buzzer akan berbunyi secara hidup-mati dan jika air kosong maka buzzer akan berbunyi secara terus menerus namun jika kondisi air cukup maka buzzer pun akan mati. Buzzer terhubung dengan NodeMCU melalui pin GND dan pin D5.

5. Modul relay 1 *channel* digunakan untuk saklar penghubung dengan pompa air yang akan menyesuaikan dengan *sketch* yang telah dibuat sebelumnya. Memilih relay 1 *channel* karena hanya akan mengontrol 1 (satu) pompa saja. Relay 1 *channel* ini memiliki 3 pin yaitu VCC, GND, dan IN terhubung dengan NodeMCU melalui pin VU, GND, dan D6. Relay ini juga yang mengatur ON atau OFF dari pompa air yang akan berhenti sesuai *sketch* program yang telah dibuat.
6. Kabel jumper digunakan karena menjadi penghubung antara board NodeMCU dengan komponen lainnya agar dapat berjalan sesuai dengan *sketch* program yang sebelumnya telah dibuat. Kabel jumper akan dihubungkan dengan disesuaikan antara pin pada NodeMCU dan pin pada tiap modul atau komponen elektronika yang digunakan. Kabel jumper dipilih karena kabel khusus yang digunakan pada proyek mikrokontroler. Penggunaan kabel jumper ini sangat banyak karena kabel jumper yang rentan putus.
7. Breadboard digunakan sebagai penghubung selain kabel jumper antara board NodeMCU ESP8266, sensor ultrasonik, LCD, buzzer dan Relay 1 *channel*. Breadboard akan sangat membantu karena sifatnya mirip pcb dan bisa membuat satu pin bisa terhubung dengan beberapa pin. Menggunakan breadboard karena banyaknya pin VCC yang terhubung dengan pin VU.
8. Adaptor DC 5v digunakan sebagai sumber arus NodeMCU ESP8266 yang akan terhubung langsung dengan sumber arus listrik PLN. Penggunaan adaptor ini karena NodeMCU hanya mampu menerima arus sebesar 5-9 volt saja. Adaptor yang digunakan adalah 5v 2A agar memberikan tegangan yang cukup bagi board.

9. Pompa air digunakan sebagai sarana memindahkan air dari satu tangki sumber ke tangki lainnya. Pompa air terhubung dengan relay 1 *channel* yang berfungsi sebagai saklar otomatis yang terhubung dengan board NodeMCU tetapi pompa air ini masih menggunakan sumber tegangan dari luar tidak langsung dari NodeMCU ESP8266.
10. Tangki Air digunakan sebagai penampung utama pada penelitian ini, pompa air yang digunakan memiliki diameter sekitar 25cm dengan tinggi dari penutup sampai dasar adalah 44cm. Tangki yang digunakan merupakan miniature dari tangki aslinya dengan perbandingan tinggi 1:8 dari aslinya.

Dibawah ini adalah kebutuhan perangkat keras yang digunakan pada penelitian *monitoring* ketinggian air yang akan dibuat *prototype*:

Tabel 4.3 Kebutuhan Alat dan Bahan

No	Nama	Jumlah
1	Board NodeMCU ESP8266 v3	1
2	Modul Sensor Ultrasonik HC-SR04	1
3	Modul Relay 1 Channel	1
4	LCD (<i>Liquid Crystal Display</i>)	1
5	Buzzer	1
6	Pompa Air	1
7	Kabel Jumper 30cm	40
8	Breadboard	1
9	Tangki Air	1
10	Adaptor DC 5v	1
11	Penyangga Tangki	1

4. Kebutuhan Pengguna

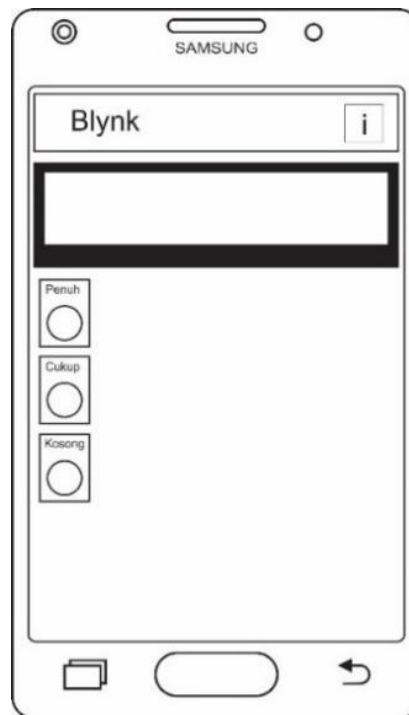
Analisis kebutuhan pengguna dilakukan agar dapat mengetahui siapa saja yang dapat mengakses dan menggunakan alat *monitoring* yang akan dibuat. Pengguna ini harus dapat melihat, mengontrol kondisi air secara jarak jauh. Pengguna merupakan *user* general yang memiliki kemampuan dalam mengoperasikan gawai berbasis android, memiliki pengetahuan tentang perangkat keras yang dibuat, serta mampu melakukan perawatan terhadap alat jika terjadi kesalahan atau kerusakan yang berakibat alat menjadi tidak berfungsi dengan baik.

Target pengguna dalam pemanfaatan alat ini ditujukan kepada pihak-pihak atau instansi yang mengharuskan mereka mengontrol ketersediaan air dalam jumlah banyak. Alat ini akan berguna untuk mengetahui ketersediaan air untuk berbagai kebutuhan. Pengguna yang bisa memanfaatkan alat ini bisa pabrik-pabrik yang menggunakan air dalam proses produksinya, dimana alat ini akan berfungsi untuk mengontrol ketersediaan air dan mengetahui berapa jumlah atau kapasitas air di dalam tangki. Selain pabrik bisa juga digunakan oleh instansi PDAM dikarenakan PDAM merupakan perusahaan air yang harus selalu mengontrol ketersediaan air untuk memasok kepada masyarakat.

4.1.4 *User Interface*

User Interface dari alat ini adalah tampilan dari aplikasi blynk yang menjadi *output* pada penelitian ini dan telah disesuaikan dengan beberapa kebutuhan dalam penelitian. *User interface* akan dibuat begitu sederhana sehingga memudahkan *user* ketika mengoperasikan aplikasi blynk yang terhubung dengan alat *monitoring* air. Dalam *user interface* ini nantinya akan menampilkan ketinggian air, volume air dan lampu indikator yang menandakan kondisi air.

Berikut adalah tampilan *user interface* dari aplikasi blynk:



Gambar 4.2 *User Interface* Aplikasi Blynk

4.1.6 Fitur-fitur

Beberapa fitur yang akan hadir pada pembuatan alat *monitoring* ketinggian air berbasis IoT ini bertujuan untuk memudahkan *pengguna*. Berikut adalah beberapa fitur pada alat *monitoring* ketinggian air:

- a. Mengetahui ketinggian air secara *realtime* yang dapat dimonitor dimana saja selama masih terhubung dengan internet.
- b. Memberikan tanda peringatan dengan menyalakan bunyi yang terdapat pada buzzer.
- c. Menampilkan status ketinggian air pada LCD (*Liquid Crystal Display*).
- d. Mampu menampilkan ketinggian, volume air dan menyalakan lampu indikator kondisi air pada *user interface* aplikasi Blynk.

4.1.7 Hasil Analisis

Hasil analisis yang didapatkan pada penelitian ini yaitu alat dapat melakukan *monitoring* ketinggian air secara jarak jauh menggunakan internet yang pada penelitian ini menggunakan *hotspot* dari *smartphone*, *monitoring* akan dilakukan secara *realtime* tetapi hanya berlaku untuk data saat itu saja, mampu memberikan peringatan dengan menyalakan buzzer serta alat mampu menampilkan ketinggian air, volume air pada *interface* aplikasi blynk yang telah terinstal di *smartphone*. Untuk menunjang tercaiapainya penelitian ini maka menggunakan beberapa modul elektronika seperti sensor ultrasonik, LCD, buzzer, breadboard, kabel jumper dan board NodeMCU ESP8266 v3 sebagai mikrokontroler.

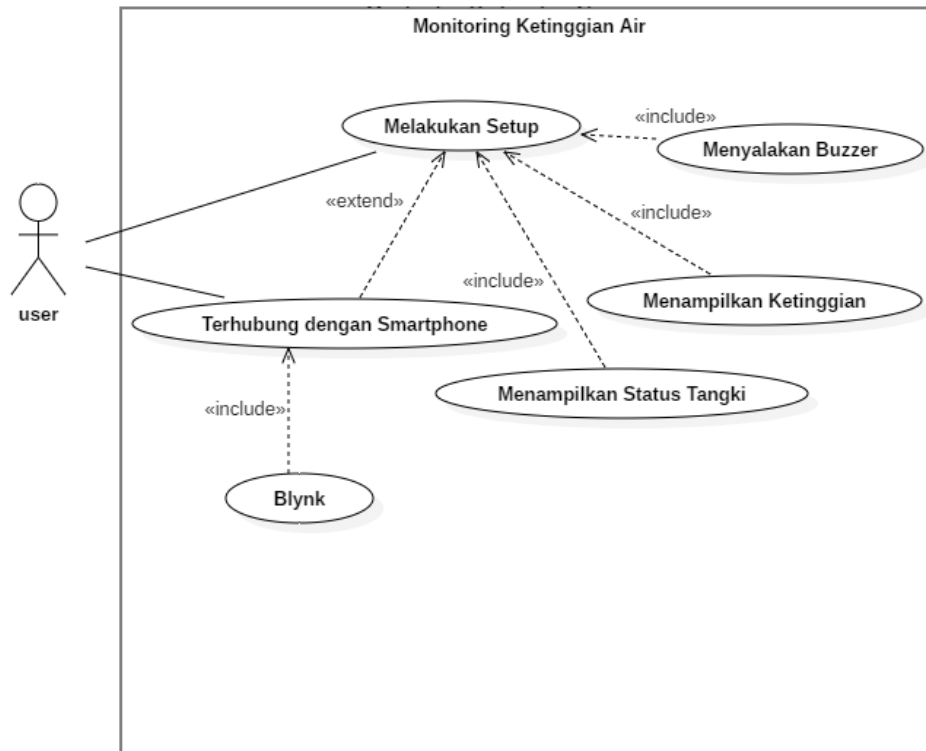
4.2 Perancangan

Proses perancangan ini juga dibagi menjadi beberapa perancangan yaitu perancangan *hardware* dan perancangan *software* serta memberikan gambaran terhadap *usecase diagram* dan *activity diagram* yang terjadi pada penelitian ini. Tahap perancangan *hardware* meliputi diagram blok sistem dan perangkaian alat. Sementara perancangan *software* adalah proses menanamkan kecerdasan melalui sketch program yang di-*upload* dalam sebuah board NodeMCU dengan mikrokontroler ESP8266 yang terdapat pada board NodeMCU.

4.2.1 Diagram Arus Data (DAD) – UML

1. Usecase Diagram Monitoring Ketinggian Air

Usecase diagram pada penelitian ini berguna untuk mendeskripsikan interaksi aktor dengan sistem *monitoring* ketinggian air yang akan dibangun. *Usecase* ini juga berguna untuk mengetahui apa saja yang dapat *user* atau pengguna lakukan terhadap sistem. Berikut ini merupakan *usecase diagram* dari perancangan pembangunan alat *monitoring* ketinggian air:



Gambar 4.3 *Usecase Diagram Monitoring Ketinggian Air*

Adapun penjelasan dari *usecase diagram monitoring ketinggian air* diatas adalah sebagai berikut:

a. Deskripsi Aktor

Deskripsi actor pada *usecase diagram* berfungsi untuk menjelaskan bagaimana peran aktor yang terdapat pada *usecase diagram*. Berikut ini adalah deskripsi aktor dari *usecase diagram* diatas:

Tabel 4.4 Tabel Deskripsi Aktor

Aktor	Deskripsi
Pengguna (<i>User</i>)	Pengguna disini merupakan orang yang mampu melakukan setup terhadap alat, mengoperasikan <i>smartphone</i>

	android, mampu membaca ketinggian air melalui <i>smartphone</i> , dan mampu melakukan perawatan pada alat jika terjadi kerusakan atau terjadi kendala.
--	--

b. Deskripsi *Usecase Monitoring Ketinggian Air*

Deskripsi *usecase* ini bertujuan untuk mengetahui proses apa saja yang dilakukan setiap *usecase* yang ada pada sistem *monitoring ketinggian air*. Berikut ini adalah deskripsi *usecase* sistem *monitoring ketinggian air*:

Tabel 4.5 Deskripsi *Usecase Monitoring Ketinggian Air*

No	<i>Usecase</i>	Deskripsi
1	Melakukan Setup	Proses dimana pengguna melakukan persiapan hingga alat benar-benar dapat berfungsi dengan baik
2	Terhubung dengan Smartphone	Proses dimana pengguna mencoba menghubungkan <i>smartphone</i> dengan alat <i>monitoring</i> yang nanti akan menjadi output dari aplikasi blynk. Dimana aplikasi ini akan menampilkan ketinggian air dan volume air pada LCD virtual.
3	Menampilkan Ketinggian Air	Proses menampilkan ketinggian air dari hasil baca sensor pada LCD jika alat monitoring sudah bekerja dengan baik.

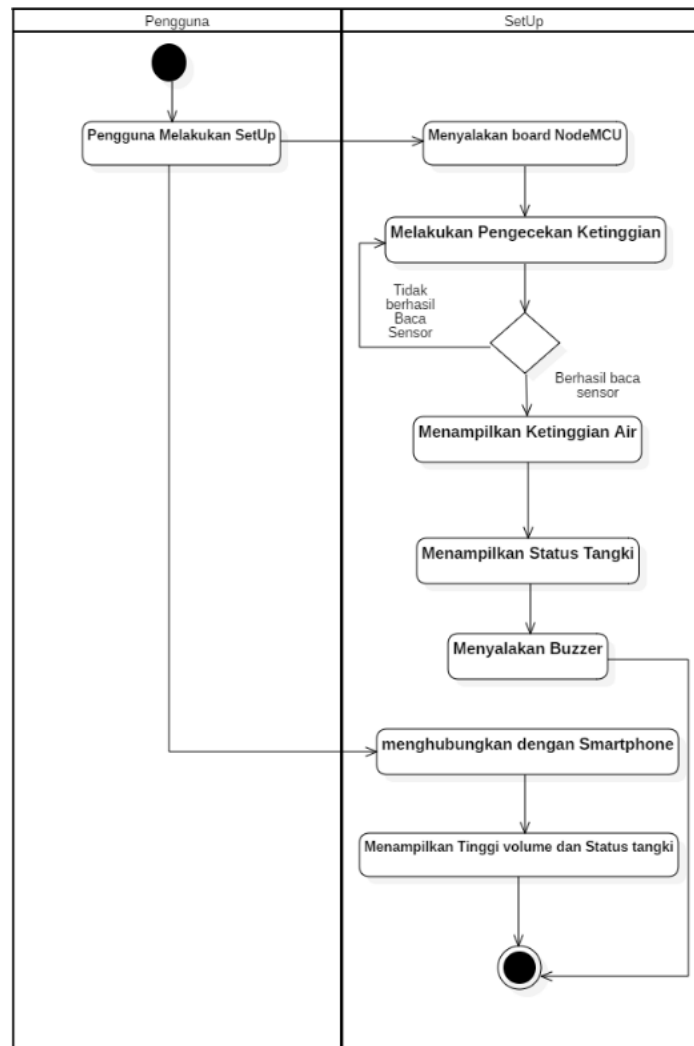
4	Menampilkan Status Tangki	Proses menampilkan status tangki dari hasil baca sensor pada LCD jika alat monitoring sudah bekerja dengan baik.
5	Menyalakan Buzzer	Proses memberikan peringatan pada pengguna pada saat status tangki dan ketinggian air berada pada titik tertentu.

2. Activity Diagram

Berikut *activity diagram* dalam alat *monitoring* ketinggian air yang menunjukkan aktifitas pengguna dengan sistem (mikrokontroler) dan timbal balik sistem tersebut pada pengguna secara sistematis:

1. Activity Melakukan Setup

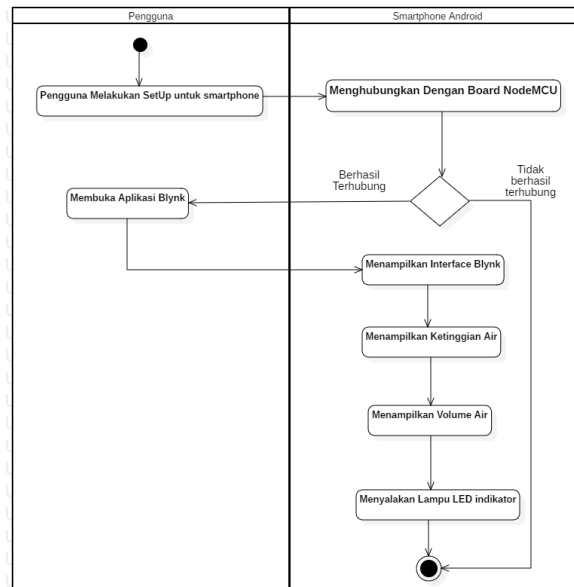
Activity melakukan setup ditunjukkan oleh gambar dibawah. Pertama pengguna melakukan persiapan (*Setup*) terhadap board NodeMCU dan persiapan terhadap *smartphone*. Setelah selesai dengan persiapan maka board akan mulai membaca ketinggian jika tidak berhasil maka sensor akan terus membaca sampai mampu menampilkan ketinggian air, status tangki dan menyalakan buzzer. Ini juga berlaku untuk tampilan pada *smartphone* setelah sensor berhasil membaca jarak maka akan menampilkan volume dan ketinggian serta menyalakan lampu LED indikator.



Gambar 4.4 Activity Melakukan SetUp

2. Activity Smartphone Android dan Blynk

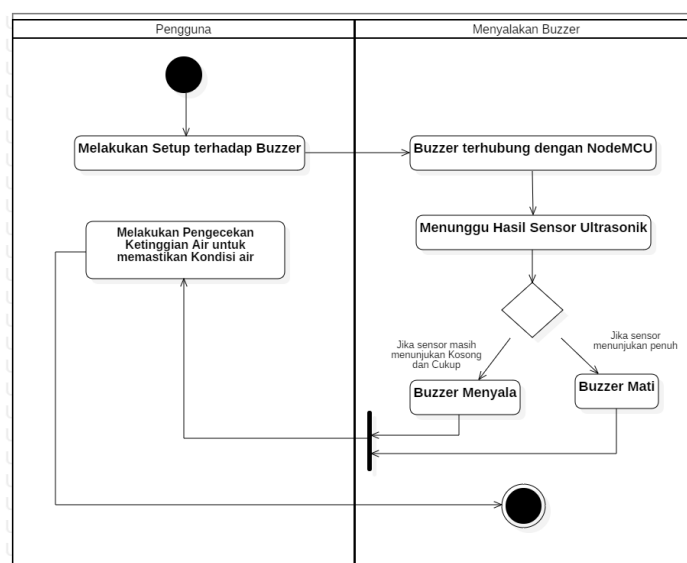
Activity smartphone dan aplikasi blynk, ditunjukkan oleh gambar dibawah. Pertama kali yang dilakukan adalah pengguna melakukan persiapan (*setup*) terhadap *smartphone* yang akan digunakan. Selanjutnya *smartphone* akan terhubung dengan board NodeMCU melalui wi-fi namun jika tidak berhasil terhubung maka langsung berakhir dan jika berhasil pengguna membuka aplikasi blynk dan menampilkan *interface* blynk yang mampu menampilkan ketinggian, volume dan menyalakan lampu LED indikator.



Gambar 4.5 Activity Smartphone Android dan Blynk

3. Activity Menyalakan Buzzer

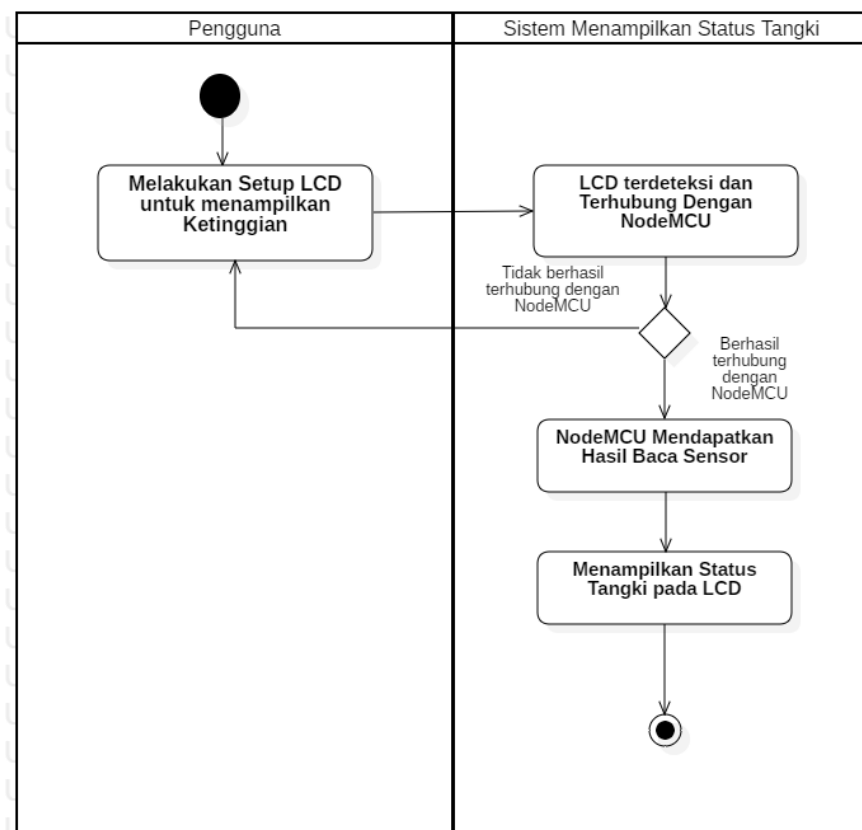
Activity menyalakan buzzer dapat dilihat pada gambar dibawah. Pertama kali pengguna melakukan *setup* terhadap buzzer sehingga buzzer terhubung dengan board NodeMCU. Setelah buzzer berhasil terhubung selanjutnya buzzer menunggu hasil baca sensor jika sensor mencapai titik tertentu dan buzzer mati atau hidup pengguna akan segera melakukan pengecekan terhadap alat apakah terjadi *error* pada pompa atau tangki.



Gambar 4.6 Activity Menyalakan Buzzer

4. Activity Menampilkan Status Tangki

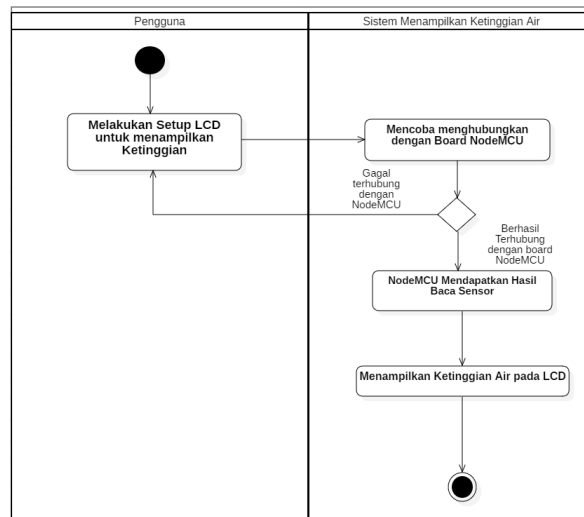
Activity untuk menampilkan status tangki dapat dilihat pada gambar dibawah. Pertama pengguna melakukan *setup* terhadap LCD yang akan menjadi media penampil status tangki sehingga dapat terhubung dengan board NodeMCU, pengguna akan melakukan *setup* lagi jika masih belum terhubung dengan board. Setelah terhubung board NodeMCU akan mendapatkan hasil baca sensor dan menampilkannya pada LCD.



Gambar 4.7 Activity Menampilkan Status Tangki

5. Activity Menampilkan Ketinggian Air

Activity menampilkan ketinggian air prosesnya sama saja dengan menampilkan status tangki air hanya berbeda pada apa yang ditampilkan pada LCD, yang ditampilkan disini yaitu ketinggian air dari hasil baca sensor yang sebelumnya telah berhasil.



Gambar 4.8 Activity Menampilkan Ketinggian Air

4.2.2 Perancangan Perangkat Keras (*Hardware*)

Pada perancangan perangkat keras (*Hardware*) berisi diagram blok sistem, rangkaian alat, dan flowchart sistem. Yang masing-masing akan dijelaskan secara lebih rinci.

1. Diagram Blok Sistem

Diagram blok sistem adalah diagram alur utama sistem yang menggambarkan struktur dari perancangan dan pembuatan alat secara keseluruhan. Adapun diagram blok sistem alat ini adalah sebagai berikut:

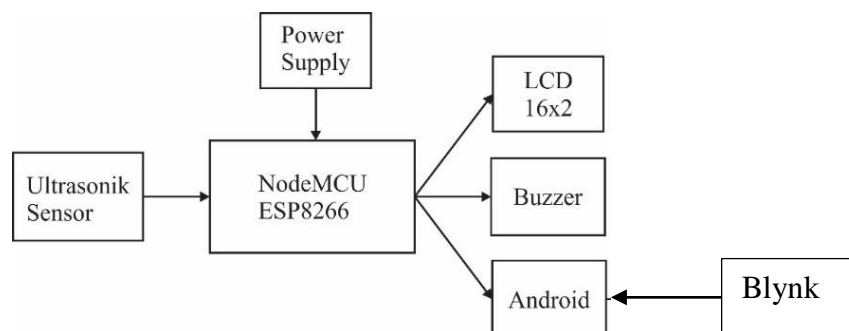
Gambar 4.9 Diagram Blok Sistem *Monitoring* Ketinggian Air

Diagram blok sistem ini menjelaskan tentang bagaimana kerja sistem *monitoring* ketinggian air secara keseluruhan. Sensor Ultrasonik akan membaca jarak antara sensor dengan permukaan air di depannya, semakin kecil angka yang dihasilkan maka semakin dekat jarak benda dengan sensor dan sebaliknya.

Berikut merupakan penjelasan dari diagram blok sistem pada peneliti ini:

- a. Sensor Ultrasonik berfungsi sebagai pembaca ketinggian air yang terhubung ke board NodeMCU melalui PIN D3 dan D4 serta dengan daya dari PIN DC dan GND (ground).
- b. NodeMCU ESP8266 berfungsi sebagai pengendali semua modul yang digunakan pada penelitian ini. NodeMCU ESP8266 akan mengendalikan modul sesuai sketch program yang telah di-*upload* melalui Arduino IDE.
- c. LCD berfungsi sebagai media yang menampilkan data hasil dari sensor Ultrasonik melalui NodeMCU ESP8266.
- d. Buzzer akan menyala terus-menerus apabila jarak antara sensor ultrasonik dan permukaan air $\geq 37\text{cm}$, buzzer akan menyala hidup-mati jika jarak sensor dan permukaan air antara 15-37cm dan buzzer akan mati ketika jarak sensor dan permukaan air $\leq 15\text{ cm}$.
- e. Fungsi Buzzer yaitu sebagai alarm peringatan pada kondisi permukaan air. Jika buzzer mati maka itu bertanda bahwa air telah penuh.
- f. Android merupakan perantara media komunikasi dengan board NodeMCU ESP8266.
- g. Aplikasi android bernama blynk akan menjadi *output* dalam *monitoring* ketinggian air ini. Pada aplikasi ini juga akan ditampilkan jarak ketinggian air, volume air, dan menyalakan led sebagai indikator pada ketinggian air. Agar terhubung dengan blynk maka *smartphone* android harus terhubung dengan internet yang sebelumnya telah disediakan sebagai suatu kebutuhan.

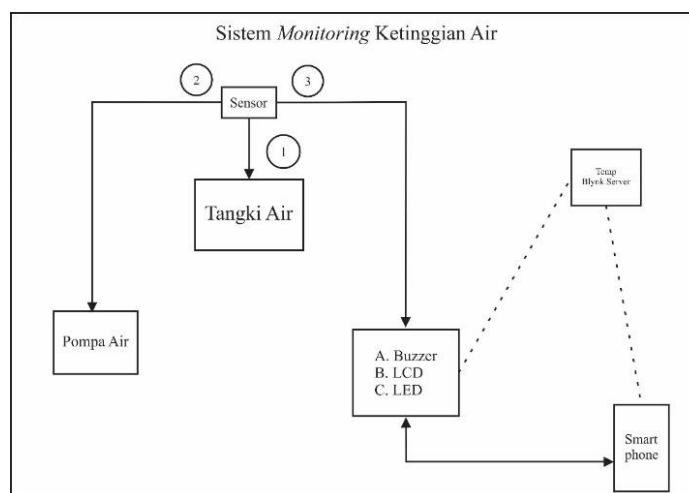
2. Perancangan Rangkaian Alat

Tahap ini meliputi proses perangkaian alat secara keseluruhan yang terdiri dari sensor Ultrasonik, board NodeMCU ESP8266, LCD, Buzzer, koneksi dengan Android menggunakan Internet dengan *range* tertentu. Rangkaian di desain dan dipasang pada masing-masing komponen sesuai dengan objek penelitian yang sedang dilakukan. Pada tahap perancangan ini akan dijelaskan mengenai rangkaian alat dan *sketch* program yang dibutuhkan sehingga dapat menjalankan alat yang sedang dibangun mulai dari *input*, proses, dan *output*.

Berikut ini merupakan sketsa dan rangkaian dari alat *monitoring* ketinggian air yang akan dibangun pada penelitian ini:

- a. Skema keseluruhan dari sistem alat *monitoring* ketinggian air.

Dibawah ini merupakan gambaran dari sistem *monitoring* ketinggian air yang akan dibangun, gambar ini menggambarkan bagaimana cara kerja alat secara keseluruhan, dimana pada gambar ini terlihat bagaimana sistem akan berjalan. Berikut adalah gambaran sistem *monitoring* ketinggian air:

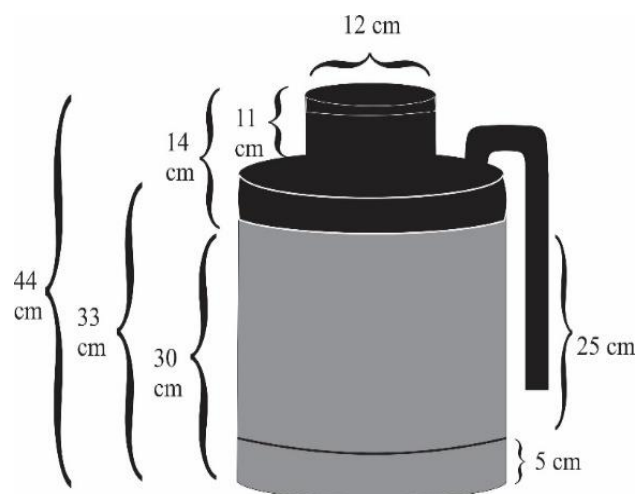


Gambar 4.10 Skema Keseluruhan Sistem *Monitoring* Ketinggian Air

Gambar diatas adalah skema dari sistem *monitoring* ketinggian air dimana sensor pertama kali akan membaca permukaan air yang ada pada tangki, jika sensor membaca ketinggian air yang menunjukkan bahwa tangki harus diisi maka pompa akan menyala selanjutnya akan menyalakan buzzer dan menampilkan kondisi tangki air pada LCD, dan menyalakan LED indikator yang ada pada *smartphone* melalui aplikasi blynk, sebelum tampil pada aplikasi blynk yang ada di *smartphone* sistem akan mengirimkan data ke server lalu server mengirimkan data tersebut ke *smartphone* melalui jaringan internet. Server yang digunakan adalah blynk server yang sudah disediakan oleh pihak blynk. Untuk menjalankan IoT maka harus dipastikan bahwa ada koneksi internet dan terhubung dengan alat *monitoring* ketinggian ini. Proses kerjanya ditunjukkan oleh nomor yang ada pada gambar.

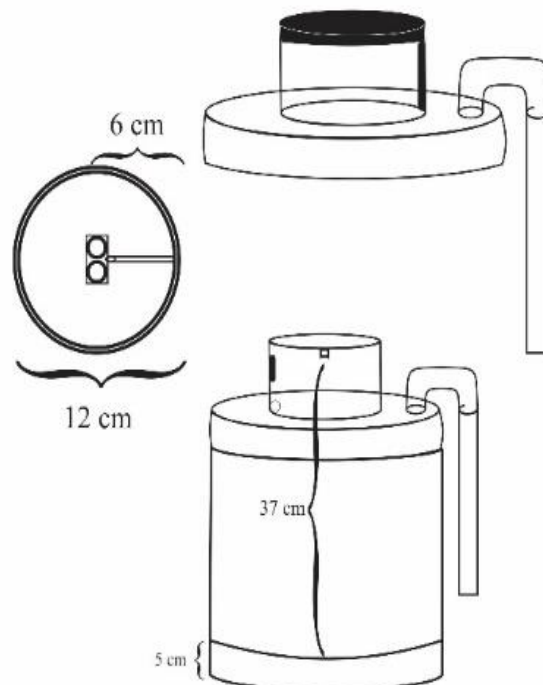
b. Sketsa rangkaian alat penampung air

Dibawah ini sketsa penampung air yang dirancang dan akan dibangun pada penelitian kali ini, dimana ketinggian dari tutup sampai ke dasar mencapai 44cm dan jarak dari tutup ke titik penuh tangki adalah 15cm.



Gambar 4.11 Sketsa Tangki Air Bagian Luar

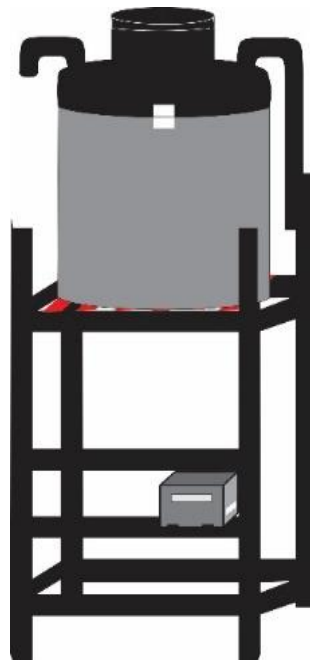
Gambar 4.11 menunjukkan penampakan dari tangki air dari luar dimana tangki ini memiliki ketinggian dari dasar sampai dengan penutup tangki adalah setinggi 44cm, dengan tinggi penutup tangki setinggi 14cm dan bagian inti tangki setinggi 30cm. Bagian inti ini 5cm dari dasar adalah batas rendah air dan 25cm keatas itu merupakan titik air sudah cukup sampai dengan penuh. Jarak dari dasar sampai ke penutup besar adalah 33cm sedangkan jarak penutup kecil adalah 11cm.



Gambar 4.12 Sketsa Tangki Air Bagian Dalam

Pada gambar 4.12 menunjukan tangki bagian dalam, gambar tampak bawah dari tutup tangki kecil dan gambar tutup tangki secara keseluruhan. Bagian dalam tangki seperti yang sebelumnya telah disebutkan 5cm dari dasar merupakan bagian terendah air. Sedangkan dari ujung sensor sampai dengan bagian terendah air adalah jarak baca sensor yaitu setinggi 37cm dan ini merupakan tinggi

maksimal yang bisa dibaca sensor pada penelitian ini. Pada tutup tangki memiliki diameter 12cm itu merupakan tutup tangki paling atas yang tampak dari bawah. Bagian dalam tutup tangki tersebut merupakan tempat menyimpan sensor yang akan mendeteksi permukaan air peletakan sensor berada pada jari-jari tutup tangki tersebut yaitu di jarak 6cm. Tutup tangki kecil merupakan bagian dari tutup yang besar yang disimpan dibagian tengah-tengah tutup tangki besar.



Gambar 4.13 Sketsa Alat Penampung Air Keseluruhan

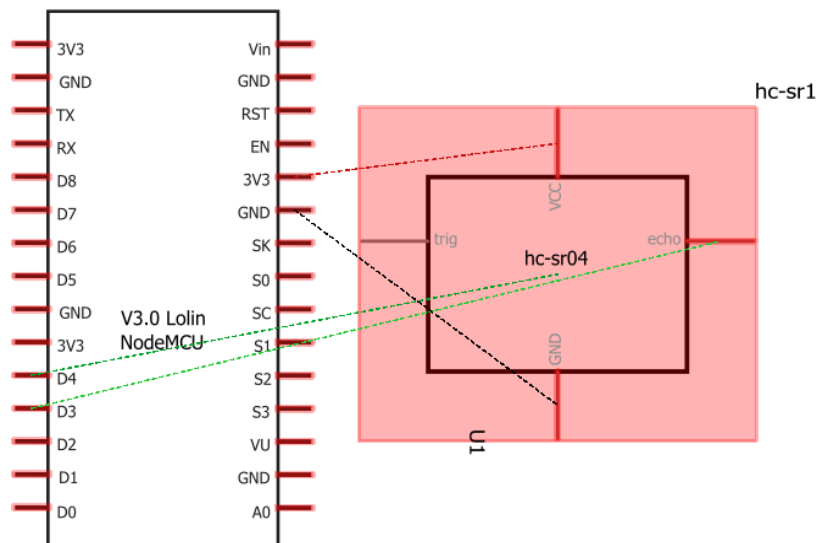
Gambar 4.13 merupakan gambar *prototype* secara keseluruhan yang merupakan gambaran bahwa alat akan terbentuk seperti pada gambar tersebut. Tangki air akan disimpan dibagian atas penyangga sehingga akan lebih terlihat mirip dengan aslinya sedangkan untuk beberapa rangkaian komponen akan dimasukkan kedalam kotak kecil yang di simpan dibawah dengan tujuan agar menyerupai

aslinya. Karena *prototype* merupakan bentuk mini dari benda aslinya perbandingan pada *prototype* ini dengan ukuran sesungguhnya 1:8.

c. Rangkaian Sensor Ultrasonik sebagai sensor jarak

Sensor ultrasonik berfungsi sebagai sensor jarak yang mampu mendeteksi jika ada benda yang berada di hadapannya. Sensor ultrasonik yang digunakan adalah tipe HC-SR04 yang mampu mendeteksi lebih dari 2 meter namun pada penelitian ini disesuaikan dengan kebutuhan yaitu disesuaikan dengan ketinggian pada tangki air yang hanya mencapai 37 cm.

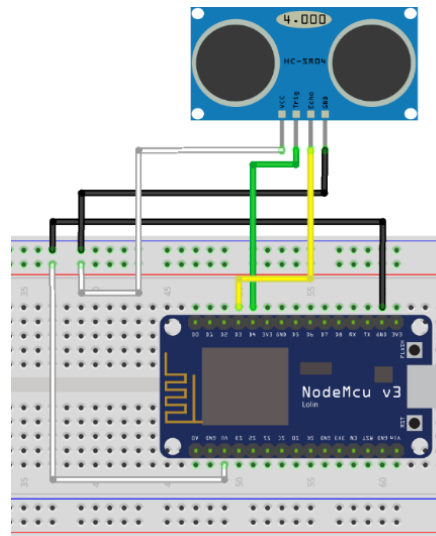
Berikut merupakan gambar dari skema rangkaian sensor ultrasonik yang terhubung dengan NodeMCU ESP8266 v3:



Gambar 4.14 Skema Rangkaian Sensor Ultrasonik

Sensor ultrasonik sendiri memiliki 4 pin, VCC, GND, Trigger dan pin Echo, pin Echo akan terhubung dengan

pin D3 sedangkan Trigger akan terhubung dengan pin D4 sedangkan pin VCC akan terhubung dengan pin VU dan GND akan terhubung dengan GND pada NodeMCU ESP8266 v3. Sensor ini akan terletak pada tutup tangki dan mengarah ke bawah sehingga dapat mendeteksi permukaan air nya. Berikut adalah gambaran dari rangkaian sensor ultrasonik:

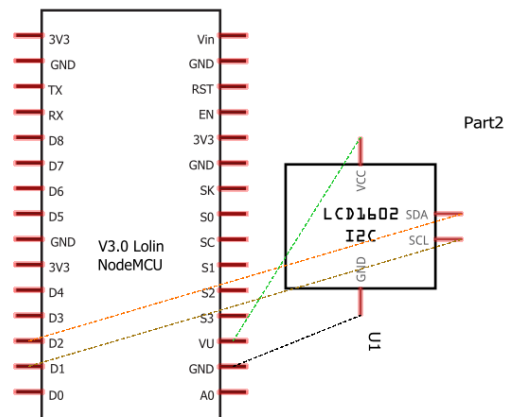


Gambar 4.15 Rangkaian Sensor Ultrasonik dengan NodeMCU

d. Rangkaian LCD (*Liquid Crystal Display*)

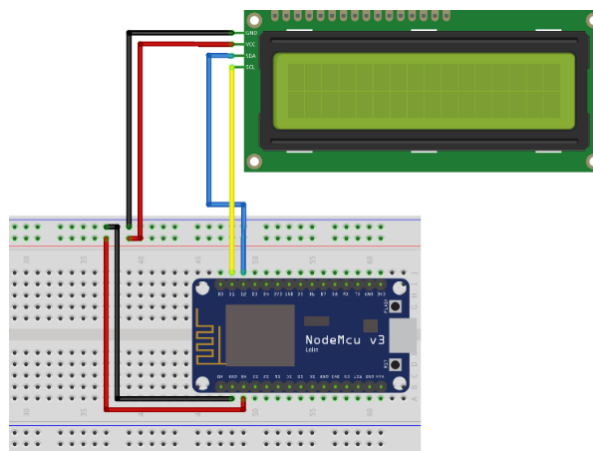
LCD berfungsi sebagai media penampil dari hasil baca sensor ultrasonik, LCD akan menampilkan bagaimana kondisi status tangki dan berapa ketinggian yang dibaca oleh sensor. LCD yang digunakan adalah LCD I2C 16x2 yang mampu menampilkan 16 karakter sebanyak 2 baris.

Berikut adalah skema dari rangkaian LCD dengan NodeMCU ESP8266 v3:



Gambar 4.16 Skema Rangkaian LCD

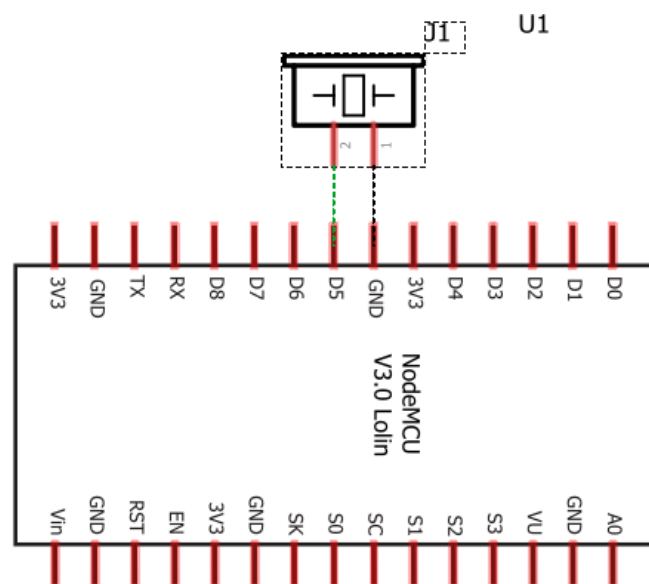
LCD memiliki 4 pin yang masing-masing terhubung dengan NodeMCU ESP8266, pin tersebut yaitu VCC terhubung dengan VU, GND terhubung dengan GND, SLC terhubung dengan D1 dan SDA terhubung dengan D2. LCD sendiri akan disimpan didalam kotak kecil didalam rangkaian alat seluruhnya seperti nampak pada gambar 4.6 gambar A. Dibawah ini merupakan rangkaian LCD dengan NodeMCU ESP8266:



Gambar 4.17 Rangkaian LCD dengan NodeMCU ESP8266

e. Rangkaian Buzzer

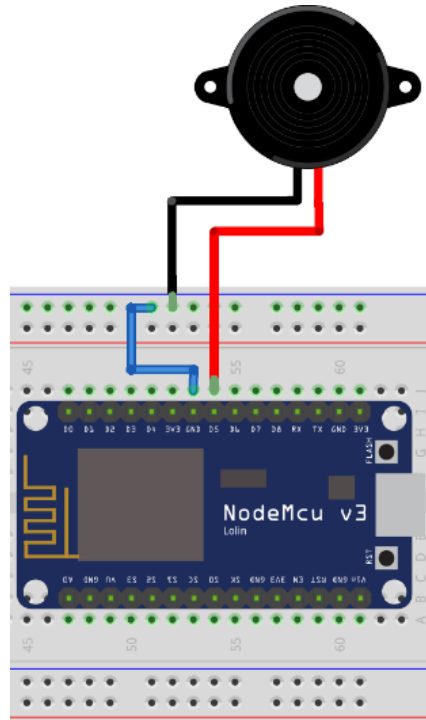
Buzzer berfungsi sebagai media notifikasi kepada pengguna yang menandakan kondisi tangki, jika buzzer berbunyi terus menerus maka tangki sedang kosong dan pompa akan menyala, dan jika buzzer berbunyi hidup-mati menandakan air cukup tetapi pompa akan tetap menyala sedangkan jika buzzer mati maka air penuh dan juga pompa akan mati. Setelah buzzer mati pengguna bisa memastikan melalui LCD apakah pompa mati atau tidak dengan melihat ketinggian air yang ditampilkan.



Gambar 4.18 Skema Rangkaian Buzzer

Buzzer hanya memiliki 2 pin saja yaitu pin positif (+) dengan ditunjukan kakinya lebih panjang dan pin negatif (-) kakinya sedikit lebih pendek. Kedua pin tersebut terhubung dengan NodeMCU melalui pin GND untuk negatif dan pin D5 untuk positif. Buzzer ini akan disimpan bersama LCD disebuah kotak kecil pada

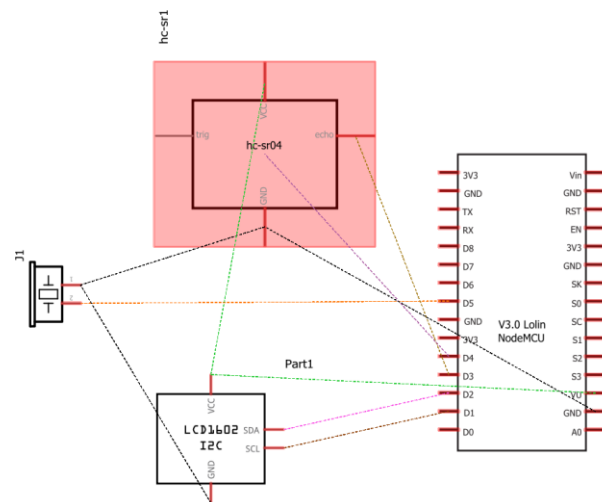
rangkaian alat. Dibawah ini adalah rangkaian elektronika yang menghubungkan buzzer dengan NodeMCU:



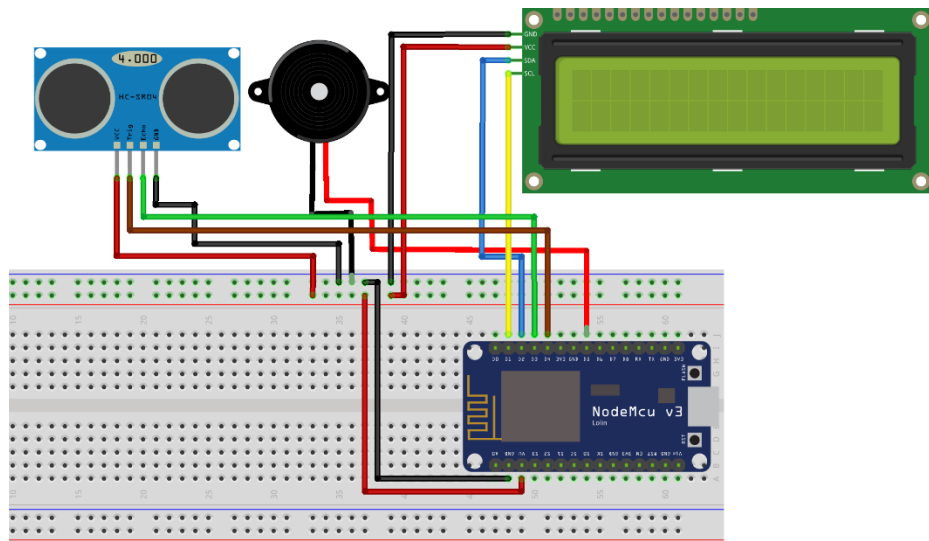
Gambar 4.19 Rangkaian Buzzer

f. Rangkaian Keseluruhan Alat *Monitoring*

Skema rangkaian keseluruhan alat *monitoring* ketinggian air ini berguna untuk mengetahui bagaimana semua modul elektronika yang digunakan terhubung. Setelah mendapat gambaran dari tiap komponen dari gambar-gambar sebelumnya maka penulis melakukan penghubungan dengan tiap komponen dan memastikan semua pin terhubung dengan benar.



Gambar 4.20 Gambar Skema Rangkaian Alat Keseluruhan

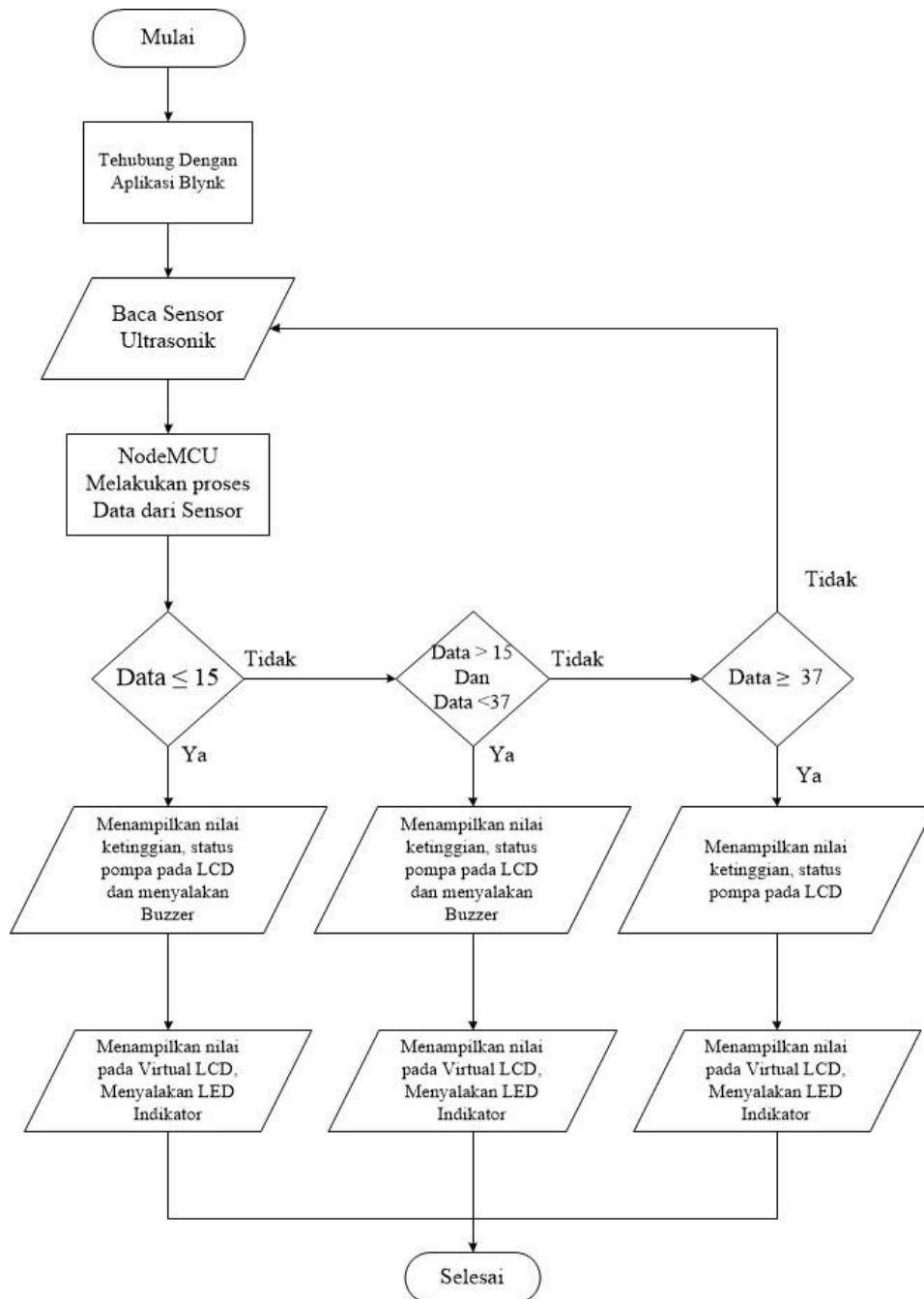


Gambar 4.21 Rangkaian Alat Secara Keseluruhan

Gambar 4.21 menunjukkan gambaran rangkaian modul alat secara keseluruhan dimana pada gambar tersebut tiap modul telah terhubung dengan board NodeMCU ESP8266 v3 sesuai dengan pin yang dijelaskan pada analisis kebutuhan. Pada gambar ini juga menjelaskan rangkaian alat

yang akan dibangun akan terhubung seperti pada gambar namun tetap akan ada penyesuaian dengan kondisi tangki air.

g. Flowchart Monitoring Ketinggian Air



Gambar 4.22 Flowchart Monitoring Ketinggian Air

Flowchart ini berfungsi sebagai indikasi bagaimana sistem akan berjalan. Pada saat dimulai maka sistem yang terhubung dengan internet akan tersambung ke aplikasi blynk sebagai *interface* dari sistem ini. Setelah itu sensor akan membaca kondisi ketinggian pada tangki air dan jika tinggi air kurang dari sama dengan 15 cm maka pompa akan menyala, LCD akan menampilkan ketinggian sesuai dengan hasil baca sensor, buzzer akan menyala menandakan air kosong sedangkan pada aplikasi blynk mampu menampilkan volume air. Jika air antara 15 cm dan 37 cm pompa masih menyala dan LCD menampilkan nilai ketinggian hasil baca sensor, dan buzzer menyala hidup mati serta aplikasi blynk menampilkan volume air sesuai dengan tinggi yang dibaca. Selanjutnya jika ketinggian air lebih dari 37 cm maka pompa dan buzzer akan mati, LCD tetap akan menampilkan ketinggian air dan aplikasi blynk menampilkan volume air terbaru.

4.2.3 Perancangan Perangkat Lunak (*Software*)

Perancangan perangkat lunak digunakan sebagai pendukung kerja sistem *monitoring*. Perintah/program pada perangkat lunak akan menjalankan sistem secara keseluruhan sesuai kondisi yang dikehendaki oleh peneliti yang terpusat pada board NodeMCU ESP8266 v3. *Software* Arduino IDE dalam penelitian ini digunakan untuk membuat sketsa logika yang akan di-*upload* ke dalam board NodeMCU ESP8266.

BAB V

IMPLEMENTASI DAN PENGUJIAN

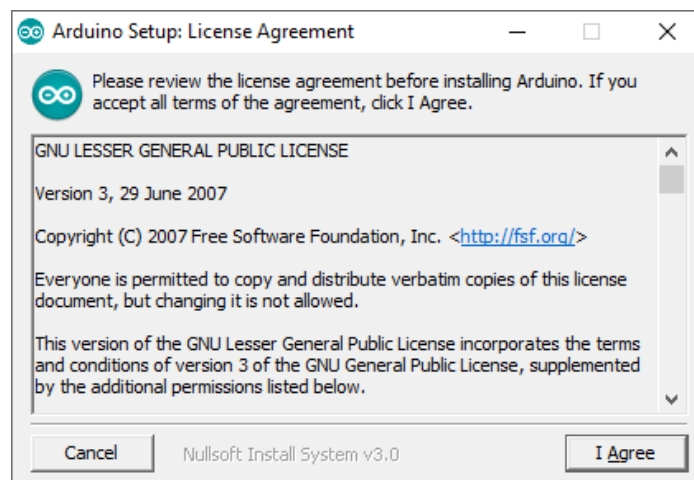
5.1 Implementasi

Implementasi pada penelitian *monitoring* ketinggian air ini dilakukan di rumah saya sendiri yang bertempat di desa Cibeet kec. Ibun.

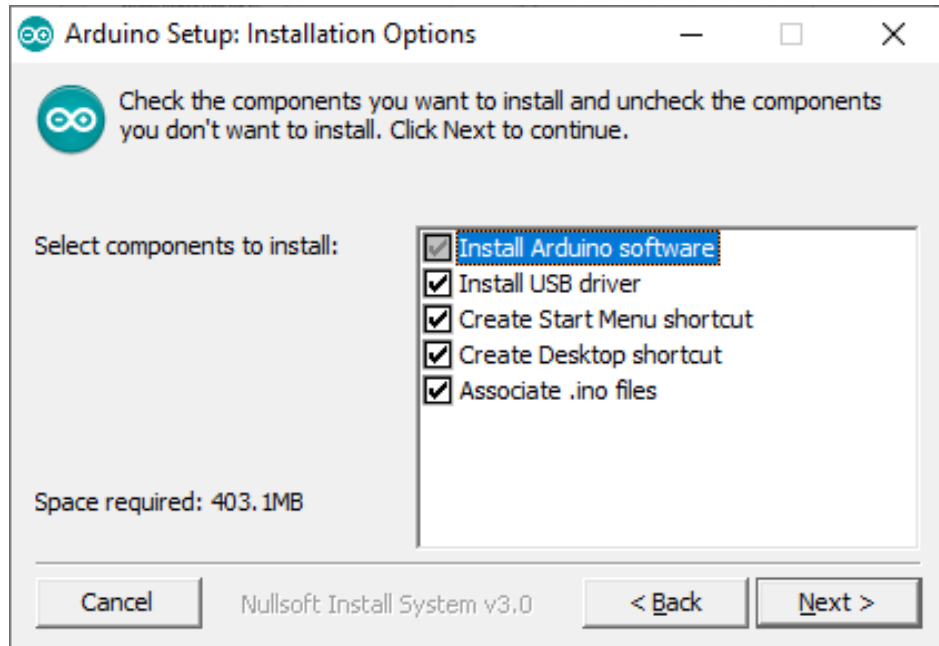
Implementasi ini dilakukan dalam bentuk eksperimen pada penelitian *monitoring* ketinggian air apakah alat yang dibangun bekerja sesuai dengan kontruksi dan desain yang telah dibuat sebelumnya. Implementasi ini berisikan beberapa gambar dari rangkaian modul-modul elektronika yang sesungguhnya. Implementasi ini dilakukan sesuai dengan perancangan yang ada pada bab sebelumnya.

5.1.1 Implementasi Arduino IDE

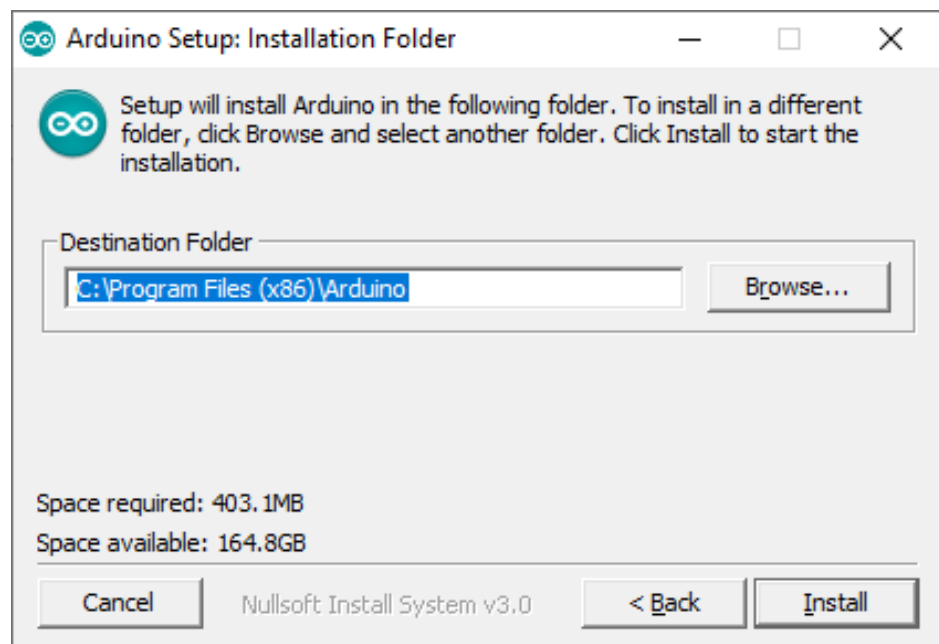
Sebelum melakukan implementasi terhadap alat atau modul elektronika yang akan digunakan pertama-tama melakukan implementasi terhadap Arduino IDE yang akan menjadi *Code Editor* pada penelitian ini. Berikut adalah gambar dari Arduino IDE saat akan melakukan instalasi dan *setting* agar bisa mendeteksi board NodeMCU ESP8266.



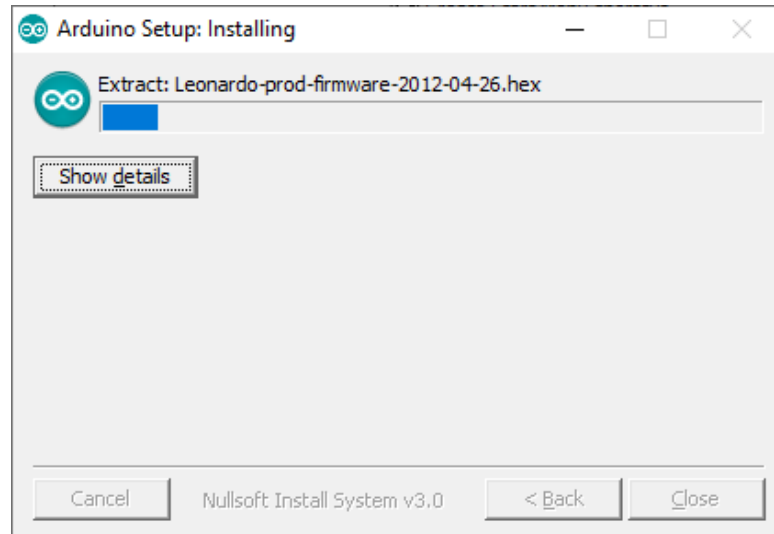
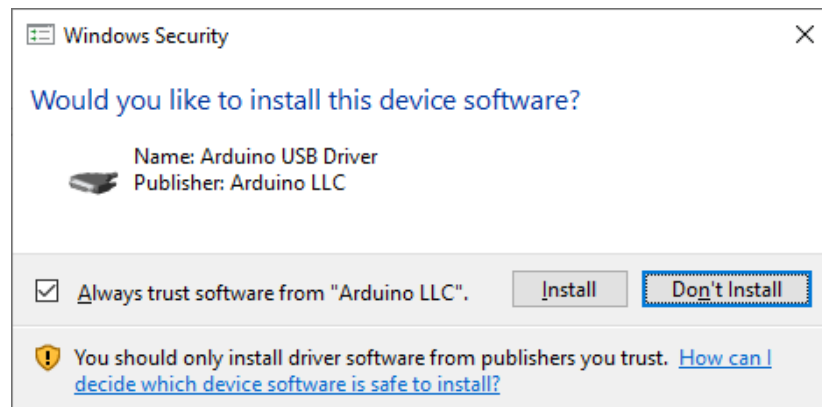
Gambar 5.1 *Arduino Setup License Agreement*



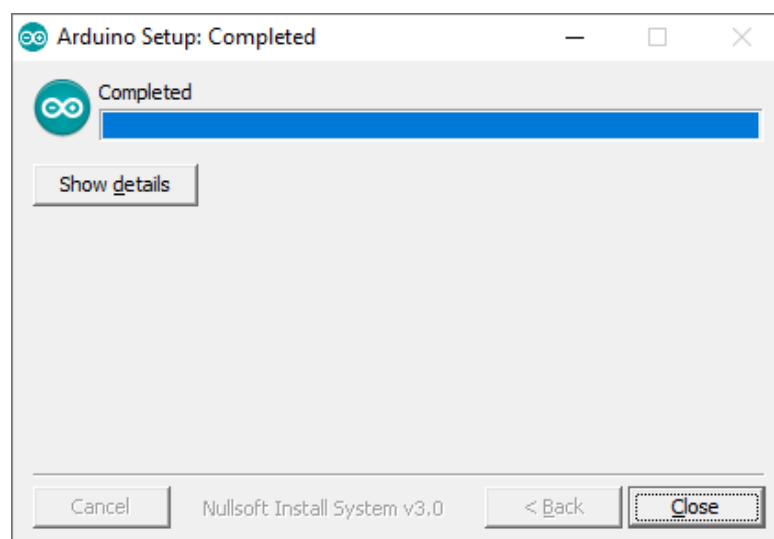
Gambar 5.2 Arduiono *Setup Installation Options*



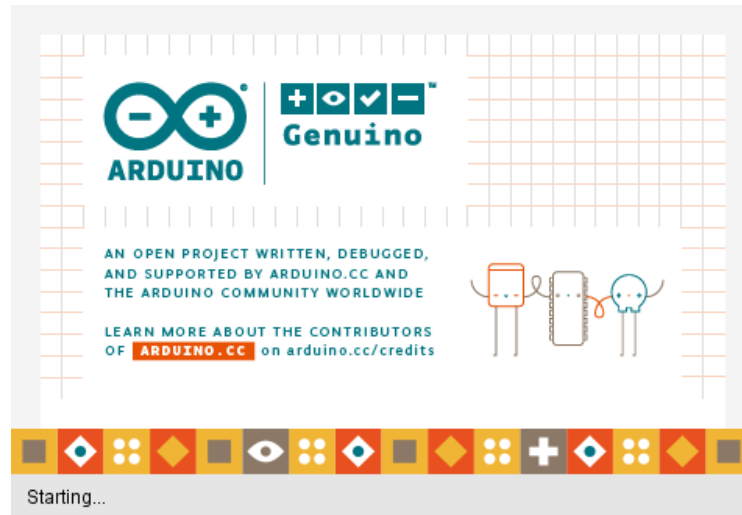
Gambar 5.3 Arduino *Setup Installation Folder*

Gambar 5.4 Proses *Installing* Arduino IDE

Gambar 5.5 Tampilan Jika Belum Instal USB Driver

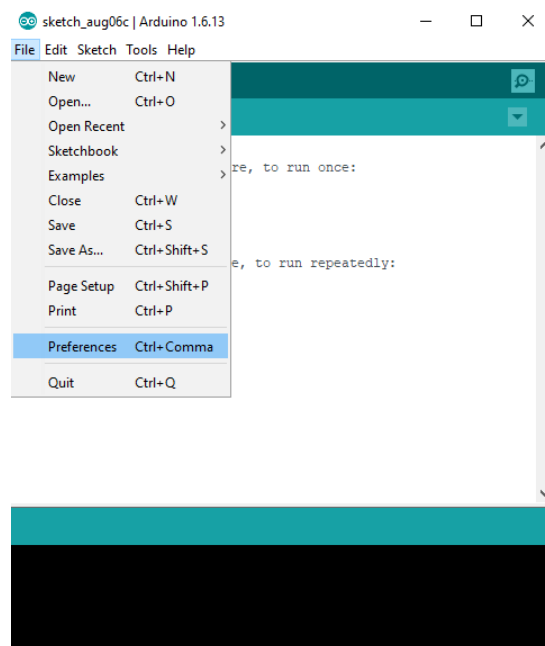


Gambar 5.6 Proses Instal Arduino Selesai

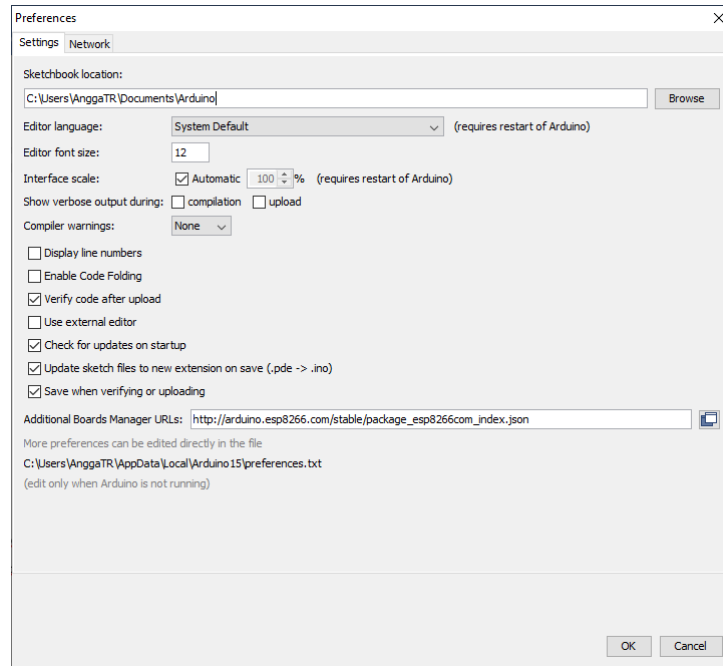


Gambar 5.7 Tampilan Mulai Dari Arduino IDE

Gambar diatas adalah beberapa gambar dari proses instalasi Arduino IDE yang akan menjadi *Code Editor* pada penelitian ini. Proses instalasi Arduino IDE sama saja dengan proses instal aplikasi lainnya. Untuk selanjutnya agar board NodeMCU bisa terdeteksi dan bisa diprogram melalui Arduino IDE maka pertama harus memasukan library NodeMCU kedalam Arduino IDE melalui File – *preferences* (ctrl + comma). Berikut adalah tahapan prosesnya:

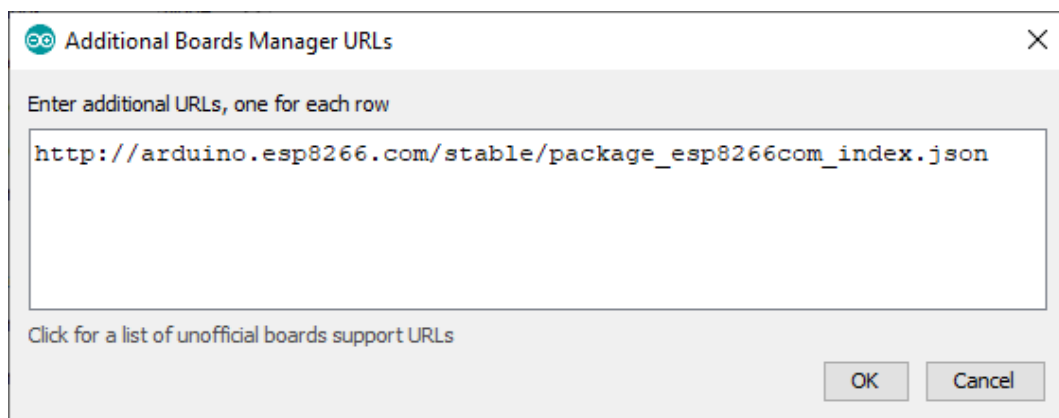


Gambar 5.8 Tampilan Arduino IDE Tab File *Preferences*



Gambar 5.9 Tampilan Preferences Arduino IDE

Gambar 5.9 merupakan tampilan jika telah memilih *preferences* selanjutnya memasukan link yang berisi library NodeMCU ESP8266 dengan cara memilih *Additional Boards Manager URLs*, untuk selanjutnya agar mendownload boardnya. link tersebut tidak hanya berisi board NodeMCU saja tetapi masih ada beberapa board lainnya.



Gambar 5.10 Memasukan Link Library NodeMCU

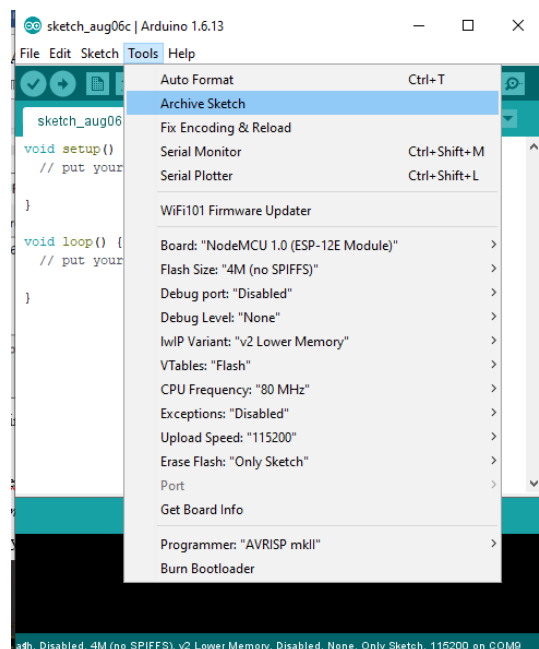
Gambar 5.10 merupakan tampilan dari link yang berisi library NodeMCU ESP8266 yang selanjutnya akan melakukan proses *download*, dan pastikan bahwa

perangkat laptop atau komputer telah terhubung dengan internet. Setelah selesai men-*download* selanjutnya memasangkannya pada Arduino IDE dengan cara sebagai berikut:

1. Memilih tab Tools

Tab ini berisi beberapa pengaturan yang berfungsi agar sebuah board dapat berjalan dengan baik. Pengaturan yang akan dipilih adalah Board dimana disini akan memilih board NodeMCU.

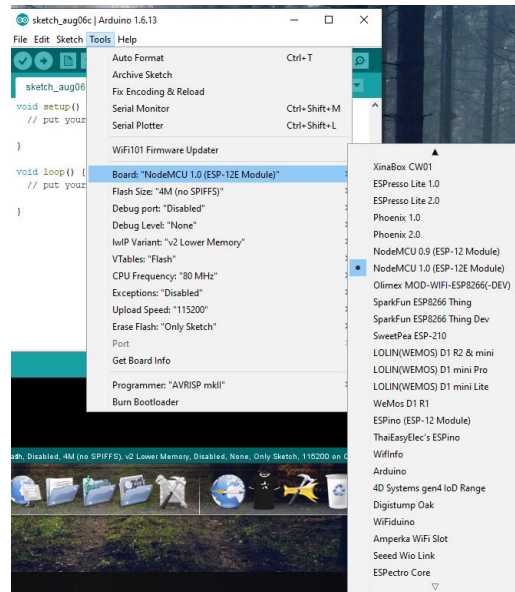
Tab tools juga berisi kecepatan dalam melakukan *upload* kedalam board, port berapa yang digunakan oleh board tersebut. Untuk pengaturan lainnya biarkan default saja.



Gambar 5.11 Tampilan Tab Tools Pada Arduino IDE

2. Melakukan Perubahan Pada Board

Pada tahapan ini melakukan perubahan pada board dan disesuaikan dengan board yang akan digunakan, yaitu NodeMCU 1.0 (ESP-12E Module) ini merupakan library NodeMCU v3 dan sesuai dengan board yang digunakan. Setelah selesai memilih board maka Arduino IDE sudah bisa digunakan pada NodeMCU ESP8266 v3.

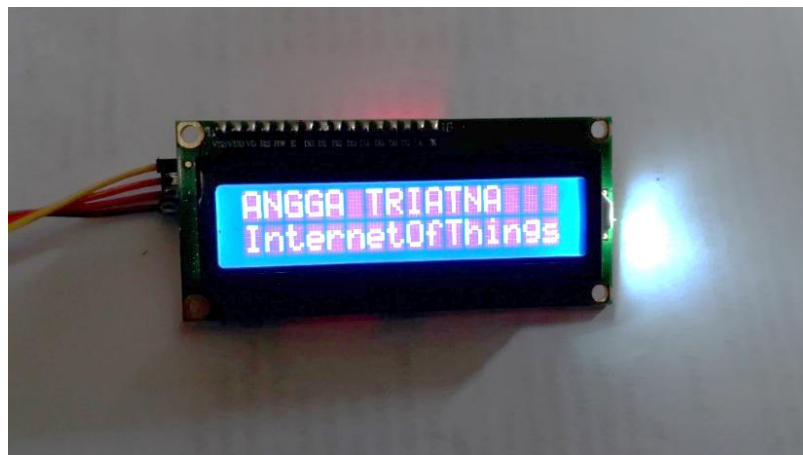


Gambar 5.12 Tampilan Tab Tools Memilih Board

5.1.2 Implementasi Modul Elektronika

Berikut adalah beberapa gambar implementasi rangkaian elektronika yang digunakan pada penelitian *monitoring* ketinggian air:

1. Tampilan LCD yang terhubung dengan board NodeMCU ESP8266

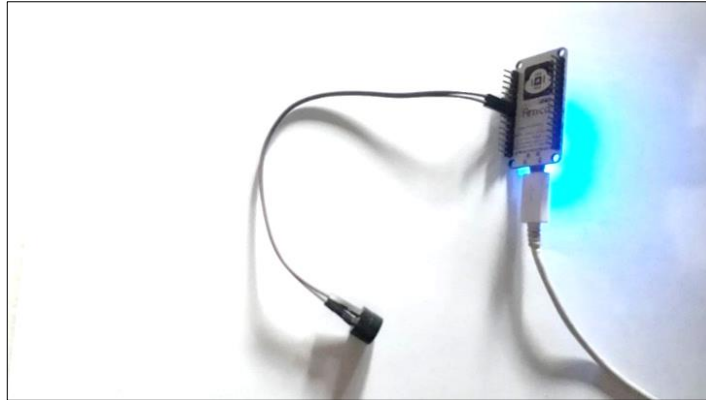


Gambar 5.13 Tampilan LCD yang terhubung dengan NodeMCU

Gambar diatas menunjukkan bagaimana tampilan dari LCD 16x2 yang digunakan pada penelitian ini. LCD diatas pada baris

pertama mencetak “ANGGA TRIATNA” dan pada baris kedua mencetak “InternetOfThings”. Percetakan ini dilakukan untuk memastikan bahwa LCD dapat berfungsi dengan normal.

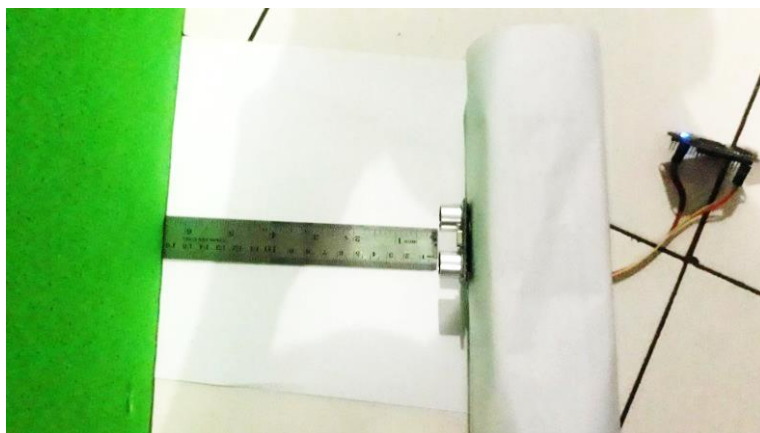
2. Tampilan buzzer yang terhubung dengan board NodeMCU ESP8266



Gambar 5.14 Tampilan buzzer yang terhubung dengan NodeMCU

Pada tampilan gambar diatas menunjukkan implementasi dari buzzer terhadap NodeMCU ESP8266 yang menjadi salah satu modul penunjang dalam penelitian *monitoring* ketinggian air. Buzzer akan memunculkan bunyi jika kondisi tangki kosong dan pompa menyala. Pada gambar diatas memastikan bahwa buzzer menyala dengan baik.

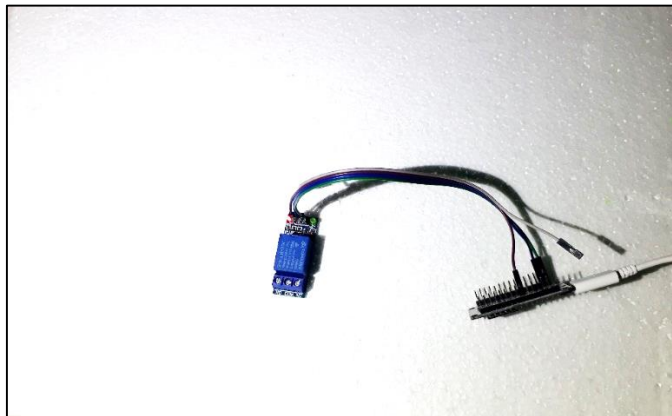
3. Tampilan Sensor yang terhubung dengan board NodeMCU ESP8266



Gambar 5.15 Tampilan Sensor Ultrasonik dengan Board NodeMCU

Pada gambar ini menunjukkan implementasi dari sensor ultrasonik yang berfungsi sebagai pendeteksi jarak permukaan air pada penelitian *monitoring* ketinggian air berbasis *internet of things*. Sensor ultrasonik ini menjadi modul elektronika yang mengatur pompa air dan buzzer. Dimana pompa dan buzzer akan bekerja setelah sensor mendeteksi permukaan air atau berhasil mendeteksi benda yang berada di depannya.

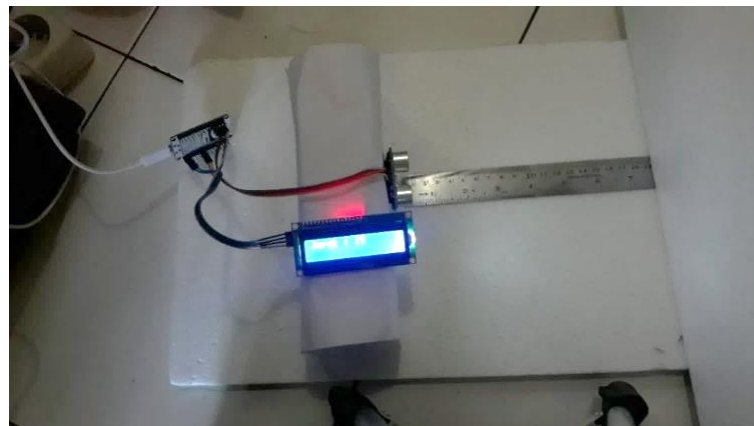
4. Tampilan NodeMCU dengan *Relay Single Channel*.



Gambar 5.16 NodeMCU dengan *Relay Single Channel*

Pada gambar diatas menunjukkan bagaimana *relay* terhubung dengan board NodeMCU ESP8266. Dimana *relay* disini akan menjadi sebuah saklar yang terhubung juga dengan pompa air dan akan mengatur pompa air tersebut hidup atau mati yang telah disesuaikan dengan kode program.

5. Tampilan LCD dan Sensor Ultrasonik dengan NodeMCU



Gambar 5.17 LCD dan Sensor Ultrasonik dengan NodeMCU

Gambar diatas menampilkan implementasi dari LCD dan Sensor Ultrasonik yang terhubung dengan board NodeMCU. Dimana pada gambar tersebut sensor akan membaca seberapa jauh jarak benda yang berada di depannya dan akan menampilkannya pada LCD yang telah terhubung. LCD berguna sebagai *output* penampil dari hasil baca sensor.

6. Tampilan LCD dan Buzzer dengan NodeMCU ESP8266.



Gambar 5.18 Tampilan LCD dan Buzzer dengan NodeMCU buzzer menyala



Gambar 5.19 Tampilan LCD dan Buzzer dengan NodeMCU buzzer mati

Pada gambar 5.18 dan gambar 5.19 ini menunjukkan implementasi dari kondisi dimana buzzer menyala dan buzzer mati. Kondisi tersebut ditampilkan pada LCD selaku media *output*. Pada kondisi menyala maka buzzer akan berbunyi selama satu detik dan akan berubah menjadi kondisi mati selama satu detik selanjutnya.

7. Tampilan Interface Virtual LCD pada aplikasi Blynk



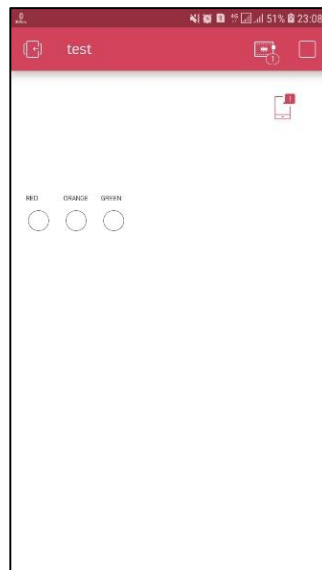
Gambar 5.20 Tampilan *Interface* Virtual LCD Pada Aplikasi Blynk

Gambar diatas menunjukkan implementasi dari LCD yang ada pada aplikasi blynk. LCD tersebut memiliki fungsi sama seperti LCD konvensional yaitu untuk menampilkan hasil baca sensor yang dilakukan oleh sensor ultrasonik. LCD ini juga sama memiliki ukuran 16x2 karena LCD ini sudah cukup untuk memenuhi kebutuhan dalam menampilkan *output*.

8. Tampilan interface virtual LED pada aplikasi blynk

Dibawah ini merupakan gambar implementasi dari LED yang terdapat pada aplikasi blynk. LED tersebut berfungsi sebagai lampu indikator yang akan menunjukkan keadaan atau kondisi tangki. Dimana jika kondisi tangki penuh maka LED warna hijau akan menyala, jika kondisi tangki airnya cukup LED oranye akan

menyala sedangkan jika kondisi tangki kosong maka LED merah akan menyala.



Gambar 5.21 Tampilan *Interface* Virtual LED pada aplikasi blynk

9. Tampilan keseluruhan modul elektronika yang digunakan



Gambar 5.22 Tampilan Keseluruhan Modul Elektronika

Tampilan pada gambar 5.22 menampilkan bagaimana tiap modul elektronika yang akan digunakan terhubung dengan NodeMCU secara keseluruhan. Pada gambar diatas sensor akan membaca jarak benda didepannya dan akan menampilkan pada LCD, jika mencapai jarak tertentu maka buzzer akan menyala. Cara kerja inilah yang digunakan untuk mengukur permukaan ketinggian

air yang akan menjadi dasar pada penelitian *monitoring* ketinggian air.

10. Tampilan Alat Secara Keseluruhan dan Siap Pakai



Gambar 5.23 Tampilan Alat Secara Keseluruhan

Gambar ini menunjukkan tampilan alat secara keseluruhan dan tampilan alat siap pakai untuk melakukan *monitoring* ketinggian air. Disini setiap komponen elektronika sudah terhubung dengan baik dan dipastikan sudah berfungsi dan sudah dikemas cukup rapih. Tangki air yang digunakan memiliki tinggi 44 cm dari atas sampai bawah, bagian dalam tangki memiliki tinggi maksimal 37 cm ini merupakan titik aman pada tangki untuk menghindari air meluap dan mengenai sensor. Pada gambar ini juga sudah terhubung dengan perangkat *smartphone* melalui aplikasi blynk agar mampu melakukan *monitoring* secara jarak jauh. Pada tampilan *smartphone* dapat mengetahui berapa volume air di dalam tangki serta mengetahui apakah kondisi air pada tangki sudah penuh, kosong atau air cukup ini ditunjukkan dengan menyalakan lampu LED pada aplikasi blynk yang digunakan.

5.2 Pengujian

Pengujian dilakukan untuk memastikan tiap alat berjalan sesuai dengan fungsinya dan sesuai dengan kode yang telah dibuat sebelumnya. Dibawah ini adalah pengujian dari beberapa modul:

5.2.1 Pengujian Modul Elektronika

Pengujian modul elektronika bertujuan untuk mengetahui kinerja setiap modul yang digunakan pada penelitian *monitoring* ketinggian air. Berikut adalah beberapa pengujian yang dilakukan:

1. Pengujian IP Address yang terdapat pada NodeMCU ESP8266.

Pengujian IP Address bertujuan untuk mengetahui IP Address pada board NodeMCU ESP8266 jika terhubung dengan internet. Berikut adalah kode program dan tampilan hasil pengujian:



```

cekwifi | Arduino 1.6.13
File Edit Sketch Tools Help

cekwifi $
#include <ESP8266WiFi.h>

const char* ssid = "Angga";
const char* pass = "privatel2";
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  delay(10);
  Serial.println();

  Serial.print("Connecting...");
  Serial.println(".....");

  WiFi.begin(ssid, pass);

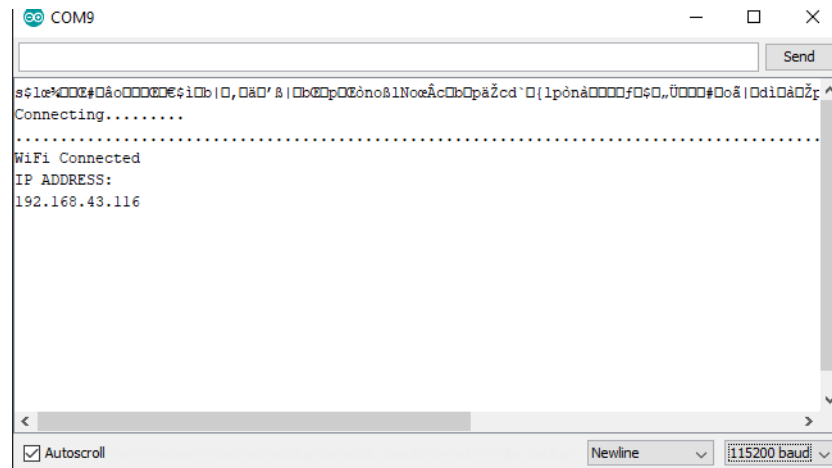
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP ADDRESS: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // put your main code here, to run repeatedly:
}

Done uploading.
..... [ 30% ]
..... [ 60% ]
..... [ 91% ]
..... [ 100% ]
5

```

Gambar 5.24 Kode Untuk Pengujian IP Address pada Board NodeMCU



Gambar 5.25 Hasil Uji Coba untuk mengetahui IP Address

Hasil uji coba untuk mengetahui IP Address seperti ditunjukkan gambar diatas, dimana IP di dapat setelah board NodeMCU terhubung dengan koneksi internet. IP tidak akan muncul jika board belum terhubung dengan internet. IP dibutuhkan sebagai pengalamatan yang akan berguna untuk mengirimkan data ke web server yang digunakan yaitu Server Blynk.

Tabel 5.1 Pengujian IP Address pada board NodeMCU

No	Nama	Hasil	Keterangan
1	Pengujian IP terhubung dengan internet	Muncul IP Address	Berhasil
2	Pengujian IP	Tidak muncul	Tidak berhasil

2. Pengujian sensor ultrasonik terhadap benda yang berada didepannya.

Pengujian Sensor ultrasonik berguna untuk memastikan bahwa sensor berjalan dengan baik dan sesuai dengan kode program yang telah dibuat. Berikut adalah gambar hasil pengujian dari sensor ultrasonik dan kode program yang digunakan:

```

ultras | Arduino 1.6.13
File Edit Sketch Tools Help

ultras
}
void sensorUltrasonic(int pinTriger, int pinEcho)
{
  Serial.println(distance);
  digitalWrite(pinTriger, LOW);
  delayMicroseconds(2);
  digitalWrite(pinTriger, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTriger, LOW);
  duration = pulseIn(pinEcho, HIGH);
  distance = (duration / 2) * 0.034;

  Serial.print("Jarak ");
  Serial.print(distance);
  Serial.print("cm");

  delay(1000);
}

Done compiling.

Sketch uses 264,172 bytes (25%) of program storage space. Maximum
Global variables use 26,824 bytes (32%) of dynamic memory, leaving
as9 Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM9

```

Gambar 5.26 Kode Program pengujian Sensor Ultrasonik

```

COM9
Send

$€/κ-Spÿ
UJI COBA Ultrasonik
Jarak 16cm
Jarak 17cm
Jarak 16cm

Autoscroll Newline 9600 baud

```

Gambar 5.27 Tampilan Hasil Baca Sensor Ultrasonik

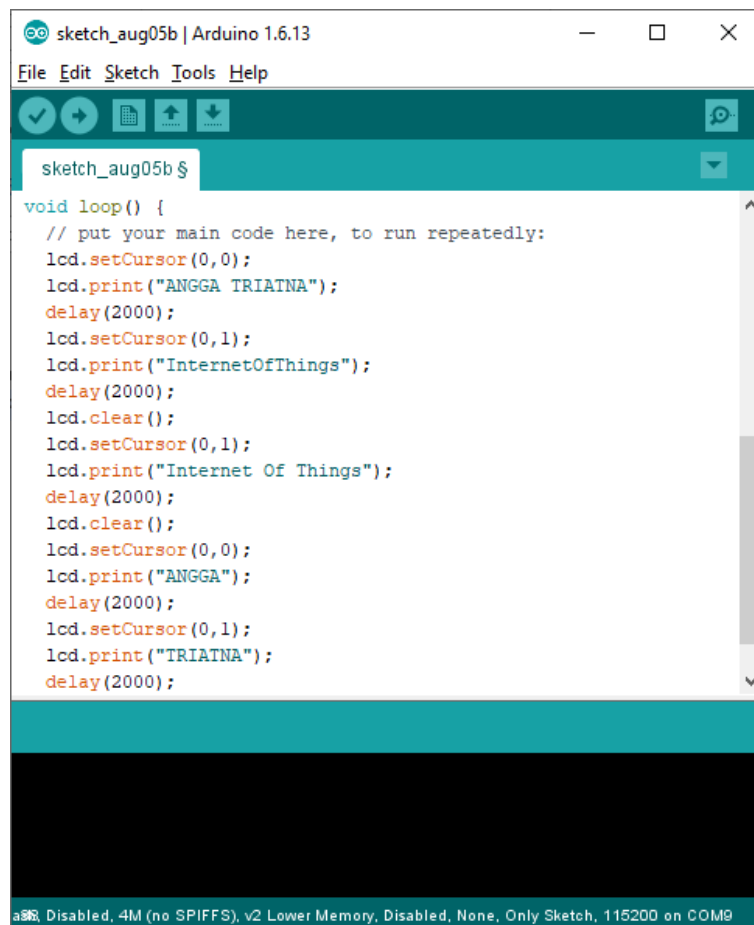
Uji coba pada sensor ultrasonik dilakukan 4 kali dengan jarak yang tidak jauh berbeda. Pengujian dilakukan dengan perpindahan yang cepat dan menguji seberapa cepat sensor membaca perubahan itu. Hasil uji coba pada sensor ultrasonik untuk mengukur jarak benda yang ada didepannya ditunjukkan oleh tabel sebagai berikut:

Tabel 5.2 Pengujian Sensor Ultrasonik

No	Hasil Baca Sensor	Jarak Sesungguhnya	Keterangan
1	16 cm	16 cm	Berhasil
2	17 cm	17 cm	Berhasil
3	16 cm	15 cm	Selisih 1
4	15 cm	15 cm	Berhasil

3. Pengujian LCD dalam menampilkan teks.

Pengujian LCD berguna untuk memastikan bahwa LCD benar-benar berfungsi dan mampu menampilkan hasil dari kode program yang telah dibuat sebelumnya. Dan melakukan uji coba apa bila karakter yang dituliskan lebih dari 16 apa yang akan terjadi. Berikut adalah kode program yang digunakan sebagai uji coba dan tampilan hasil dari serial monitor.



```

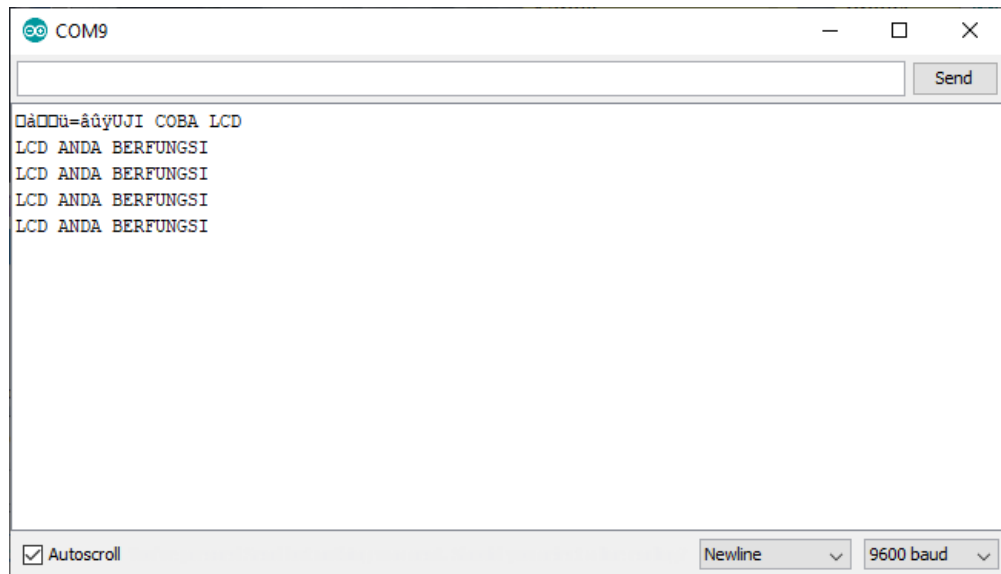
sketch_aug05b | Arduino 1.6.13
File Edit Sketch Tools Help

sketch_aug05b $
void loop() {
  // put your main code here, to run repeatedly:
  lcd.setCursor(0,0);
  lcd.print("ANGGA TRIATNA");
  delay(2000);
  lcd.setCursor(0,1);
  lcd.print("InternetOfThings");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,1);
  lcd.print("Internet Of Things");
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("ANGGA");
  delay(2000);
  lcd.setCursor(0,1);
  lcd.print("TRIATNA");
  delay(2000);
}

```

Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM9

Gambar 5.28 Kode Program Pengujian LCD



Gambar 5.29 Tampilan Hasil Pengujian LCD

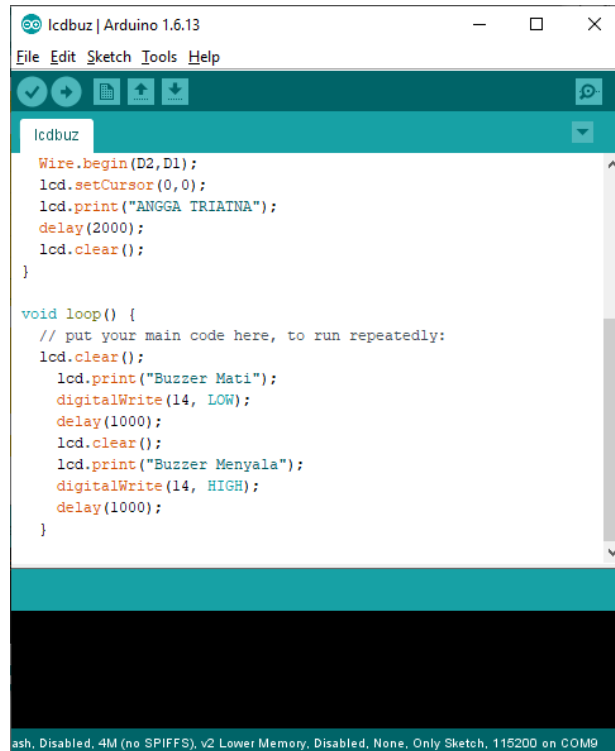
Uji coba pada LCD dilakukan 1 kali dengan 1 kode program yang setiap 2 detik akan berubah tampilan dan akan menampilkan karakter seperti pada tabel dibawah. Hasil uji coba pada LCD untuk menampilkan karakter ditunjukkan pada tabel berikut ini:

Tabel 5.3 Pengujian LCD

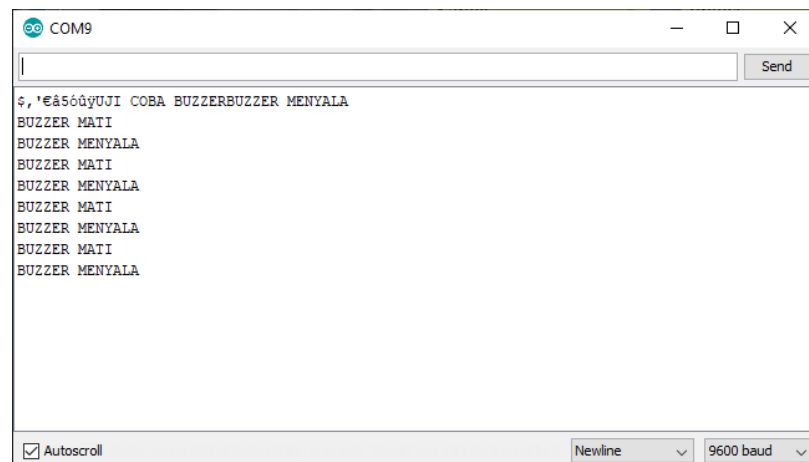
No	Menampilkan	LCD Baris 1	LCD Baris 2	Ket
1	ANGGA TRIATNA	ANGGA TRIATNA	-	Berhasil
2	ANGGA TRIATNA InternetOfThings	ANGGA TRIATNA	InternetOfThings	Berhasil
3	Internet Of Things	Internet Of Thin	-	Melebihi batas
4	ANGGA TRIATNA	ANGGA	TRIATNA	Berhasil

4. Pengujian LCD dengan buzzer

Dari rangkaian LCD yang terdapat pada gambar 5.18 dan gambar 5.19 dilanjutkan dengan pengujian pada LCD dan Buzzer. Pengujian LCD dengan buzzer bertujuan untuk mengetahui bahwa LCD dan buzzer bekerja dengan baik dan sesuai dengan kode yang dibuat. Berikut ini adalah kode program dan tampilan pada serial monitor Arduino IDE.



Gambar 5.30 Kode Program LCD dengan Buzzer



Gambar 5.31 Tampilan hasil LCD dengan Buzzer

Uji coba LCD dan buzzer dilakukan sekali dimana setiap detik buzzer akan bergantian hidup dan mati, serta LCD akan mencetak kondisi buzzer tersebut. Jika buzzer kondisi menyala maka LCD akan menampilkan “Buzzer Menyala” sedangkan jika mati maka LCD akan menampilkan “Buzzer Mati” hal ini selaras dengan yang ditampilkan pada serial monitor pada gambar 5.31.

5. Pengujian LCD yang terhubung dengan Sensor dan menampilkan jarak pada LCD.

Pada pengujian kali ini menampilkan hasil baca sensor pada LCD, dimana LCD menjadi *output* untuk menampilkan setiap kali sensor mendeteksi berapa jarak benda yang ada didepannya. Pengujian ini dilakukan beberapa kali untuk memastikan bahwa sensor bisa mendeteksi terhadap perubahan posisi pada benda yang ada didepannya. Pengujian ini sama seperti pengujian sensor ultrasonik sebelumnya yang berbeda berada pada *outputnya* dimana pengujian kali ini *output* akan tampil pada LCD. Untuk tampilan LCD dan sensor terdapat pada gambar 5.17, *output* akan tampil seperti gambar tersebut. Dibawah ini adalah kode program untuk mendeteksi jarak dan ditampilkan pada LCD:



```

lcdsensor | Arduino 1.6.13
File Edit Sketch Tools Help

lcdsensor$

void sensorUltrasonic(int pinTriger, int pinEcho)
{
    Serial.println(distance);
    digitalWrite(pinTriger, LOW);
    delayMicroseconds(2);
    digitalWrite(pinTriger, HIGH);
    delayMicroseconds(10);
    digitalWrite(pinTriger, LOW);
    duration = pulseIn(pinEcho, HIGH);
    distance = (duration / 2) * 0.034;

    lcd.setCursor(0,1);
    lcd.print("Jarak:");
    lcd.print(distance);
    lcd.print("cm");

    delay(1000);
}

Done Saving.

WARNING: library NewPing claims to run on [avr architecture(s) and

Sketch uses 269,960 bytes (25%) of program storage space. Maximum :
Global variables use 27,496 bytes (33%) of dynamic memory, leaving

Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM9

```

Gambar 5.32 Kode Program Pengujian LCD dengan Sensor

Tabel 5.4 Pengujian LCD dan Sensor Ultrasonik

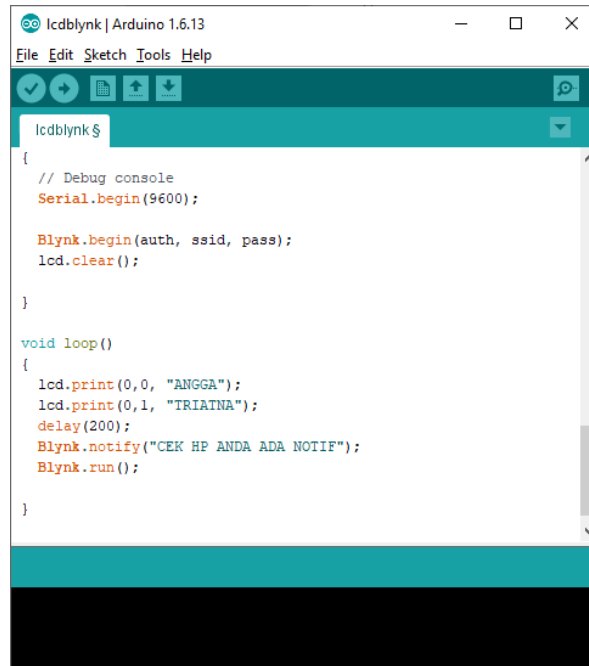
No	Jarak Sesungguhnya	Hasil Baca Sensor	Tampilan LCD	Ket
1	16 cm	16 cm	Jarak : 16 cm	Sesuai
2	17 cm	16 cm	Jarak : 16 cm	Selisih 1
3	18 cm	18 cm	Jarak : 18 cm	Sesuai
4	15 cm	16 cm	Jarak : 16 cm	Selisih 1

Uji coba LCD dan Sensor dilakukan sebanyak 4 kali dimana 2 pengujian sesuai dengan jarak sesungguhnya sedangkan 2 lainnya terdapat selisih 1 cm ini dikarenakan perpindahan jarak benda yang cepat dan tidak stabil sehingga mempengaruhi hasil baca sensor ultrasonik, tetapi karena selisih yang dihasilkan tidaklah besar maka dapat dikatakan kalau sensor ultrasonik berhasil membaca jarak benda yang ada didepannya dengan baik dan sesuai dengan kode program yang telah dibuat.

6. Pengujian Menampilkan teks pada virtual LCD yang terdapat pada aplikasi blynk

Pengujian virtual LCD pada aplikasi blynk memiliki tujuan yang sama dengan pengujian LCD konvensional dimana pengujian ini dilakukan untuk memastikan virtual LCD berjalan dengan baik dan mampu menampilkan apa yang telah dituliskan sesuai dengan sketch atau kode program. Selain itu pengujian ini juga dapat berjalan dengan board NodeMCU yang sudah terkoneksi dengan internet. Berikut adalah tampilan kode program dan hasil pengujian pada *interface* aplikasi blynk ditampilkan pada gambar 5.33 dan gambar 5.34:

Hasil pengujian menunjukkan bahwa virtual LCD dapat berfungsi dengan baik dan sesuai dengan kode program yang telah dibuat sebelumnya. Selanjutnya akan mencoba menambahkan LED pada aplikasi blynk sebagai salah satu uji coba karena LED akan menjadi lampu indikator pada penelitian ini.



Gambar 5.33 Kode Program untuk LCD virtual Blynk

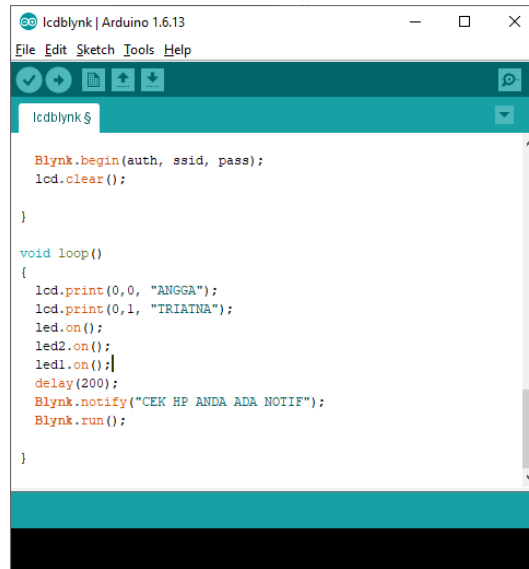


Gambar 5.34 Tampilan Virtual LCD pada Aplikasi Blynk

7. Pengujian lampu indikator yaitu virtual LED yang terdapat pada aplikasi blynk.

Pengujian virtual LED yang terdapat pada aplikasi blynk bertujuan untuk memastikan LED ini berjalan dengan baik dan sesuai dengan kode program yang telah dibuat. Selain itu sama halnya dengan pengujian LCD

pengujian ini juga akan berjalan setelah board NodeMCU terkoneksi dengan internet. Berikut adalah tampilan kode program dan hasil pengujian pada *interface* LED aplikasi blynk:



```

lcdblynk | Arduino 1.6.13
File Edit Sketch Tools Help

lcdblynk $

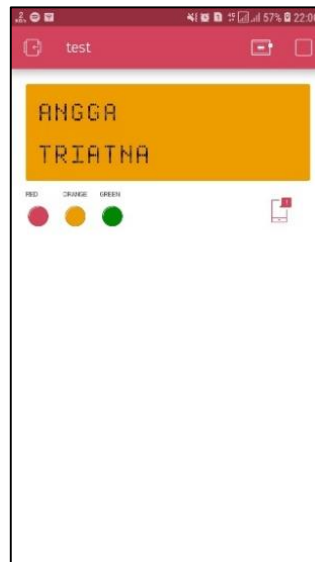
Blynk.begin(auth, ssid, pass);
lcd.clear();

}

void loop()
{
  lcd.print(0,0, "ANGGA");
  lcd.print(0,1, "TRIATNA");
  led.on();
  led2.on();
  led1.on();
  delay(200);
  Blynk.notify("CEK HP ANDA ADA NOTIF");
  Blynk.run();
}

```

Gambar 5.35 Tampilan Kode Program Pengujian LED



Gambar 5.36 Tampilan Virtual LED pada Aplikasi Blynk

8. Pengujian menampilkan jarak pada virtual LCD dan LED.

Setelah berhasil dengan pengujian terhadap LCD dan LED virtual yang terdapat pada aplikasi blynk selanjutnya dilakukan pengujian

pengukuran jarak ketinggian yang akan ditampilkan pada LCD dan menyalakan LED virtual. Berikut ini adalah kode program dan tampilan hasil dari pengujian:



```

lcdlednode | Arduino 1.6.13
File Edit Sketch Tools Help

lcdlednode
digitalWrite(pinTriger, HIGH);
delayMicroseconds(10);
digitalWrite(pinTriger, LOW);
duration = pulseIn(pinEcho, HIGH);
distance = (duration / 2) * 0.034;

delay(100);

lcd1.print(0,1, "Jarak : ");
lcd1.print(8,1, distance);
lcd1.print(14,1, "cm");

if (distance <= 20) {
  lcd1.print(0,0, "Mengukur Jarak");
  led1.on();
  led1.off();
  led2.off();
}

Done compiling.

Sketch uses 281,948 bytes (26%) of program storage space. Maximum
Global variables use 29,588 bytes (36%) of dynamic memory, leaving
a82 Disabled, 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM9
  
```

Gambar 5.37 Kode Program Mengukur Jarak Pada Virtual LCD dan LED



Gambar 5.38 Tampilan Hasil Pengukuran Jarak Pada Virtual LCD dan LED

Tabel 5.5 Pengujian Mengukur Jarak Pada Virtual LCD

No	Jarak Sesungguhnya	Hasil Baca Sensor	Virtual LCD	Virtual LED	Ket
1	17 cm	17 cm	Jarak : 17cm	Merah	Sesuai
2	19 cm	19 cm	Jarak : 19cm	Merah	Sesuai
3	22 cm	23 cm	Jarak : 23cm	Oranye	Selisih 1
4	25 cm	25 cm	Jarak: 25cm	Oranye	Sesuai

Pengujian ini dilakukan dengan beberapa kali percobaan terhadap perubahan jarak yang tidak terlalu jauh dengan perpindahan jarak secara cepat dan perpindahan secara biasa. Sama seperti hasil-hasil pengujian sensor jarak sebelumnya jika melakukan perpindahan secara cepat sensor tidak dapat membaca dengan tepat. Tetapi selisih jarak yang sebenarnya dengan yang ditampilkan tidak besar maka ini bisa dikatakan sesuai. Untuk tampilan implementasi nya sama dengan gambar 5.17 hanya berbeda pada kode programnya saja karena ini ditampilkan pada virtual LCD pada aplikasi blynk.

9. Pengujian sensor ultrasonik, LCD dan buzzer dalam melakukan pengukuran jarak.

Rangkaian elektronika yang digunakan untuk pengujian ini ada pada Gambar 5.22 keseluruhan rangkaian yang saling terhubung satu dengan yang lainnya. Pada pengujian kali ini sensor ultrasonik akan mendeteksi jarak benda yang berada di depannya lalu menampilkan pada LCD dan menyalakan buzzer ketika mencapai jarak tertentu. Pengujian ini bertujuan untuk mengetahui kinerja dari seluruh modul elektronika yang digunakan dan pengujian ini merupakan awal mula dari konsep alat *monitoring* ketinggian air. Berikut adalah kode program dan hasil dari pengujian dari keseluruhan alat:


```

keseluruhanalat$
delayMicroseconds(10);
digitalWrite(pinTriger, LOW);
duration = pulseIn(pinEcho, HIGH);
distance = (duration / 2) * 0.034;
delay(100);

lcd.setCursor(0,0);
lcd.print("Mengukur Jarak");
delay(100);
lcd.clear();

lcd.setCursor(0,1);
lcd.print("Jarak :");
lcd.print(distance);
lcd.print("cm");

if (distance <= 10) {

```

Gambar 5.39 Kode Program Keseluruhan Alat

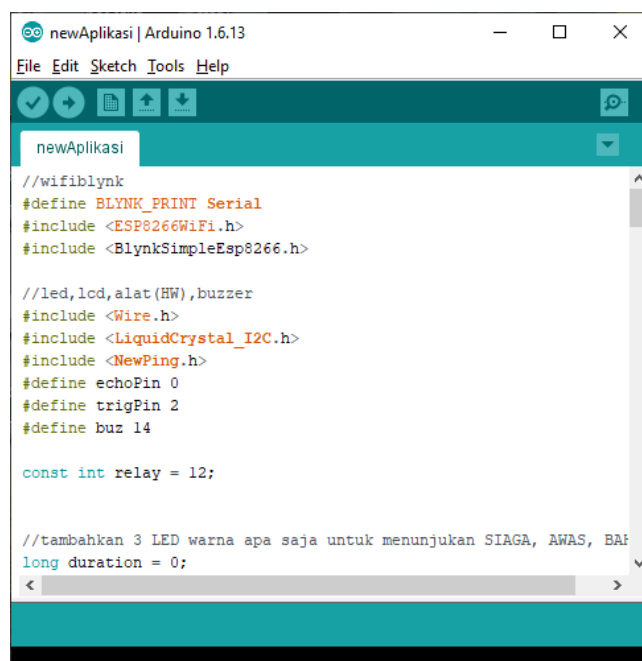
Tabel 5.6 Pengujian Keseluruhan Alat

No	Baca Sensor	Tampilan LCD	Buzzer	Jarak Sebenarnya	Ket
1	9cm	Mengukur Jarak Jarak : 9cm	Buzzer Mati	9cm	
2	13cm	Mengukur Jarak Jarak : 13cm	Buzzer Hidup Mati	13cm	
3	25cm	Mengukur Jarak Jarak : 25cm	Buzzer Hidup Mati	25cm	
4	23cm	Mengukur Jarak Jarak : 23cm	Buzzer Hidup Mati	22cm	
5	30cm	Mengukur Jarak Jarak : 30cm	Buzzer Hidup Mati	30cm	

Pengujian keseluruhan alat ini dilakukan dengan beberapa kali uji coba dengan mengukur benda dihadapan sensor dan dengan melakukan beberapa kali perpindahan. Untuk sensor ultrasonik sendiri jika terjadi perpindahan jarak yang cukup cepat sensor tidak dapat membaca dengan benar, hasil baca sensor tidak sesuai dengan jarak yang sesungguhnya dan terdapat selisih hanya saja ini terjadi sampai benda yang berada di depan sensor stabil. Selisih jarak yang terjadi akibat cepatnya perpindahan benda dapat diabaikan karena tidak terlalu signifikan dan jika diterapkan pada *monitoring* ketinggian air tidak akan terlalu berpengaruh.

10. Pengujian keseluruhan kerja alat terhadap ketinggian air.

Pengujian ini adalah akhir dari pengujian sebelumnya dan merupakan penerapan dari pembuatan alat *monitoring* ketinggian air. Pada pengujian ini akan melakukan uji coba untuk menampilkan ketinggian air dan volme air pada virtual LCD dan menyalakan lampu indikator. Sedangkan pada LCD konvensional akan menampilkan kondisi tangki dan tinggi air. Jika air berada dikondisi tertentu maka buzzer akan memberikan alarm. Berikut adalah tampilan kode program dan tampilan hasil uji coba aplikasi blynk:



```

newAplikasi | Arduino 1.6.13
File Edit Sketch Tools Help
newAplikasi
//wifiblynk
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

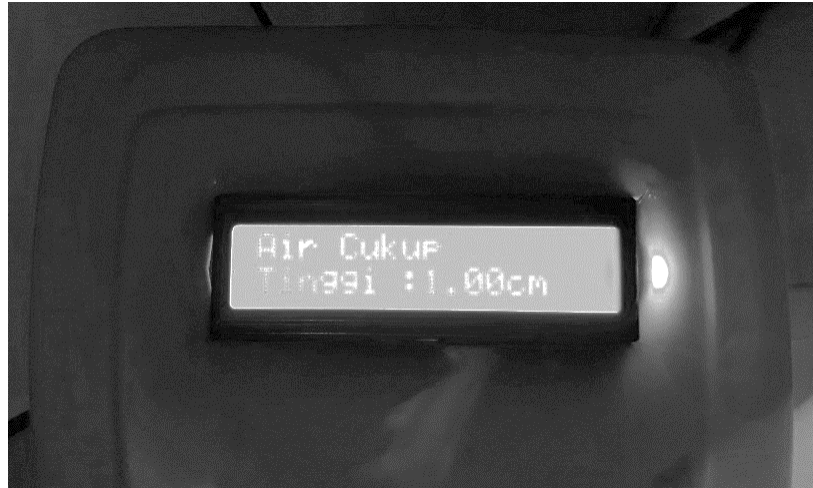
//led, lcd, alat (HW), buzzer
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
#define echoPin 0
#define trigPin 2
#define buz 14

const int relay = 12;

//tambahkan 3 LED warna apa saja untuk menunjukan SIAGA, AWAS, BAI
long duration = 0;

```

Gambar 5.40 Kode Program Keseluruhan Alat Monitoring



Gambar 5.41 Hasil Pengujian Kondisi Cukup pada LCD



Gambar 5.42 Hasil Pengujian Kondisi Kosong pada LCD

Hasil pengujian untuk mengukur ketinggian air ditampilkan seperti gambar diatas, LCD akan menampilkan tinggi air dan keadaan tangki air. Meskipun tinggi air di LCD menunjukkan 1cm tetapi itu sudah termasuk cukup karena tangki air tidak berada pada kondisi kering dan sebelumnya sudah memiliki tinggi minimal. Selain ditampilkan pada LCD hasilnya pun ditampilkan pada aplikasi *smartphone* blynk. Untuk hasil dari tampilan *smartphone* dapat dilihat pada gambar dibawah ini.



Gambar 5.43 Tampilan Hasil Pengujian Pada Blynk

Tampilan LCD yang berada pada aplikasi blynk menunjukkan volume air yang ada di dalam tangki sesuai dengan tinggi, pada gambar 5.43 menampilkan volume air sebanyak 5.42 liter dengan tinggi air 11 cm. Hasil ini didapat dari rumus menghitung volume air yaitu $(\pi \cdot r^2 \cdot t)/1000$ karena 1 liter = 1000 mili liter = 1000 cm³ jadi pada rumus tersebut untuk mendapat satuan liter harus di bagi 1000. Dan untuk mendapatkan tinggi sesungguhnya dapat menggunakan rumus: (tinggi maksimal – tinggi hasil baca sensor), setelah didapatkan maka akan ditampilkan pada LCD blynk tersebut. Selain menampilkan tinggi dan volume air pada aplikasi blynk juga menampilkan LED indikator untuk melihat kondisi tangki air yang mengacu pada ketinggian air.

Tabel 5.7 Pengujian Pengukuran Ketinggian Air

No	LCD	LCD Blynk	LED Blynk	Buzzer	Jarak Ukur Sensor
1	Air Cukup Tinggi :3cm	Volume : 1.36lt TINGGI: 3.0cm	Oranye	Buzzer Hidup Mati	33cm
2	Air Cukup Tinggi :6cm	Volume : 2.71t TINGGI: 6.0cm	Oranye	Buzzer Hidup Mati	31cm
3	Air Cukup Tinggi :11cm	Volume : 4.97lt TINGGI: 11.0cm	Oranye	Buzzer Hidup Mati	26cm
4	Air Penuh Tinggi :22cm	Volume : 9.95lt TINGGI: 22.0cm	Hijau	Buzzer Mati	15cm

5.2.2 Kesimpulan Hasil Pengujian

Kesimpulan hasil pengujian pada penelitian kali ini adalah setiap modul dapat berfungsi dengan baik dan berjalan sesuai dengan kode yang telah dibuat di Arduino IDE. Pengujian pada aplikasi blynk juga dapat dikatakan berhasil karena mampu menampilkan sesuai dengan kode program yang telah dibuat dan sesuai dengan konsep. Aplikasi blynk memiliki peran penting karena merupakan penghubung antara pengguna dengan alat agar mampu memantau secara jarak jauh dan dapat dikatakan bahwa ini *internet of things*. Dalam pengujian keseluruhan alat *monitoring* ketinggian air menampilkan volume air pada blynk sehingga pengguna tau berapa kapasitas air yang ada didalam tangki dan berapa ketinggian airnya.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan penelitian yang dilakukan penulis melalui beberapa tahapan yang dilakukan pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa:

1. Penggunaan sistem *monitoring* ketinggian air dengan teknologi *internet of things* mampu memberikan informasi yang lebih cepat pada pengguna sistem, dan informasi yang didapatkan lebih akurat daripada melakukan *monitoring* dengan cara konvensional karena dapat melakukan *monitoring* dimana saja selama alat terhubung dengan internet.
2. Dengan menggunakan sistem *monitoring* selain lebih akuratnya informasi yang didapatkan tetapi juga dapat mengefesienkan penggunaan SDM yang akan berpengaruh terhadap biaya.
3. Penggunaan aplikasi blynk sangat berguna pada sistem *monitoring* ini karena blynk menjadi *output* sekaligus menjadi server yang menjadi media komunikasi antara alat dengan *smartphone*. *Interface* nya pun sangat mudah untuk digunakan dan mudah dimengerti oleh pengguna.

6.2 Saran

Berdasarkan hasil penelitian, maka peneliti merekomendasikan atau menyarankan beberapa hal mengenai alat atau sistem *monitoring* ketinggian air agar dapat menjadi alat *monitoring* yang lebih baik. Berikut ini beberapa saran yang mungkin bisa menjadikan sistem lebih baik:

1. Untuk meningkatkan penggunaan alat *monitoring* ketinggian air ini bisa mendapatkan koneksi internet yang baik dan stabil agar *monitoring* ketinggian air bisa terus dilakukan dan dipantau dimana saja.
2. Agar lebih memudahkan dalam pelaporan jika dibutuhkan maka dapat menggunakan *database* agar setiap hasil baca sensor bisa disimpan dalam

database dan jika diperlukan data sebelumnya akan mudah mencari karena sudah tersimpan.

3. Jika alat ini dapat berguna dan diterapkan sebagai proyek pada suatu perusahaan bisa menggunakan server sendiri agar lebih tenang dan aman.
4. Lebih dikembangkan lagi agar bisa melakukan *monitoring* tidak hanya pada satu tangki saja.

Secara keseluruhan lakukan pengembangan pada sistem *monitoring* ketinggian air agar sistem menjadi lebih baik dan sangat efisien dalam penggunaan tenaga SDM, efisien dalam penggunaan waktu serta mampu memiliki sebuah *database* yang mampu menyimpan hasil *monitoring* pada waktu sebelumnya.

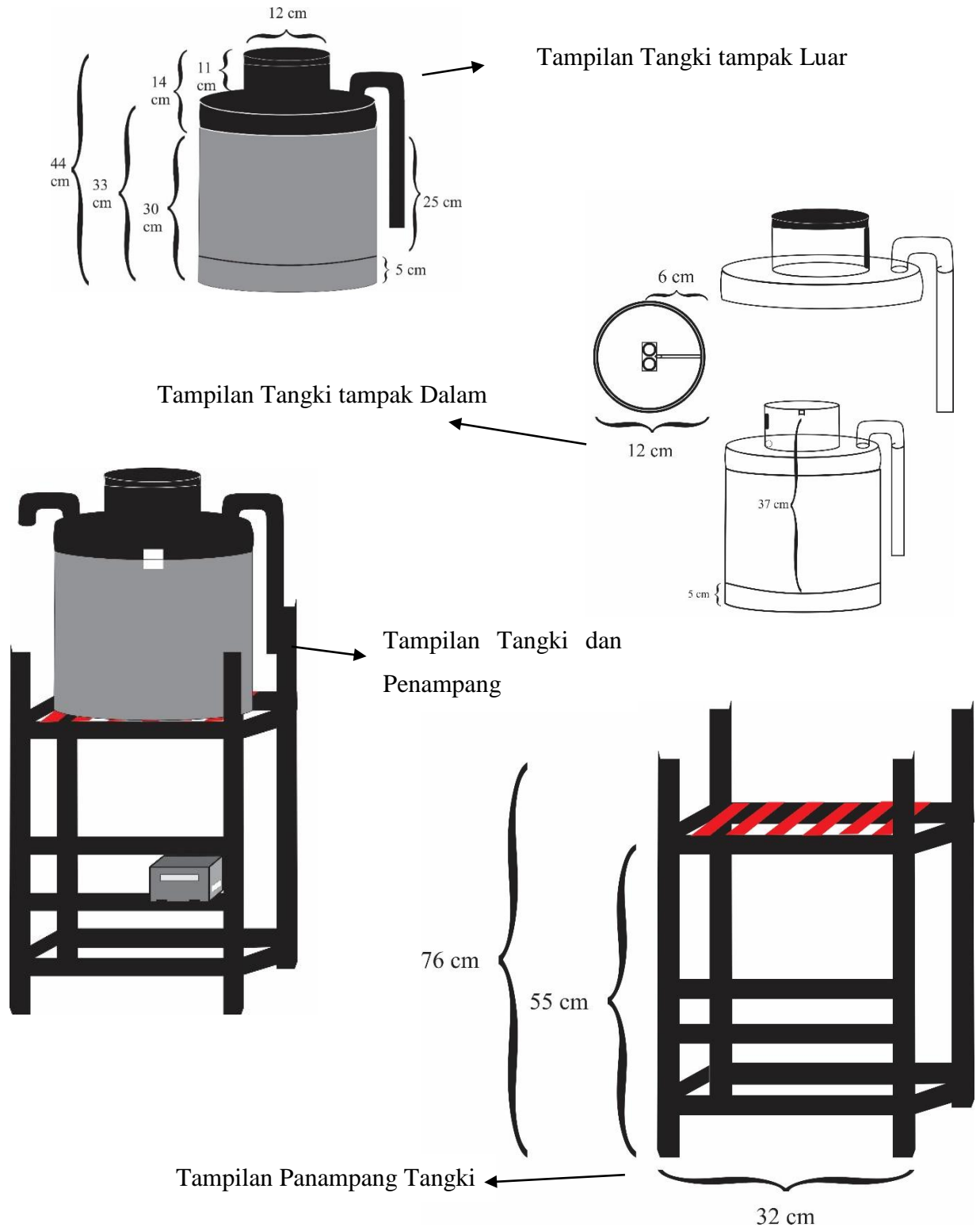
DAFTAR PUSTAKA

- Adato, L. (2015). *Monitoring 101*. United States: Solarwinds.
- Amelia, Alawiah, A., & Al Tahtawi, A. R. (2017). Sistem Kendali dan Pemantauan Ketinggian Air Pada Tangki Berbasis Sensor Ultrasonic. *Jurnal Ilmiah Manajemen Informatika dan Komputer*, 6.
- Ashton, K. (2017). *Making sense of IoT How the Internet of Things became humanity's nervous system*. Hewlett Packard Enterprise.
- Clayton, E., & Petry, F. (1983). *Monitoring for Agricultural and Rural Development Projects. Vol.2: Food & Agriculture Org.* London: The Macmillan.
- Fowler, M. (2005). *UML Distilled 3rd Ed. Panduan Singkat Bahasa Pemodelan Objek Standar*. Yogyakarta: Andi Offset.
- Handoko, T. (1995). *Pengantar Manajemen*. Yogyakarta: BPFE.
- Henderi, M. (2009/1010). Object Oriented Modelling With Unified Modeling Language (UML). *Diktat Perancangan Sistem Informasi*, 5-6.
- Humas. (2015, July 24). *Pengertian Monitoring Dan Evaluasi Kebijakan Pemerintah*. Retrieved from Sekretariat Kabinet Republik Indonesia:
- McLeod, R. (2008). *Sistem Informasi Manajemen Edisi.10*. Jakarta: Bhuana Ilmu Komputer.
- Mehta, M. (2015). ESP8266: Breakthrough in Wireless Sensor Networks And Internet of Things. *International Journal of Electronics and Communication Engineering & Technology (IJECE)*, 07-11. Retrieved May 4, 2019.
- Nugroho, A. (2010). *Rekayasa Perangkat Lunak Berorientasi Objek Dengan Metode USDP*. Yogyakarta: Andi.
- Permana, A., Triyanto, D., & Rismawan, T. (2015). Rancang Bangun Sistem Monitoring Volume dan Pengisian Air Menggunakan Sensor Ultrasonik Berbasis Mikrokontroler AVR ATmega8. *Jurnal Coding, Sistem Komputer Untan*, 1-12.
- Prabawa, S. A. (2015, October 3). *Project Board*. Retrieved may 4, 2019, from Satria Agung Prabawa ilearning: <https://satria1996.ilearning/?p=2485>
- Rosa, A. S., & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: INFORMATIKA.
- Rubin, F. (1995). *A Basic Guide to Evaluation for Development Workers*. UK and Ireland: Oxfam.

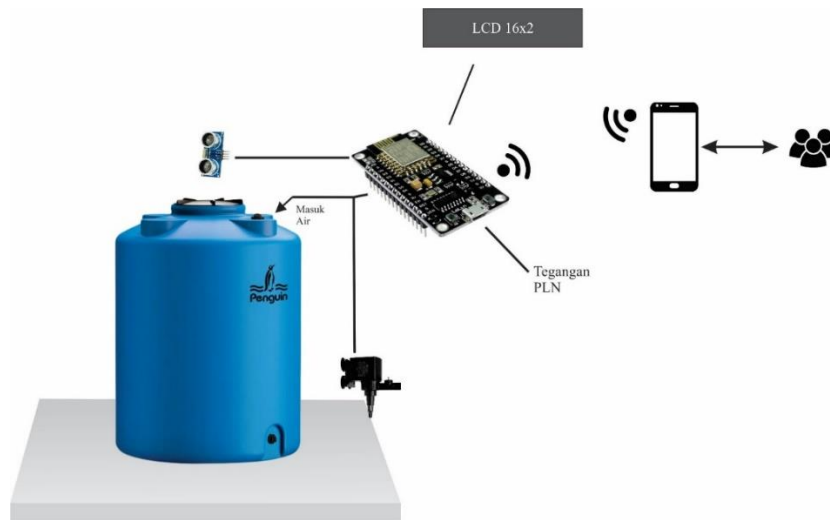
- Schwartz, M. (2016). Internet of Things With ESP8266. In M. Schwartz, *Build Amazing Internet of Things Projects Using the ESP8266 Wi-Fi Chip* (p. 209). Birmingham-Mumbai: Packt Publishing Ltd.
- Simanjuntak, M. (2012). *Rancang bangun teknologi Pemurni air*. Sumatera Utara: Universitas Sumatera Utara.
- Sulistiyowati, R., Sujono, H. A., & Musthofa, A. K. (2015). Sistem Pendeteksi Banjir Berbasis Sensor Ultrasonic dan Mikrokontroler Dengan Media Komunikasi SMS Gateway. *IoT*, 10.
- Sunarti. (2009). *Metode-metode Penelitian*. Bandung. Alfabeta
- Ulumuddin, Sudrajat, M., Rachmildha, T., Hamidi, E., & Ismail, N. (2017). Prototipe Sistem Monitoring Air Pada Tangki Berbasis Internet of Things Menggunakan NodeMCU ESP8266 Dan Sensor Ultrasonik. *ISBN: 978-602-512-810-3*, 6.
- Yudhanto, Y. (2015, May 15). *Apa itu IOT (Internet of Things)?* Retrieved 5 4, 2019, from IlmuKomputer.com: <https://ilmukomputer.org/2015/05/15/apa-itu-iot-internet-of-things/>

LAMPIRAN

Lampiran 1 Sketsa Pembuatan Alat *Monitoring* Ketinggian Air



Lampiran 2 Konsep Alat *Monitoring* Ketinggian Air



Sensor ultrasonik akan mendeteksi ketersediaan air pada tangki, jika tangki kosong maka akan menampilkan keterangan pada LCD dan mengirimkan notifikasi ke android dan akan ada *buzzer* yang menyala terus menerus, setelah itu maka pompa akan aktif dan mengisi air hingga ketinggian yang telah di tentukan, pada penelitian ini maksimal ketinggian air 28 cm untuk menghindari air meluap dari tangki. Jika ketinggian air sudah mencapai 28 cm maka otomatis pompa akan mati dan *buzzer* akan hidup mati menandakan air telah terisi penuh.

Dikategorikan tangki kosong yaitu jarak permukaan dan sensor sekitar 42 cm – 37 cm atau ketinggian air 0–5cm, dan kategori air pada tangki cukup yaitu jarak antara sensor dan permukaan 36 cm – 14 cm atau ketinggian air sudah mencapai 28cm. Dan dikategorikan air pada tangki penuh jika jarak antara sensor dan permukaan kurang dari 14 cm, ketinggian tidak mencapai titik paling atas sensor untuk mengurangi sensor terendam air.

Sensor ultrasonik yang terhubung dengan NodeMCU berguna untuk mengukur ketinggian air secara *realtime* sehingga tidak perlu mengukur secara manual. NodeMCU akan mengirimkan data melalui aplikasi blynk yang telah terinstal di android jika air kosong, cukup atau penuh maka akan memberikan notifikasi kepada pengguna.

Lampiran 3 Uji Coba Alat Monitoring (Awal Percobaan)



Dari hasil percobaan yang telah dilakukan, terkadala pada pompa air yang lemah sehingga jika menggunakan besi penyangga air tidak dapat mengisi tangki seperti yang diharapkan. Tetapi jika mengisi tangki tanpa menggunakan besi penyangga air dapat masuk dengan cepat kedalam tangki.

Kendala lainnya terjadi pada *power supply* jika tegangan di dapat dari sumber PLN maka pin GND dan pin VCC sensor ultrasonik harus dilepas terlebih dahulu agar bisa memberikan *supply* kepada mikrokontrolernya.

Pembacaan sensor pada lcd masih sesekali tidak tepat karena kondisi permukaan air yang tidak stabil atau tidak tenang. Yang mengakibatkan LCD salah menampilkan jarak antara baca sensor dan permukaan air. Percobaan ini dilakukan saat tidak menggunakan penyangga besi

Ada 2 pilihan untuk modifikasi tutup tangki, yang pertama dengan ukuran yang cukup tinggi dan yang kedua dengan ukuran yang relatif lebih pendek, yang dipilih adalah yang relatif pendek karena cukup stabil dalam pembacaan sensor.

Lampiran 4 Form Wawancara pada Penelitian *Monitoring Ketinggian Air*

Wawancara dilakukan pada seorang teman yang mengerti tentang monitoring ketinggian air. Berikut adalah identitas narasumber dan form wawancara:

Nama : Sukma Cahyana F
 Hari / Tanggal : Rabu, 24 April 2019
 Waktu : 8.45 WIB
 Tempat : via Whatsapp

No	Pertanyaan	Ya	Tidak
1	Apakah krisis air sangat mengganggu proses pendistribusian pada masyarakat?	Ya	
2	Apakah ada antisipasi khusus yang dilakukan untuk mengatasi krisis air yang terjadi?		Tidak
3	Untuk memenuhi kebutuhan masyarakat perihal air apakah pendistribusian air dapat mencapai lebih dari 500liter dalam satu bulan?	Ya	
4	Apakah 500liter itu untuk satu keluarga? Dan apakah itu termasuk ke dalam pemborosan?		Tidak
5	Apakah penghematan air perlu dilakukan karena menilik bahwa sumber air sudah mulai kering?	Ya	
6	Apakah masyarakat sudah mulai melakukan penghematan air dengan menggunakan penampung tersebut?	Ya	
7	Menurut anda apakah dengan menggunakan penampung dapat menjadi sarana penghematan Air?	Ya	
8	Apakah penyimpanan penampung air yang disimpan diatas bangunan cukup efektif?	Ya	
9	Menurut anda apakah seseorang masih melakukan pengukuran pada ketinggian air?		Tidak
10	Lalu apakah melakukan monitoring terhadap penampung air perlu dilakukan?	Ya	
11	Apakah sudah ada alat monitoring yang bisa mengukur ketinggian air dan mengetahui berapa volume?		Tidak
12	Apakah alat yang digunakan monitoring masih bersifat mesin seperti <i>switch</i> pompa atau bandul?	Ya	
13	Apakah tangki dengan ukuran 250liter cukup untuk memenuhi kebutuhan-kebutuhan rumah 1 keluarga?	Ya	

14	Apakah sudah ada alat pengukur ketinggian air selain menggunakan bandul yang bersifat aplikatif?		Tidak
15	Jika di setiap rumah menggunakan bandul apakah ini sudah efektif dalam melakukan penghematan air?	Ya	
16	Apakah anda pernah mendengar soal IoT (internet of thing)?	Ya	
17	Apakah sudah pernah ada orang yang mencoba melakukan pembuatan alat monitoring dengan teknologi itu?		Tidak
18	Apakah menurut anda jika penggunaan IoT pada monitoring ketinggian air diterapkan akan efektif?	Ya	
19	Dengan internet yang memadai sekarang, monitoring dapat dilakukan dimana saja dan kapan saja. apakah ini dapat menjadi sebuah terobosan baru di dunia monitoring?	Ya	
20	Apakah dengan menggunakan bandul dapat mengetahui jumlah volume air secara <i>realtime</i> ?		Tidak
21	Apakah penggunaan IoT dibidang monitoring air akan berjalan?	Ya	
22	IoT adalah salah satu teknologi yang saat ini sedang digandrungi untuk memudahkan pekerjaan manusia. Jika ini berhasil diterapkan pada monitoring ketinggian air dan memiliki fungsi mirip dengan bandul hanya saja alat ini bersifat <i>realtime</i> dan mampu menampilkan volume air dan ketinggiannya secara jelas. Apakah akan sangat berguna bagi masyarakat	Ya	
JAWABAN		15	7

Dengan melihat hasil wawancara diatas penulis memutuskan untuk melakukan pembuatan *prototype monitoring* ketinggian air berbasis IoT (*internet of things*) yang mampu menampilkan volume air, tinggi air secara *realtime* dan bisa diakses dimana saja tanpa takut air akan tumpah.

Lampiran 5 Kumpulan Sketch Program Alat *Monitoring*

Sketch Program Mendeteksi IP Address

```
#include <ESP8266WiFi.h>

const char* ssid = "Angga"; *ssid bebas
const char* pass = "private12";
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  delay(10);
  Serial.println();

  Serial.print("Connecting...");
  Serial.println(".....");

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi Connected");
  Serial.println("IP ADDRESS: ");
  Serial.println(WiFi.localIP());
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Sketch Program LCD dengan Buzzer

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#define buz 14

LiquidCrystal_I2C lcd(0x27,16,2);

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

  lcd.begin();
  Wire.begin(D2,D1);
  lcd.setCursor(0,0);
  lcd.print("ANGGA TRIATNA");
  delay(2000);
  lcd.clear();
}

void loop() {
  // put your main code here, to run repeatedly:
  lcd.clear();
  lcd.print("Buzzer Mati");
  digitalWrite(14, LOW);
  delay(1000);
  lcd.clear();
  lcd.print("Buzzer Menyala");
  digitalWrite(14, HIGH);
  delay(1000);
}
```


Sketch Program Sensor Ultrasonik

```
#include <NewPing.h>
#define echoPin 0
#define trigPin 2

long duration = 0;
long distance, ultrasensor;;

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  sensorUltrasonic(trigPin, echoPin);
  ultrasensor = distance;

  Serial.println(ultrasensor);
}

void sensorUltrasonic(int pinTriger, int pinEcho)
{
  Serial.println(distance);
  digitalWrite(pinTriger, LOW);
  delayMicroseconds(2);
  digitalWrite(pinTriger, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTriger, LOW);
  duration = pulseIn(pinEcho, HIGH);
  distance = (duration / 2) * 0.034;

  Serial.print("Jarak ");
  Serial.print(distance);
  Serial.print("cm");

  delay(1000);
}
```

Sketch Program LCD dengan Sensor

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
#define echoPin 0
#define trigPin 2
long duration = 0;
long distance, ultrasensor;;

LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  lcd.begin();
  Wire.begin(D2,D1);
  lcd.setCursor(0,0);
  lcd.print("ANGGA TRIATNA");
  delay(2000);
  lcd.clear();
}

void loop() {
  // put your main code here, to run repeatedly:
  sensorUltrasonic(trigPin, echoPin);
  ultrasensor = distance;

  Serial.println(ultrasensor);
}

void sensorUltrasonic(int pinTriger, int pinEcho)
{
  Serial.println(distance);
  digitalWrite(pinTriger, LOW);
  delayMicroseconds(2);
  digitalWrite(pinTriger, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTriger, LOW);
  duration = pulseIn(pinEcho, HIGH);
  distance = (duration / 2) * 0.034;

  lcd.setCursor(0,1);
  lcd.print("Jarak :");
  lcd.print(distance);
  lcd.print("cm");

  delay(1000);
}

```

Sketch Program Virtual LCD dan LED

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "FyQh_plZhCksfgTgPYGix9JYY0Yntj0v";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Angga";
char pass[] = "private12";

WidgetLCD lcd(V0);
WidgetLED led(V1);
WidgetLED led1(V2);
WidgetLED led2(V3);

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
  lcd.clear();
}

void loop()
{
  lcd.print(0,0, "Mengukur Jarak");
  lcd.print(0,1, "Jarak : 19cm");
  led.on();
  led2.off();
  led1.off();
  delay(200);
  Blynk.run();
}
```

Sketch Program Virtual LCD dan LED menampilkan Hasil Sensor

```

//wifiblynk
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

//led,lcd,alat(HW),buzzer
#include <NewPing.h>
#define echoPin 0
#define trigPin 2

//tambahkan 3 LED warna apa saja untuk menunjukan SIAGA, AWAS,
BAHAYA
long duration = 0;
long distance, ultrasensor;

char auth[] = "PuCs0t1MY1Ta0iGN1pkRTBqtMywRIzrl";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Angga";
char pass[] = "private12";

WidgetLCD lcd1(V3);
WidgetLED led(V0);
WidgetLED led1(V1);
WidgetLED led2(V2);

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);

  Blynk.begin(auth, ssid, pass);
  lcd1.clear();
}

void loop() {
  // put your main code here, to run repeatedly:
  sensorUltrasonic(trigPin, echoPin);
  ultrasensor = distance;

  Serial.println(ultrasensor);
}

void sensorUltrasonic(int pinTriger, int pinEcho)

```

```
{  
  Serial.println(distance);  
  digitalWrite(pinTriger, LOW);  
  delayMicroseconds(2);  
  digitalWrite(pinTriger, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(pinTriger, LOW);  
  duration = pulseIn(pinEcho, HIGH);  
  distance = (duration / 2) * 0.034;  
  
  delay(100);  
  
  lcd1.print(0,1, "Jarak : ");  
  lcd1.print(8,1, distance);  
  lcd1.print(14,1, "cm");  
  
  if (distance <= 20) {  
    lcd1.print(0,0, "Mengukur Jarak");  
    led.on();  
    led1.off();  
    led2.off();  
    delay(100);  
  }  
  
  else if (distance >= 37) {  
    lcd1.print(0,0, "Mengukur Jarak");  
    led.off();  
    led1.on();  
    led2.off();  
    delay(100);  
  }  
  
  else if (distance > 50) {  
    lcd1.print(0,0, "Mengukur Jarak");  
    led.off();  
    led1.off();  
    led2.on();  
    delay(100);  
  }  
  delay(1000);  
}
```

Sketch Program Sebelum IoT

```

//led,lcd,alat(HW),buzzer
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
#define echoPin 0
#define trigPin 2
#define buz 14
const int relay = 12;

//tambahkan 3 LED warna apa saja untuk menunjukkan SIAGA, AWAS, BAHAYA
long duration = 0;
long distance, ultrasensor;
LiquidCrystal_I2C lcd(0x27,16,2);

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  pinMode (buz, OUTPUT);
  pinMode (relay, OUTPUT);
  digitalWrite(relay, HIGH);

  lcd.begin();
  Wire.begin(D2,D1);
  lcd.setCursor(0,0);
  lcd.print("ANGGA TRIATNA");
  delay(2000);
  lcd.clear();
}

void loop() {
  // put your main code here, to run repeatedly:
  sensorUltrasonic(trigPin, echoPin);
  ultrasensor = distance;

  Serial.println(ultrasensor);
}

void sensorUltrasonic(int pinTriger, int pinEcho)
{
  Serial.println(distance);
  digitalWrite(pinTriger, LOW);
  delayMicroseconds(2);
  digitalWrite(pinTriger, HIGH);
  delayMicroseconds(10);
  digitalWrite(pinTriger, LOW);
  duration = pulseIn(pinEcho, HIGH);
  distance = (duration / 2) * 0.034;
  delay(100);

  lcd.setCursor(0,0);
  lcd.print("Mengukur Jarak");
}

```

```

delay(100);
lcd.clear();

lcd.setCursor(0,1);
lcd.print("Jarak :");
lcd.print(distance);
lcd.print("cm");

if (distance <= 10) {
  lcd.setCursor(0,0);
  delay(200);
  lcd.print("Buzzer Mati");
  delay(100);

  delay(100);
  digitalWrite(14, LOW);
  delay(100);
  digitalWrite(relay, 1);
  delay(100);
}

else if (distance >= 35) {
  lcd.setCursor(0,0);
  delay(200);
  lcd.print("Buzzer Menyala");
  delay(100);

  digitalWrite(14, HIGH);
  delay(100);
  digitalWrite(relay, 0);
  delay(100);
}

else if ((distance >10) && (distance <35)){
  lcd.setCursor(0,0);
  delay(200);
  lcd.print("Buzzer Hidup Mati");
  delay(100);

  delay(100);
  digitalWrite(14, HIGH);
  delay(100);
  digitalWrite(14, LOW);
  delay(100);
  digitalWrite(14, HIGH);
  delay(100);
  digitalWrite(14, LOW);
  delay(100);
  digitalWrite(relay, 0);
  delay(100);
}
delay(1000);
}

```

Sketch Program IoT

```

//wifiblynk
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

//led,lcd,alat(HW),buzzer
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <NewPing.h>
#define echoPin 0
#define trigPin 2
#define buz 14

const int relay = 12;

//tambahkan 3 LED warna apa saja untuk menunjukkan SIAGA, AWAS, BAHAYA
long duration = 0;
long distance, ultrasensor;
float volume, jarak;

LiquidCrystal_I2C lcd(0x27,16,2);

char auth[] = "PuCsOt1MY1Ta0iGN1pkRTBqtMywRIzr1";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Angga";
char pass[] = "private12";

WidgetLCD lcd1(V3);
WidgetLED led(V0);
WidgetLED led1(V1);
WidgetLED led2(V2);

void setup() {
  // put your setup code here, to run once:
  Serial.begin (9600);
  pinMode (trigPin, OUTPUT);
  pinMode (echoPin, INPUT);
  pinMode (buz, OUTPUT);
  pinMode (relay, OUTPUT);
  digitalWrite(relay, HIGH);

  Blynk.begin(auth, ssid, pass);
  lcd1.clear();

  lcd.begin();
  Wire.begin(D2,D1);
  lcd.setCursor(0,0);
  lcd.print("ANGGA TRIATNA");
  delay(2000);

```



```

    lcd.clear();
}

void loop() {
    // put your main code here, to run repeatedly:
    sensorUltrasonic(trigPin, echoPin);
    ultrasensor = distance;

    Serial.println(ultrasensor);
}

void sensorUltrasonic(int pinTriger, int pinEcho)
{
    Serial.println(distance);
    digitalWrite(pinTriger, LOW);
    delayMicroseconds(2);
    digitalWrite(pinTriger, HIGH);
    delayMicroseconds(10);
    digitalWrite(pinTriger, LOW);
    duration = pulseIn(pinEcho, HIGH);
    distance = (duration / 2) * 0.034;

    jarak = (37-distance);
    volume = ((3.14 * (12*12) * jarak)/1000);

    delay(100);

    lcd.setCursor(0,0);
    lcd.print("KETINGGIANA AIR");
    delay(100);
    lcd.clear();

    lcd.setCursor(0,1);
    lcd.print("Tinggi :");
    lcd.print(jarak);
    lcd.print("cm");

    if (distance >37) {
        volume = 0;
        jarak = 0;
        lcd1.print(0,0, "Volume :");
        lcd1.print(10,0, volume);
        lcd1.print(14,0, "lt");
        lcd1.print(0,1, "TINGGI:");
        lcd1.print(8,1, jarak);
        lcd1.print(14,1, "cm");
    } else{
        lcd1.print(0,1, "TINGGI:");
        lcd1.print(8,1, jarak);
        lcd1.print(14,1, "cm");
    }

    if (distance <= 15) {
        lcd.setCursor(0,0);
        delay(200);
        lcd.print("Air Penuh");
    }
}

```

```

    delay(100);
    lcd1.print(0,0, "Volume :");
    lcd1.print(10,0, volume);
    lcd1.print(14,0, "lt");

    delay(100);
    digitalWrite(14, LOW);
    delay(100);
    digitalWrite(relay, 1);
    delay(100);
    led.on();
    led1.off();
    led2.off();
    delay(100);
}

else if (distance >= 37) {
    lcd.setCursor(0,0);
    lcd.print("Air Kosong");
    lcd1.print(0,0, "Volume :");
    lcd1.print(10,0, volume);
    lcd1.print(14,0, "lt");

    digitalWrite(14, HIGH);
    delay(100);
    digitalWrite(relay, 0);
    delay(100);
    led.off();
    led1.on();
    led2.off();
    delay(100);
}

else if ((distance >15) && (distance <37)){
    lcd.setCursor(0,0);
    lcd.print("Air Cukup");
    lcd1.print(0,0, "Volume :");
    lcd1.print(10,0, volume);
    lcd1.print(14,0, "lt");

    delay(100);
    digitalWrite(14, HIGH);
    delay(100);
    digitalWrite(14, LOW);
    delay(100);
    digitalWrite(14, HIGH);
    delay(100);
    digitalWrite(14, LOW);
    delay(100);
    digitalWrite(relay, 0);
    led.off();
    led1.off();
    led2.on();
    delay(100);
}
delay(1000);
}

```

RIWAYAT HIDUP PENULIS



ANGGA TRIATNA lahir di Bandung, pada 17 Juni 1996. Merupakan anak pertama dari pasangan Entis Sutisna Wijaya dan Ai Tarmanah, yang beralamat di Kp. Cibeet Rt.01 Rw.04 Ds. Cibeet Kec. Ibun Kab. Bandung. Memulai Pendidikan di SD Negeri Pasirbitung (2002-2008), kemudian melanjutkan pendidikan ke SMP Negeri 1 Ibun (2008-2011), serta melanjutkan masa SMA di SMA Negeri 1 Majalaya (2011-2014). Untuk meraih gelar sarjana (S1), penulis menempuh pendidikan di UNIVERSITAS BALE BANDUNG (UNIBBA) Fakultas Teknologi Informasi (FTI) dengan Program Studi Teknik Informatika (IF) sejak tahun 2015. Penulis juga pernah aktif sebagai anggota HIMA-IF. Penulis juga pernah ikut serta sebagai peserta PILMAPRES tingkat Universitas dan mewakili Universitas di tingkat KOPERTIS IV Jabar Rayon VI di Universitas Komputer Indonesia pada tahun 2018.