



**SISTEM KONTROL KETINGGIAN AIR
PADA SAWAH BAWANG MERAH BERBASIS IOT**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi
Jenjang Program Diploma Tiga

Oleh :

Nama	NIM
Zakaria	(18041094)

**PROGRAM STUDI DIII TEKNIK KOMPUTER
POLITEKNIK HARAPAN BERSAMA TEGAL
2021**

HALAMAN PERNYATAAN KEASLIAN
TUGAS AKHIR UNTUK KEPERLUAN AKADEMIS

Yang bertanda tangan dibawah ini :

Nama : Zakaria
NIM : 18041094
Jurusan / Program Studi : D III Teknik Komputer
Jenis Karya : Tugas Akhir

Adalah mahasiswa Progam Studi DIII Teknik Komputer Politeknik Harapan Bersama, dengan ini kami menyatakan bahwa laporan Tugas Akhir yang berjudul **“SISTEM KONTROL KETINGGIAN AIR PADA SAWAH BAWANG MERAH BERBASIS IOT”**.

Merupakan hasil pemikiran dan kerjasama sendiri secara orisinil dan saya susun secara mandiri dan tidak melanggar kode etika hak karya cipta. Pada pelaporan Tugas Akhir ini juga bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademika tertentu di suatu perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau disebutkan dalam daftar pustaka.

Apabila dikemudian hari ternyata Laporan Tugas Akhir ini terbukti melanggar kode etik karya cipta atau merupakan karya yang dikategorikan mengandung unsur plagiatisme, maka saya bersedia untuk melakukan penelitian baru dan menyusun laporannya sebagai Laporan Tugas Akhir, sesuai ketentuan yang berlaku.

Demikian Pernyataan ini Saya buat dengan sebenarnya dan sesungguhnya.

Tegal, ... Juli 2021

Yang membuat pernyataan,


Zakaria

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

Sebagai civitas akademika Politeknik Harapan Bersama Tegal, Saya yang bertanda tangan di bawah ini :

Nama : Zakaria
NIM : 18041094
Jurusan / Program Studi : D III Teknik Komputer
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Harapan Bersama Tegal **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas Tugas Akhir kami yang berjudul :

“SISTEM KONTROL KETINGGIAN AIR PADA SAWAH BAWANG MERAH BERBASIS IOT”

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti *Noneksklusif* ini Politeknik Harapan Bersama Tegal berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan Tugas Akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Tegal

Pada Tanggal : Juli 2021

Yang menyatakan



Zakaria

HALAMAN PERSETUJUAN

Tugas Akhir (TA) yang berjudul “**SISTEM KONTROL KETINGGIAN AIR PADA SAWAH BAWANG MERAH BERBASIS IOT**” yang disusun oleh Zakaria, NIM 18041094 telah mendapat persetujuan pembimbing dan siap dipertahankan di depan tim penguji Tugas Akhir (TA) Progam Studi DIII Teknik Komputer Politeknik Harapan Bersama Tegal.

Tegal, April 2021

Menyetujui,

Pembimbing I



Mohammad Humam, M.Kom
NIPY. 12.002.007

Pembimbing II



Abdul Basit, S.Kom., MT
NIPY. 01.015.198


HALAMAN PENGESAHAN

Judul : "SISTEM KONTROL KETINGGIAN AIR PADA
SAWAH BAWANG MERAH BERBASIS IOT"
Oleh : ZAKARIA
NIM : 18041094
Program Studi : Teknik Komputer
Jenjang : Diploma III

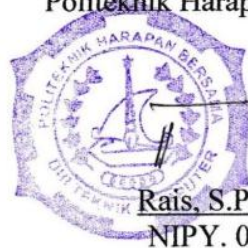
Dinyatakan LULUS setelah dipertahankan didepan Tim Penguji Tugas Akhir Program Studi DIII Teknik Komputer Politeknik Harapan Bersama Tegal.

Tegal, Juli 2021

Tim Penguji:

	Nama	Tanda Tangan
1. Ketua	: Arfan Haqiqi Sulasmoro, M.Kom	1. 
2. Anggota I	: Nurohim, S.ST, M.Kom	2. 
3. Anggota II	: Abdul Basit, S.Kom., MT	3. 

Mengetahui,
Ketua Program Studi DIII Teknik Komputer,
Politeknik Harapan Bersama Tegal


Rais, S.Pd., M.Kom
NIPY. 07.011.083

ABSTRAK

Adanya pemanasan global mengakibatkan terjadinya perubahan iklim yang tidak menentu beberapa tahun belakangan ini. Hal ini merupakan ancaman yang dihadapi dalam usaha pengembangan usaha budidaya bawang merah. Musim kemarau yang berkepanjangan menimbulkan adanya kekeringan di banyak lahan pertanian yang berujung pada gagal panen akibat kekurangan air. Berubahnya waktu musim hujan juga menyulitkan petani untuk menentukan masa tanam yang tepat. Demikian juga adanya perubahan cuaca yang ekstrem mengakibatkan pertumbuhan tanaman terganggu sehingga menimbulkan dampak menurunnya kualitas dan produktivitas lahan. Mengatasi permasalahan diatas, maka penulis mempunyai ide untuk merancang dan membuat suatu alat yang dapat mengurangi beberapa kelemahan yang terjadi untuk meringankan permasalahan yang dihadapi petani yaitu sebuah sistem kontrol ketinggian air pada sawah bawang merah berbasis IoT.

Kata kunci: Irigasi Otomatis Bawang Merah.

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Allah SWT, Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya hingga terselesaikannya laporan Tugas Akhir dengan Judul **“SISTEM KONTROL KETINGGIAN AIR PADA SAWAH BAWANG MERAH BERBASIS IOT”**.

Tugas Akhir merupakan suatu kewajiban yang harus dilaksanakan untuk memenuhi salah satu syarat kelulusan dalam mencapai derajat Ahli Madya Komputer pada Program Studi DIII Teknik Komputer Politeknik Harapan bersama Tegal.

Selama melaksanakan penelitian dan kemudian tersusun dalam laporan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan, dukungan dan bimbingan.

Pada kesempatan ini, tidak lupa diucapkan terima kasih yang sebesar – besarnya kepada :

1. Bapak Nizar Suhendra, SE, MPP selaku Direktur Politeknik Harapan Bersama Tegal.
2. Bapak Rais, S.Pd, M.Kom selaku Ketua Program Studi D III Teknik Komputer Politeknik Harapan Bersama Tegal.
3. Bapak Mohammad Humam, M.Kom selaku dosen pembimbing I.
4. Bapak Abdul Basit, S.Kom,.MT selaku dosen pembimbing II.
5. Semua pihak yang telah mendukung, membantu serta mendoakan penyelesaian laporan Tugas Akhir ini.

Semoga laporan Tugas Akhir ini dapat memberikan sumbangan untuk pengembangan ilmu pengetahuan dan teknologi.

Tegal, Agustus 2021

DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN KEASLIAN	ii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PENGESAHAN.....	v
ABSTRAK.....	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN.....	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	5
1.4 Tujuan dan Manfaat	5
1.4.1 Tujuan.....	5
1.4.2 Manfaat.....	5
1.5 Sistematika Penulisan Laporan	6
BAB II TINJAUAN PUSTAKA.....	9
2.1 Penelitian Terkait	9
2.2 Landasan Teori.....	10
2.2.1 Irigasi.....	10
2.2.2 Monitoring.....	11
2.2.3 IOT	11
2.2.4 NodeMCU 8266	13
2.2.5 Laravel.....	15
2.2.6 Arduino IDE.....	18
2.2.7 Visula Studio Code.....	19
2.2.8 Database	21
2.2.9 HTML.....	22
2.2.10 CSS.....	22
2.2.11 PHP.....	22
2.2.12 Java Script	23
2.2.13 Flow Chart.....	24
2.2.14 UML	25
2.2.15 Sensor Ultrasonic HC-SR04.....	32
2.2.16 Servo.....	35

2.2.17 Pompa Air	38
2.2.18 Relay.....	39
2.2.19 Adaptor / Power Supply	39
2.2.20 Kabel Jumper.....	41
BAB III METODOLOGI PENELITIAN.....	42
3.1. Prosedur Penelitian.....	42
3.1.1. Rencana / <i>Planning</i>	42
3.1.2. Data Analisis	43
3.1.3. Rancangan dan Desain	43
3.1.4. Implementasi	45
3.2. Metode Pengumpulan Data	45
3.2.1 Observasi.....	45
3.2.2 Wawancara	49
3.3. Tools.....	51
3.4. Waktu dan Tempat Penelitian	51
3.4.1 Waktu Penelitian	51
3.4.2 Tempat Penelitian.....	51
BAB IV ANALISA DAN PERANCANGAN.....	52
4.1 Analisa Permasalahan	52
4.2 Analisa Kebutuhan Sistem	53
4.2.1 Analisa Kebutuhan <i>Hardware</i>	53
4.2.2 Analisa Kebutuhan <i>Software</i>	54
4.3 Perancangan Sistem	54
4.3.1 Perancangan Diagram Blok Sistem.....	54
4.3.2 Flowchart.....	55
4.3.3 Desain Input Output	59
BAB V IMPLEMENTASI SISTEM.....	62
5.1 Implementasi Sistem	62
5.1.1 Implementasi Perangkat Keras.....	62
5.1.2 Implementasi Perangkat Lunak dan Instalasi Aplikasi	62
5.2 Tahap Instalasi	63
BAB VI KESIMPULAN DAN SARAN	71
6.1 Kesimpulan	71
6.2 Saran.....	71
DAFTAR PUSTAKA	72
LAMPIRAN	

DAFTAR TABEL

Halaman

Tabel 2. 1 NodeMCU 8266.....	14
Tabel 2. 2 <i>Flowchart Program</i>	24
Tabel 2. 3 Simbol-simbol <i>use case</i>	26
Tabel 2. 4 simbol-simbol <i>activity Diagram</i>	28
Tabel 2. 5 simbol-simbol <i>sequence Diagram</i>	28
Tabel 2. 6 simbol-simbol <i>class diagram</i>	29
Tabel 2. 7 simbol-simbol <i>component diagram</i>	30
Tabel 2. 8 simbol-simbol <i>deployment diagram</i>	31
Tabel 5. 1 Alat / Komponen yang digunakan	64
Tabel 5. 2 Pembagian Pin <i>Nodemcu</i>	64
Tabel 5. 3 Penjelasan Pengujian Sistem.....	66
Tabel 5. 4 Hasil pengujian alat.....	67
Tabel 5. 5 Hasil pengujian unjuk kerja keseluruhan alat	67

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 NodeMCU 8266.....	14
Gambar 2. 2 Skematik Posisi Pin Node MCU	15
Gambar 2. 3 Laravel.....	16
Gambar 2. 4 Arduino IDE.....	19
Gambar 2. 5 Tampilan Visual Studio Code	21
Gambar 2. 6 Sensor <i>Ultrasonic</i>	33
Gambar 2. 7 Motor <i>Servo</i>	36
Gambar 2. 8 Pompa Air	39
Gambar 2. 9 Module Relay 2 Chanel.....	39
Gambar 2. 10 Power Supply	40
Gambar 2. 11 Kabel <i>Jumper</i>	41
Gambar 3. 1 Blok Diagram Sistem (Pompa Air Otomatis)	44
Gambar 3. 2 Blok Diagram Sistem (<i>Monitoring</i>)	44
Gambar 3. 3 Ladang Tanaman Bawang Merah	46
Gambar 3. 4 Jalur keluar masuk air di embung	46
Gambar 3. 5 Kondisi air di embung yang rendah	47
Gambar 3. 6 Jalur irigasi menuju sawah / lahan tanaman bawang merah	47
Gambar 3. 7 Lokasi Embung Observasi 1	48
Gambar 3. 8 Lokasi Embung Observasi 2	49
Gambar 3. 9 Wawancara dengan petani yang sedang menanam bawang.....	50
Gambar 3. 10 Wawancara dengan petani yang sedang memompa air.....	50
Gambar 4. 1 Diagram blok Rangkaian.....	54
Gambar 4. 2 <i>Flowchart</i> Sistem kontrol ketinggian air sawah.....	56
Gambar 4. 3 <i>Flowchart</i> sistem kontrol ketinggian air sawah	57
Gambar 4. 4 Rangkaian Alat / Komponen	60
Gambar 5. 1 <i>Prototype</i> alat sistem kontrol ketinggian air pada sawah.....	65
Gambar 5. 2 <i>Prototype</i> alat sistem kontrol ketinggian air pada sawah.....	66

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Kesediaan Membimbing TA Pembimbing 1	A-1
Lampiran 2 Surat Kesediaan Membimbing TA Pembimbing 2	A-2
Lampiran 3 Surat Keterangan Observasi	A-3
Lampiran 4 Foto Kegiatan Observasi	B-1
Lampiran 5 Foto Kegiatan Observasi	B-2
Lampiran 6 Foto Perakitan	C-1
Lampiran 7 Foto Perakitan	C-2
Lampiran 8 Source Code	D-1

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perubahan iklim mempunyai dampak yang cukup besar bagi Indonesia. Banyak peristiwa yang sudah terjadi di Indonesia sebagai akibat dari perubahan iklim dan pemanasan global seperti: perubahan pola dan distribusi curah hujan. meningkatnya kejadian kekeringan, banjir dan tanah longsor. menurunnya produksi pertanian/ gagal panen, meningkatnya kejadian kebakaran hutan, meningkatnya suhu di daerah perkotaan, naiknya permukaan air laut. Bencana sebagai akibat perubahan cuaca atau bencana *hidrometeorologi* di Indonesia semakin meningkat. Banjir, longsor, cuaca ekstrim dan kekeringan menjadi lebih sering terjadi [1].

Air adalah salah satu kebutuhan dari tanaman yang harus dipenuhi dalam suatu proses budidaya tanaman. Untuk memenuhi kebutuhan tersebut maka perlu dilakukannya pengairan. Pengairan atau irigasi adalah usaha pemberian air dan pengaturan air untuk menunjang pertanian yang jenisnya meliputi irigasi air permukaan, irigasi air bawah tanah, irigasi pompa, dan irigasi tambak. Namun secara umum metode pemberian air irigasi dapat dibagi menjadi 4 bagian, yakni: (a) Irigasi Permukaan, (b) Irigasi Bawah-permukaan, (c) Irigasi Curah (*sprinkler irrigation*), dan (d) Irigasi Tetes (*drip irrigation*). Metode irigasi yang akan digunakan tergantung pada faktor ketersediaan air, tipe tanah, topografi lahan dan jenis tanaman [2].

Jaringan irigasi sebagai media untuk memenuhi kebutuhan air pertanian perlu dikelola secara efektif dan efisien, satu cara pengelolaan air bawah tanah dan dari sungai tersebut dimanfaatkan secara optimal perlu sistem yang tepat dalam penerapannya diantaranya air bawah tanah dengan pompa yang didistribusikan ke area persawahan sesuai kapasitas airnya, untuk aliran air dari sungai perlu diterapkan dengan membuat saluran terbuka baik lahan kering atau basah dibuatkan tampungan dengan sistem gravitasi yang mampu mengalir lahan persawahan dengan teknik perhitungan debit dan kebutuhan air pada tanaman sesuai musim tanam dan jenis tanaman yang dikembangkan secara optimum, dengan penerapan jaringan irigasi yang mempertimbangkan aspek ketersediaan air, penerapan sistem irigasi secara berkelanjutan (*sustainable*) untuk lebih efisien [3].

Embung atau cekungan penampung (*retention basin*) adalah cekungan yang digunakan untuk mengatur dan menampung suplai aliran air hujan serta untuk meningkatkan kualitas air di badan air yang terkait (sungai, danau). Embung digunakan untuk menjaga kualitas air tanah, mencegah banjir, estetika, hingga pengairan. Embung menampung air hujan di musim hujan dan lalu digunakan petani untuk mengairi lahan di musim kemarau. Selain menampung air hujan, embung juga menampung air yang berasal dari aliran irigasi yang air nya bersumber dari waduk, air di embung ini akan digunakan ketikan irigasi yang biasa teraliri air dari waduk tidak teraliri karena faktor tertentu.

Bawang merah merupakan salah satu komoditas sayuran unggulan Indonesia yang memiliki banyak manfaat dan bernilai ekonomis tinggi dan telah lama diusahakan oleh petani secara intensif serta ditetapkan sebagai salah satu komoditi dalam kelompok produk pertanian penting pengendali inflasi selain cabai dan bawang putih dalam Rencana Strategis Kementerian Pertanian Tahun 2015 – 2019 [4].

Produksi bawang merah di pengaruhi oleh perubahan iklim terutama kekeringan. Kekeringan berhubungan dengan ketersediaan air yang merupakan faktor penting dalam pertumbuhan tanaman. Kurangnya ketersediaan air memberikan cekaman/stress kekeringan pada bawang merah yang dapat menghambat pertumbuhannya. Pengaruh tersebut bervariasi sesuai *kultivar*, besar, dan lama cekaman kekeringan, Ketersediaan air merupakan syarat penting untuk mendapatkan hasil dan kualitas umbi yang optimal [5]

Pertumbuhan tanaman yang terpengaruh kondisi iklim yang tidak menentu seperti terjadi peristiwa iklim ekstrem dapat menyebabkan kegagalan produksi tani dan berkembangnya hama penyakit yang menjadi masalah pertanian. Salah satu tanaman pertanian yang bergantung terhadap kondisi iklim yaitu bawang merah. Tanaman bawang merah sangat rentan terhadap curah hujan yang tinggi yang mengakibatkan bawang merah dapat terendam air ataupun busuk sehingga kualitas bawang merah menjadi menurun [6].

Rendahnya produktivitas bawang merah tergantung dari faktor lingkungan, beberapa faktor penyebab rendahnya produktivitas antara lain adanya tingkat kesuburan tanah yang rendah, adanya peningkatan serangan organisme pengganggu tanaman, adanya perubahan iklim mikro serta bibit yang digunakan bermutu rendah. Salah satu upaya untuk meningkatkan hasil bawang merah adalah dengan menggunakan media tanam yang tepat, yaitu media tanam yang mempunyai sifat fisik tanah yang ringan, gembur dan subur serta memiliki kandungan bahan organik yang tinggi.[7]

Berdasarkan permasalahan tersebut penulis merasa perlunya merancang sistem kontrol ketinggian air otomatis yang dapat memaksimalkan ketersediaan air pada musim kemarau dan juga dapat meminimalisir terjadinya kekeringan pada lahan pertanian yang tentunya dapat mengakibatkan penurunan produksi pada komoditas pertanian. Dan dengan dibangunnya sistem tersebut diharapkan dapat memaksimalkan hasil produksi pertanian dan kesejahteraan petani dapat meningkat.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, diperoleh rumusan masalah yaitu, bagaimana merancang dan membuat Prototype Sistem kontrol ketinggian air pada sawah bang merah berbasis *IOT*.

1.3 Batasan Masalah

Dalam pembuatan tugas akhir ini penulis membatasi masalah, agar tidak meluasnya pembahasan-pembahasan yang timbul. Adapun batasan masalah dalam pembuatan tugas akhir ini adalah:

1. rancangan Sistem kontrol air otomatis.
2. sistem di buat dalam bentuk *Prototype*.
3. sistem ini menggunakan *microcontroller Node MCU*
4. data Base Menggunakan MySQL.
5. sensor ultrasonic HC-SR04 digunakan untuk memonitoring kondisi kebutuhan air di lahan pertanian.
6. sistem monitoring dan kontrol menggunakan Website.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari penelitian ini adalah membuat *prototype* sistem kontrol air otomatis sekaligus juga Sistem monitoring ketersediaan air di lahan pertanian khusus nya tanaman bawang merah, sehingga dengan *prototype* ini dapat memberikan gambaran tentang manfaat, kemudahan dan juga efisiensi ketikan menerapkan teknologi pada sektor pertanian.

1.4.2 Manfaat

1. Bagi Masyarakat

Memberikan kemudahan bagi masyarakat petani dalam

melakukan pengairan dan juga mengatasi kekeringan pada lahan pertaniannya.

2. Bagi Politeknik Harapan Bersama
 - a. Sebagai tolak ukur kemampuan dari mahasiswa dalam menyusun proposal.
 - b. Memberikan kesempatan pada mahasiswa untuk terjun dan berkomunikasi langsung dengan masyarakat
3. Bagi Mahasiswa
 - a. Menambah wawasan mahasiswa tentang bagaimana cara kerja mikrokontroller.
 - b. Memberi bekal untuk menyiapkan diri dalam dunia kerja.
 - c. Menggunakan hasil atau data-data untuk dikembangkan menjadi Tugas Akhir.

1.5 Sistematika Penulisan Laporan

Laporan Tugas Akhir ini terdiri dari enam bab, yang masing-masing bab dengan perincian sebagai berikut:

BAB I PENDAHULUAN

Dalam bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Dalam bab ini menjelaskan tentang penelitian terkait yang diambil dari abstrak jurnal yang didapatkan dan juga menjelaskan landasan teori tentang kajian yang di teliti.

BAB III METODOLOGI PENELITIAN

Bab ini membahas tentang langkah-langkah/ tahapan perencanaan dengan bantuan beberapa metode, teknik, alat (*tools*) yang digunakan seperti Prosedur Penelitian, metode pengumpulan data serta tempat dan waktu pelaksanaan penelitian.

BAB IV ANALISA DAN PERANCANGAN SISTEM

Bab ini menguraikan analisis semua permasalahan yang ada, dimana masalah-masalah yang muncul akan diselesaikan melalui penelitian. Pada bab ini juga dilaporkan secara detail rancangan terhadap penelitian yang dilakukan. Perancangan sistem meliputi Analisis Permasalahan, kebutuhan *hardware* dan *software*, perancangan (diagram blok, *flowchart*), perancangan *database* dan tabel.

BAB V HASIL DAN PEMBAHASAN

Bab ini berisi tentang uraian rinci hasil yang didapatkan dari penelitian yang dilakukan. Pada bab ini juga berisi analisis tentang bagaimana hasil penelitian dapat menjawab pertanyaan pada latar belakang masalah.

BAB VI KESIMPULAN DAN SARAN

Bab ini menguraikan kesimpulan seluruh isi laporan Tugas Akhir dan saran-saran untuk mengembangkan hasil penelitian ini.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Hendri Nurdin dkk (2017) dalam jurnal penelitiannya yang berjudul Optimalisasi Pemanfaatan Mesin Pompa Untuk Mensuplai Kebutuhan Air Sawah Tadah Hujan Di Nagasai Sumani Penerapan teknologi dalam memanfaatkan potensi sungai untuk menyelesaikan permasalahan masyarakat sehingga dapat menikmati suplai air untuk kebutuhan air persawahan melalui mesin pompa sebagai wujud dari Ipteks sosial dan budaya dalam meningkatkan kesejahteraan para petani yang merupakan program pemerintah [7].

Penelitian yang dilakukan Sumardi Sadi dan Ilham Syah Putra (2018) dalam jurnal penelitiannya yang berjudul Rancang bangun Monitoring Ketinggian Air Dan Sistem Kontrol Pada Pintu Air Berbasis Arduino Dan Sms Gateway mengatakan bahwa Ketinggian permukaan air pada sungai adalah salah satu parameter yang perlu diukur untuk mendeteksi banjir secara dini. Penelitian ini bertujuan untuk mengembangkan *prototype system* peringatan dini banjir dengan menggunakan *sensor ultrasonic* yang diintegrasikan dengan arduino uno untuk mengukur ketinggian air [8].

Penelitian yang dilakukan Subianto dkk (2019) dalam jurnal yang berjudul Rancang Bangun System Monitoring Level Air Bendungan Untuk Pengendalian Banjir menyatakan bahwa Banjir terjadi dikarenakan

keterlambatan petugas membuka pintu air bendungan. Minimnya petugas pengawas bendungan saat ini mengakibatkan pemantauan bendungan tidak dapat dilakukan dengan maksimal, hal ini menyebabkan terjadinya banjir karena keterlambatan pengaturan pintu air. Pada penelitian ini penulis melakukan perancangan suatu sistem otomasi monitoring level air bendungan untuk pengendalian banjir. Perangkat yang digunakan adalah Raspberry Pi sebagai perangkat yang menerima data ketinggian level air yang diambil oleh *sensor ultrasonik* HC-SR04 [9].

2.2 Landasan Teori

2.2.1 Irigasi

Pengairan atau irigasi adalah usaha pemberian air dan pengaturan air untuk menunjang pertanian, yang jenisnya meliputi irigasi air permukaan, irigasi air bawah tanah, irigasi pompa, dan irigasi tambak. Namun secara umum metode pemberian air irigasi dapat dibagi menjadi 4 bagian, yakni: (a) Irigasi Permukaan, (b) Irigasi Bawah-permukaan, (c) Irigasi Curah (*sprinkler irrigation*), dan (d) Irigasi Tetes (*drip irrigation*). Metode irigasi yang akan digunakan tergantung pada faktor ketersediaan air, tipe tanah, *topografi* lahan dan jenis tanaman. [3]

Dalam pemberian irigasi pada tanaman perlu juga memperhatikan kebutuhan air tanaman tersebut, untuk itu diperlukan pengontrolan pada pemberian air irigasi untuk mencegah terjadinya kekurangan dan kelebihan pemberian air yang dilakukan. [3]

2.2.2 Monitoring

Monitoring adalah proses pengumpulan dan analisis informasi berdasarkan indikator yang ditetapkan secara sistematis dan kontinu tentang suatu kegiatan atau program sehingga mampu dilaksanakan tindakan koreksi untuk penyempurnaan kegiatan itu selanjutnya. [8]

Monitoring akan memberikan informasi tentang status dan kecenderungan bahwa pengukuran dan evaluasi yang diselesaikan berulang dari waktu ke waktu, 8 pemantauan umumnya dilakukan untuk tujuan tertentu, untuk memeriksa terhadap proses berikut objek atau untuk mengevaluasi kondisi maupun kemajuan menuju tujuan hasil manajemen atas efek tindakan dari beberapa jenis antara lain tindakan untuk mempertahankan manajemen yang sedang berjalan. [8]

Umumnya, *output monitoring* berupa *progress report* proses. *Output* tersebut diukur secara deskriptif maupun non-deskriptif, *output monitoring* bertujuan untuk mengetahui kesesuaian proses telah berjalan. *Output monitoring* berguna pada perbaikan mekanisme proses kegiatan dimana *monitoring* dilakukan. [8]

2.2.3 IOT

Menurut Wikipedia, *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. Makna serupa yang lain, *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 atau manusia ke komputer. "*A Things*" pada *Internet of Things* dapat didefinisikan sebagai subjek misalkan orang dengan *monitor implant* jantung, hewan peternakan dengan *transponder biochip*, sebuah mobil yang telah dilengkapi *built-in* sensor untuk memperingatkan pengemudi ketika tekanan ban rendah.

Sejauh ini, *IoT* paling erat hubungannya dengan komunikasi *machine-to-machine* (M2M) di bidang manufaktur dan listrik, perminyakan, dan gas. Produk dibangun dengan kemampuan komunikasi M2M yang sering disebut dengan sistem cerdas atau "*smart*". (contoh: *smart label*, *smart meter*, *smart grid sensor*). Meskipun konsep ini kurang populer hingga tahun 1999, namun *IoT* telah dikembangkan selama beberapa dekade. [10]

Alat Internet pertama, misalnya, adalah mesin *Coke* di *Carnegie Mellon University* di awal 1980-an. Para programmer dapat terhubung ke mesin melalui Internet, memeriksa status mesin dan

menentukan apakah ada atau tidak minuman dingin yang menunggu, tanpa harus pergi ke mesin tersebut. Istilah *IoT (Internet of Things)* mulai dikenal tahun 1999 yang saat itu disebutkan pertama kalinya dalam sebuah presentasi oleh Kevin Ashton, *cofounder and executive director of the Auto-ID Center* di MIT. [10]

Dengan semakin berkembangnya infrastruktur internet, maka manusia menuju babak berikutnya, di mana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan internet. Namun berbagai macam benda nyata akan terkoneksi dengan internet. Sebagai contohnya dapat berupa: mesin produksi, mobil, peralatan elektronik, peralatan yang dapat dikenakan manusia (*wearables*), dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal dan global menggunakan sensor dan atau aktuator yang tertanam. [10]

2.2.4 NodeMCU 8266

Node MCU adalah sebuah *board* elektronik yang berbasis *chip ESP8266* dengan kemampuan menjalankan fungsi mikrokontroler dan juga koneksi *internet (WiFi)*. Terdapat beberapa pin *I/O* sehingga dapat dikembangkan menjadi sebuah aplikasi monitoring maupun *controlling* pada proyek *IoT*. *NodeMCU ESP8266* dapat diprogram dengan compiler-nya Arduino, menggunakan Arduino IDE. Bentuk fisik dari *NodeMCU ESP 8266*, terdapat *port USB*

(*mini USB*) sehingga akan memudahkan dalam pemrogramannya.

[11]

NodeMCU ESP8266 merupakan modul turunan pengembangan dari modul *platform IoT (Internet of Things)* keluarga *ESP8266* tipe ESP-12. Secara fungsi modul ini hampir menyerupai dengan *platform* modul arduino, tetapi yang membedakan yaitu dikhususkan untuk “*Connected to Internet*“. [11]



Gambar 2. 1 NodeMCU 8266.

Keterangan :

Tabel 2. 1 NodeMCU 8266

No.	Nama Bagian Node MCU	Keterangan
1	Tegangan output	3,3 – 5V
2	GPIO	13 Pin
3	Kanal PWM	10 Kanal
4	10 bit ADC pin	1 Pin
5	Memory	4 MB
6	Kecepatan	40/26/24 MHz
7	WiFi	IEEE 802.11 b/g/n
8	Frekuensi	2,4 GHz – 22,5 GHz
9	USB Port	Micro USB
10	USB ke serial converter	CH340G



Gambar 2. 2 Skematik Posisi Pin Node MCU

2.2.5 Laravel

Pengertian *framework* menurut Naista adalah suatu struktur konseptual dasar yang digunakan untuk memecahkan atau menangani suatu masalah yang kompleks. Singkatnya, *framework* adalah wadah atau kerangka kerja dari sebuah website yang akan dibangun. Dengan menggunakan kerangka tersebut waktu yang digunakan dalam membuat *website* lebih singkat dan memudahkan dalam melakukan perbaikan. Salah satu *framework* yang banyak digunakan oleh *programmer* adalah *framework laravel*. Laravel adalah *framework* berbasis PHP yang sifatnya *open source*, dan menggunakan konsep *model-view-controller*. Laravel berada di bawah lisensi MIT *License* dengan menggunakan Github sebagai tempat berbagi *code* menjalankannya. [12]



Gambar 2. 3 Laravel

Dalam penggunaannya laravel memiliki beberapa kekurangan salah satunya yaitu ukuran *file* yang cukup besar. Di dalam laravel terdapat *file* yang sifatnya *default* seperti vendor. *File* tersebut tidak boleh dihapus sembarangan sehingga ukuran *website* yang dibuat berukuran cukup besar. Selain itu, dibutuhkan koneksi internet untuk instalasi dan mengunduh *library* laravel, dan PHP minimal versi 5.4 untuk menjalankannya. [12]

Berikut adalah dasar-dasar laravel

1. Artisan

Artisan adalah *command line* atau perintah yang dijalankan melalui terminal dan disediakan beberapa perintah perintah yang dapat digunakan selama melakukan pengembangan dan pembuatan aplikasi. Salah satu fungsi dari php artisan yaitu “*php artisan serve*”. *Php artisan serve* berfungsi untuk membuka *website* yang telah dibuat tanpa menggunakan *web server* lokal.

2. Routing

Routing adalah suatu proses yang bertujuan agar suatu item yang diinginkan dapat sampai ke tujuan. Dengan menggunakan

routing dapat ditentukan halaman halaman yang akan muncul ketika dibuka oleh *user*. Pengaturan *routing* di laravel biasanya terletak di *file* web.php. *File* web.php terletak di dalam *folder* *routes*.

3. Controller

Controller adalah suatu proses yang bertujuan untuk mengambil permintaan, menginisialisasi, memanggil model untuk dikirimkan ke *view*. Ada dua cara membuat *controller* di laravel. Cara pertama adalah dibuat *file* controller secara manual dan dituliskan *code extends controller* di dalamnya. Cara kedua adalah dibuat *file controller* menggunakan command line dengan menuliskan “*php artisan make controller nama_file_controller*”. Permintaan yang dibuat dalam laravel harus berada di dalam controller, kemudian dilempar melalui *routing* untuk mendapat permintaan yang diinginkan.

4. View (*blade templating*)

Blade adalah *template engine* bawaan dari laravel. *Blade* memiliki kode kode yang lebih mudah untuk menghasilkan laravel. Cara membuat *file.blade* dilakukan secara manual dengan membuat nama *file.php.blade* di dalam *folder views*. Di dalam blade dapat dibuat *template* master dan *template inheritance*. Pembuatan *template* master dan turunannya ini bertujuan agar elemen yang sama tidak ditulis secara berulang-

ulang. Pada template *inheritance* diberikan kode “*extend* (nama_Layout) dan section (nama_content)”.

5. Middleware

Middleware adalah penengah Antara *request* yang masuk dengan *controller* yang dituju. Cara membuat *middleware* menggunakan artisan dengan mengetikkan “*php artisan make:middleware nama_file*”. File *middleware* berada di dalam folder *middleware*.

6. Session

Session adalah sebuah cara yang digunakan untuk penyimpanan pada *server* dan penyimpanan tersebut digunakan pada beberapa halaman termasuk halaman itu sendiri. Dalam menggunakan *session* ada dua cara. Cara yang pertama *session* dapat dibuat menggunakan *Request*. Cara yang kedua dapat digunakan fungsi *global helper session*.

2.2.6 Arduino IDE

Arduino *IDE* dibuat dari bahasa pemrograman JAVA. Arduino *IDE* juga dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino *IDE* ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino *IDE* khusus untuk pemrograman dengan Arduino. [13]

Arduino memiliki *open-source* yang memudahkan untuk menulis kode dan mengupload *board* ke arduino. Arduino *IDE* (*Integrated Development Enviroment*) ini merupakan media yang digunakan untuk memberikan informasi kepada arduino sehingga dapat memberikan *output* sesuai dengan apa yang diinginkan. *Software* arduino yaitu berupa *software processing* yang digunakan untuk menulis program kedalam Arduino Uno, merupakan penggabungan antara bahasa C++ dan Java. *Software* Arduino dapat diinstall di berbagai *operating* sistem seperti Linux, Mac OS, Windows. *IDE* (*Integrated Development Enviroment*) arduino merupakan pemograman dengan menggunakan bahasa C. Setiap program *IDE* arduino yang biasa disebut *sketch Interface* Arduino IDE. [13]



Gambar 2. 4 Arduino IDE

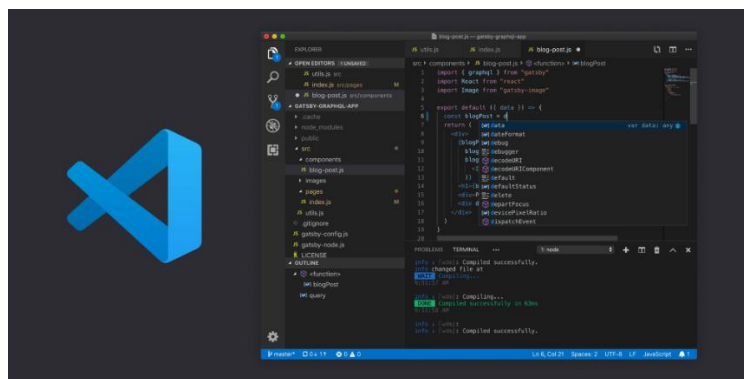
2.2.7 Visual Studio Code

Visual studio code adalah *Software* yang sangat ringan, namun kuat editor kode sumbernya yang berjalan dari desktop. Muncul dengan *built-in* dukungan untuk *javascript*, *python* dan *Node.js* dan

memiliki *array* dengan beragam ekstensi yang tersedia untuk Bahasa lain, termasuk *C++*, *C#*, *Python*, dan *PHP*. Hal ini didasarkan sekitar Github dan Elektron, yang merupakan versi *cross-platform* dari Atom komponen kode editing, berdasarkan *JavaScript* dan *HTML5*. Editor ini adalah fitur lengkap lingkungan pengembangan terpadu (*IDE*) dirancang untuk pengembangan yang bekerja dengan teknologi *cloud* yang terbuka *Microsoft*. Visual Studio Code menggunakan *open source* NET perkakas untuk memberikan dukungan untuk *ASP.NET C#* kode, membangun pengembangan *omnisharp NET* dan *Compiler Roslyn*. Antarmuka yang mudah untuk bekerja dengan, karena didasarkan pada gaya *explorer* umum, dengan panel di sebelah kiri, yang menunjukkan semua file dan *folder* yang memiliki akses ke panel editor di sebelah kanan, yang menunjukkan isi dari *file* yang telah di buka. Dalam hal ini, *editor* telah dikembangkan dengan baik, dan menyenangkan pada mata. Ia juga memiliki fungsi yang baik, dengan *intellisense* dan *autocomplete* bekerja dengan baik untuk *JSON*, *CSS*, *HTML*, dan *Node.js*. [14]

Visual Studio Code telah dirancang untuk bekerja dengan alat-alat yang ada, dan Microsoft menyediakan dokumentasi untuk membantu pengembangan bersama, dengan bantuan untuk bekerja dengan *ASP.NET 5*, *Node.js* dan *Microsoft* naskah, serta alat-alat yang dapat digunakan untuk membantu membangun dan mengelola

aplikasi Node.js. visual studio code benar-benar sedang ditargetkan pada pengembangan *JavaScript* yang ini alat pengembangannya lengkap untuk *Scripting server-sidenya* dan yang mungkin ingin usaha dan Node.js untuk kerangka berbasis NET. Visual studio code, adalah belum slid, lintas platform kode editor ringan, yang dapat digunakan siapa saja untuk membangun aplikasi *WEB*. [14]



Gambar 2. 5 Tampilan Visual Studio Code

2.2.8 Database

Basis data (*database*) adalah data yang saling terkumpul dan terorganisi yang berhubungan satu sama lain dimana dapat menghasilkan kegiatan mendapatkan informasi lebih mudah. Tujuan dari basis data ialah agar masa didalam sistem yang menggunakan penghampiran berdasar *file* dapat diatasi. Basis data terdiri dari dua kata, yaitu basis dan data. Basis bermakna sebagai gudang Sedangkan data ialah repretasi. [15]

Bukti dunia nyata yang mewakili suatu objek seperti manusia, hewan barang, konsep, peristiwa dan sebagainya. Kemudian data

tadi direkam dalam bentuk angka, huruf, teks, gambar, simbol, bunyi, ataupun kombinasinya. [15]

2.2.9 HTML

HTML (HyperText Markup Language) adalah suatu cara memberikan tanda yang memberikan perintah kepada *browser* bagaimana suatu teks terstruktur. *HMTL* memberikan perintah kepada browser bagaimana struktur dari dokumen, bagaimana *heading*-nya, bagaimana paragrafnya, bagaimana suatu teks akan ditampilkan, dan lainnya. Dengan informasi yang diberikan, *browsers* dibangun dengan perintah dasar bagaimana menampilkan setiap elemen yang ada. [16]

2.2.10 CSS

CSS (Cascading Style Sheet) adalah bahasa pemrogramman yang digunakan untuk mendeskripsikan bagaimana suatu konten akan ditampilkan. *CSS* memberikan karakteristik tampilan dari elemen yang ada pada *HTML* menggunakan *CSS*. (Elisabeth & Eric 2012) Kode *CSS* dapat dituliskan dengan tiga cara yaitu *inline*, *internal* dan *external*. Ketiganya bisa lakukan sesuai dengan kebutuhan. [17]

2.2.11 PHP

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal *PHP* adalah ketika itu Radmus ingin mengetahui jumlah pengunjung yang membaca

resume online-nya. *Script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi pengunjung, dan menampilkan jumlah pengunjung dari suatu *website*. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia *web* saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan *script* buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page* (PHP). *PHP* dibangun dari *scripts* yang ditulis secara *plaintext*. *PHP Interpreter* adalah bagian dari perangkat lunak yang ada pada *Web Server*, yang membaca *File* tersebut dan mengartikannya, memberikan keluaran *HTML* dan petunjuk mengenai bagaimana perilaku yang ada maupun menginterpretasikan masukan dari pengguna. [18]

2.2.12 Java Script

Javascript menjadikan suatu web lebih interaktif, ini menjadikannya bisa mengerti apa yang dibutuhkan, memroses masukan pengguna, serta memberikan respon yang lebih baik. Javascript digunakan bersama dengan *HTML* dan *CSS* sebagai salah satu dari tiga *Modern Web Page Construction*. *HTML* menyediakan struktur, *CSS* memberikan tampilan, dan *JavaScripts* menjalankannya dan membuat perintah berjalan. Javascript dikembangkan oleh Brendan Eich di Netspace pada tahun 1995-1996. Dulu disebut dengan *LiveScript*. Javascript bukanlah bahasa

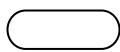
yang di *compile*, ini membuatnya terlihat serta terasa memiliki banyak kelebihan. Javascript pada awalnya terfokus pada sisi pengguna sebagai *Form-Validation*, dan digunakan dengan gambar untuk menyempurnakannya, lebih membantu, membuatnya lebih interaktif serta memberikan umpan balik bagi pengunjung. [19]

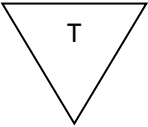
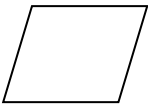

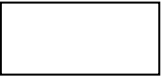

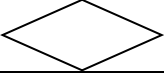

2.2.13 Flow Chart

Flow chart adalah bagan yang menggambarkan aliran dokumen dalam suatu sistem informasi.” Menurut Al-Bahra bin ladjamudin mengatakan bahwa: “*flow chart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah. *Flow chart* merupakan cara penyajian dari suatu algoritma.” [20]

Dari dua definisi diatas maka dapat disimpulkan bahwa pengertian *flow chart* adalah suatu simbol yang digunakan untuk menggambarkan suatu arus data yang berhubungan dengan suatu sistem transaksi akuntansi. Simbol dari bagan alir (*flow chart*) adalah sebagai berikut ini :

Tabel 2. 2 *Flowchart Program*

No	Simbol	Pengertian	Keterangan
1.		Mulai / berakhir (<i>Terminal</i>)	Digunakan untuk memulai, mengakhiri, atau titik henti dalam sebuah proses atau program; juga digunakan untuk menunjukkan pihak eksternal.

No	Simbol	Pengertian	Keterangan
2.		Arsip	Arsip dokumen disimpan dan diambil secara manual. Huruf didalamnya menunjukkan cara pengurutan arsip: N = Urut Nomor; A = Urut Abjad; T = Urut Tanggal.
3.		Input / Output; Jurnal / Buku Besar	Digunakan untuk menggambarkan berbagai media input dan output dalam sebuah bagan alir program.
4.		Penghubung Pada Halaman Berbeda	Menghubungkan bagan alir yang berada di halaman yang berbeda.
5.		Pemrosesan Komputer	Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi
6.		Arus Dokumen atau Pemrosesan	Arus dokumen atau pemrosesan; arus normal adalah ke kanan atau ke bawah.
7.		Keputusan	Sebuah tahap pembuatan keputusan
8.		Penghubung Dalam Sebuah Halaman	Menghubungkan bagan alir yang berada pada halaman yang sama.

2.2.14 UML

Unified Modeling Language (UML) adalah bahasa standar untuk menulis denah perangkat lunak. *UML* dapat digunakan untuk memvisualisasikan, menentukan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak. Dengan kata lain, seperti arsitek bangunan membuat denah yang akan

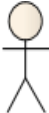
digunakan oleh sebuah perusahaan konstruksi, arsitek *software* membuat diagram *UML* untuk membantu pengembang perangkat lunak membangun perangkat lunak. Jika memahami kosakata *UML*, maka dapat lebih mudah memahami dan menentukan sistem dan menjelaskan desain sistem kepada orang lain. [21]



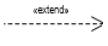

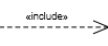
Unified Modeling Language merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. *UML* merupakan sebuah standar penulisan atau semacam *blue print* dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Terdapat beberapa diagram *UML* yang sering digunakan dalam pengembangan sebuah sistem, yaitu :

1. *use Case* : merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat actor yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem.

Berikut adalah simbol-simbol *use case* :






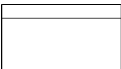
Tabel 2. 3 Simbol-simbol *use case*

No	Nama	Simbol	Keterangan
1	Aktor		Orang proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan

No	Nama	Simbol	Keterangan
			dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
2	Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case.
3	Asosiasi / <i>Association</i>		Komunikasi antara <i>actor</i> dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan <i>actor</i> .
4	Ekxtensi / <i>Extend</i>		Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki nama depan yang sama dengan use case yang di tambahkan.
5	Generalisasi / <i>Generalization</i>		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<i>Include</i>		Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini.


2. *activity Diagram* : merupakan gambaran alir dari aktivitas-aktivitas didalam sistem yang berjalan.

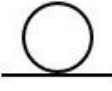
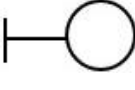


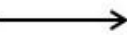
Tabel 2. 4 simbol-simbol *activity Diagram*

No	Nama	Simbol	Keterangan
1	Status Awal / <i>Initial</i>		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas / <i>Activity</i>		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>Decision</i>		Asosiasi percabangan dimana lebih dari satu aktivitas digabungkan menjadi satu.
4	Penggabungan / <i>Join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas lebih dari satu.
5	Status Akhir / <i>Final</i>		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status satu.
6	<i>Swumline</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

3. *sequence Diagram* : menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa message yang digambarkan terhadap waktu.



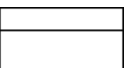
Tabel 2. 5 simbol-simbol *sequence Diagram*


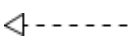
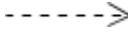

No	Nama	Simbol	Keterangan
1	<i>Actor</i>		Menggambarkan orang yang sedang berinteraksi dengan sistem

No	Nama	Simbol	Keterangan
2	<i>Entity Class</i>		Menggambarkan hubungan yang akan dilakukan
3	<i>Boundary Class</i>		Menggambarkan sebuah gambaran dari <i>form</i>
4	<i>Control Class</i>		Menggambarkan penghubung antara boundary dengan tabel.
5	<i>A focus of control & A life line</i>		Menggambarkan tempat mulai dan berakhirnya <i>message</i>
6	<i>A message</i>		Menggambarkan pengiriman pesan

4. *class diagram* : merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling berhubungan seperti diantaranya pewarisan, asosiasi dan lainnya.


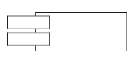
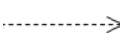
Tabel 2. 6 simbol-simbol *class diagram*



No	Nama	Simbol	Keterangan
1	<i>Generalization</i>		Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2	<i>Nary Association</i>		Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3	<i>Class</i>		Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.

No	Nama	Simbol	Keterangan
4	<i>Collaboration</i>		Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5	<i>Realization</i>		Operasi yang benar-benar dilakukan oleh suatu objek.
6	<i>Dependency</i>		Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7	<i>Association</i>		Apa yang menghubungkan antara objek satu dengan objek lainnya

5. *component diagram* : diagram yang menunjukkan secara fisik komponen perangkat lunak pada sistem dan hubungannya antar *component*. *Component Diagram* merupakan bagian dari sistem yang diuraikan menjadi subsistem atau modul yang lebih kecil.

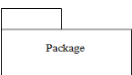

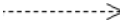

Tabel 2. 7 simbol-simbol *component diagram*

No	Nama	Simbol	Keterangan
1	<i>Package</i>		merupakan sebuah bungkus dari satu atau lebih komponen.
2	<i>Komponen / component</i>		Komponen merupakan komponen sistem.
3	<i>Dependency</i>		<i>dependency</i> atau kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.

No	Nama	Simbol	Keterangan
4	<i>Interface</i>		<i>interface</i> merupakan antarmuka sama dengan interface pada pemrograman berorientasi objek, yaitu sebagai antar muka komponen agar tidak mengakses langsung komponen.
5	<i>Link</i>		Menggambarkan relasi antar komponen.

6. *deployment diagram* : mendeskripsikan arsitektur fisik dalam node untuk perangkat lunak dalam sistem. Komponen perangkat lunak, processor, dan peralatan lain yang membangun arsitektur sistem secara *runtime*.

Tabel 2. 8 simbol-simbol *deployment diagram*

No	Nama	Simbol	Keterangan
1	<i>Package</i>		merupakan sebuah bungkus dari satu atau lebih komponen.
2	<i>Node</i>		<i>Node</i> biasa mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak dibuat sendiri (<i>software</i>), jika di dalam node disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang telah didefinisikan sebelumnya pada diagram komponen.
3	<i>Dependency</i>		<i>dependency</i> atau kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai.
4	<i>Link</i>		Menggambarkan relasi antar komponen.

2.2.15 Sensor Ultrasonic HC-SR04

Sensor *Ultrasonic(HC-SR04)* adalah sensor pengukur jarak berbasis gelombang *Ultrasonic*. Prinsip kerja sensor ini mirip dengan radar *Ultrasonic*. Gelombang *Ultrasonic* dipancarkan kemudian diterima balik oleh *receiverultrasonic*. Jarak antara waktu pancar dan waktu terima adalah representasi dari jarak objek. Sensor ini cocok untuk aplikasi elektronik yang memerlukan deteksi jarak termasuk untuk sensor pada robot. [22]

Sensor HC-SR04 adalah versi *low cost* dari sensor *ultrasonic* PING buatan *parallax*. Perbedaannya terletak pada pin yang digunakan. *HC- SR04* menggunakan 4 pin sedangkan PING buatan *parallax* menggunakan 3 pin. Pada *Sensor HC-SR04* pin *trigger* dan *output* diletakkan terpisah. Sedangkan jika menggunakan PING dari *Parallax* pin *trigger* dan *output* telah diset *default* menjadi satu jalur. Tidak ada perbedaan signifikan dalam pengimplementasiannya. Jangkauan karak sensor lebih jauh dari PING buatan *parallax*, dimana jika ping buatan *parallax* hanya mempunyai jarak jangkauan maksimal 350 cm sedangkan sensor *HC-SR04* mempunyai kisaran jangkauan maksimal 400-500c. [22]

Sensor *Ultrasonic* merupakan sensor yang memanfaatkan bunyi dengan frekuensi *Ultrasonic*. Frekuensi ini tidak dapat didengar oleh manusia karena memiliki frekuensi ini tidak dapat didengar oleh manusia karena memiliki frekuensi di atas 20 Khz, sensor ini sendiri

biasanya menggunakan Rx. Sehingga umumnya sensor ini dipakai untuk pengukur jarak. [22]



Gambar 2. 6 Sensor *Ultrasonic*

Cara Kerja :

1. sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.
2. sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
3. setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus : $S = 340.t/2$. Dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu pemancaran gelombang oleh *transmitter* dan waktu ketika gelombang pantul diterima *receiver*.

Rangkaian Dasar Sensor Ultrasonik ada 3 yaitu :

1. piezoelektrik berfungsi untuk mengubah energi listrik menjadi energi mekanik. Bahan piezo elektrik adalah material yang memproduksi medan listrik ketika dikenai regangan atau tekanan mekanis. Sebaliknya, jika medan listrik diterapkan, maka material tersebut akan mengalami regangan atau tekanan mekanis. Jika rangkaian pengukur beroperasi pada mode pulsa elemen *piezoelektrik* yang sama, maka dapat digunakan sebagai *transmitter* dan *reiceiver*. Frekuensi yang ditimbulkan tergantung pada osilatornya yang disesuaikan frekuensi kerja dari masing-masing *transduser*. Karena kelebihanannya inilah maka *tranduser piezoelektrik* lebih sesuai digunakan untuk sensor ultrasonik.
2. *Transmitter* adalah sebuah alat yang berfungsi sebagai pemancar gelombang ultrasonik dengan frekuensi tertentu (misal, sebesar 40 kHz) yang dibangkitkan dari sebuah osilator. Untuk menghasilkan frekuensi 40 KHz, harus dibuat sebuah rangkaian osilator dan keluaran dari osilator dilanjutkan menuju penguat sinyal. Besarnya frekuensi ditentukan oleh komponen RLC / kristal tergantung dari disain osilator yang digunakan. Penguat sinyal akan memberikan sebuah sinyal listrik yang diumpankan ke piezoelektrik dan terjadi reaksi mekanik sehingga bergetar

dan memancarkan gelombang yang sesuai dengan besar frekuensi pada osilator.

3. *receiver* terdiri dari transduser ultrasonik menggunakan bahan *piezoelektrik*, yang berfungsi sebagai penerima gelombang pantulan yang berasal dari *transmitter* yang dikenakan pada permukaan suatu benda atau gelombang langsung LOS (*Line of Sight*) dari *transmitter*. Oleh karena bahan *piezoelektrik* memiliki reaksi yang *reversible*, elemen keramik akan membangkitkan tegangan listrik pada saat gelombang datang dengan frekuensi yang resonan dan akan menggetarkan bahan *piezoelektrik* tersebut.

2.2.16 Servo

Servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik *loop* tertutup (*Servo*), sehingga dapat *diset-up* atau diatur untuk menentukan dan memastikan posisi sudut dari poros *output* motor. *Motor Servo* merupakan perangkat yang terdiri dari motor *DC*, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros *motor DC* akan memperlambat putaran poros dan meningkatkan torsi *Motor Servo*, sedangkan potensiometer dengan perubahan resistansinya saat *motor* berputar berfungsi sebagai penentu batas posisi putaran poros *Motor Servo*. Penggunaan sistem kontrol *loop* tertutup pada *Motor Servo* berguna untuk mengontrol

gerakan dan posisi akhir dari poros *Motor Servo*. Penjelasan sederhananya begini, posisi poros *output* akan di sensor untuk mengetahui posisi poros sudah tepat seperti yang diinginkan atau belum, dan jika belum, maka kontrol input akan mengirim sinyal kendali untuk membuat posisi poros tersebut tepat pada posisi yang diinginkan. [23]

Untuk lebih jelasnya mengenai sistem kontrol *loop* tertutup, perhatikan contoh sederhana beberapa aplikasi lain dari sistem kontrol *loop* tertutup, seperti penyetelan suhu pada AC, kulkas, setrika dan lain sebagainya.



Gambar 2. 7 Motor Servo

Motor Servo biasa digunakan dalam aplikasi-aplikasi di industri, selain itu juga digunakan dalam berbagai aplikasi lain seperti pada mobil mainan radio kontrol, robot, pesawat, dan lain sebagainya. Motor Servo mempunyai 3 kabel, yaitu kabel *power*, *ground* dan kendali. Terdiri dari *motor* dc, *gear box*, potensio meter dan rangkaian kendali. Tipe *Motor Servo* menentukan kapasitas

motor untuk menanggung beban. Operasional dari *Motor Servo* dikendalikan oleh pulsa selebar kurang lebih 20 ms yang mana lebar pulsa antara 0,5ms dan 2ms menyatakan akhir dari range sudut maksimum. Terdapat dua jenis *Motor Servo* yang dan terdapat di pasaran, yaitu *Motor Servo rotation 180⁰* dan *Servorotation continuous*. [23]

1. *Motor Servo standard* (*Servo rotation 180⁰*) adalah jenis yang paling umum dari *Motor Servo*, dimana putaran poros *outputnya* terbatas hanya 90⁰ kearah kanan dan 90⁰ kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180⁰.
2. *Motor Servo rotation continuous* merupakan jenis *Motor Servo* yang sebenarnya sama dengan jenis *Servo standard*, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

Motor Servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (*Pulse Wide Modulation/PWM*) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros *motor Servo*. Sebagai contoh, lebar pulsa dengan waktu 1,5 ms (mili detik) akan memutar poros *Motor Servo* ke posisi sudut 90⁰ . Bila pulsa lebih pendek dari 1,5 ms maka akan berputar ke arah posisi 0⁰ atau ke kiri (berlawanan dengan arah jarum jam), sedangkan bila pulsa yang diberikan lebih lama dari 1,5ms maka

poros *Motor Servo* akan berputar ke arah posisi 180° atau ke kanan (searah jarum jam). Bentuk pulsa sinyal kontrol *Motor Servo* ketika lebar pulsa kendali telah diberikan, maka poros *Motor Servo* akan bergerak atau berputar ke posisi yang telah diperintahkan, dan berhenti pada posisi tersebut dan akan tetap bertahan pada posisi tersebut. Jika ada kekuatan eksternal yang mencoba memutar atau mengubah posisi tersebut, maka *Motor Servo* akan mencoba menahan atau melawan dengan besarnya kekuatan torsi yang dimilikinya (*rating torsi Servo*). Namun *Motor Servo* tidak akan mempertahankan posisinya untuk selamanya, sinyal lebar pulsa kendali harus diulang setiap 20ms untuk menginstruksikan agar posisi poros *Motor Servo* tetap bertahan pada posisinya. [23]

2.2.17 Pompa Air

Pompa merupakan alat yang berfungsi mengubah tenaga mekanis dari suatu sumber tenaga (penggerak) menjadi tenaga kinetis (kecepatan), dimana tenaga ini berguna untuk mengalirkan cairan dan mengatasi hambatan yang ada sepanjang pengaliran. Dalam aplikasi kehidupan sehari-hari banyak sekali aplikasi yang berkaitan dengan pompa. Contoh pompa yang ditemui dalam kehidupan sehari-hari antara lain pompa air, pompa diesel, pompa hydram, pompa bahan bakar dan lain-lain, yang digunakan oleh masyarakat pada umumnya. Dari sekian banyak pompa yang ada

tentunya mempunyai prinsip kerja dan kegunaan yang berbeda-beda namun memiliki fungsi yang sama. [24]



Gambar 2. 8 Pompa Air

2.2.18 Relay

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electro mechanical* (Elektro mekanikal) yang terdiri dari 2 bagian utama yakni Elektro magnet (Coil) dan Mekanikal (seperangkat Kontak Saklar/*Switch*). *Relay* menggunakan Prinsip Elektro *magnetic* untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. [25]



Gambar 2. 9 Module Relay 2 Chanel

2.2.19 Adaptor / Power Supply

Adaptor yaitu piranti elektronik yang bisa mengubah tegangan listrik (*AC*) yang tinggi jadi tegangan listrik (*DC*) yang rendah,

namun ada juga jenis adaptor yang bisa mengubah tegangan listrik yang rendah jadi tegangan listrik yang tinggi, dan ada beberapa jenis adaptor [26] diantaranya:



Gambar 2. 10 Power Supply

1. Adaptor *DC Converter*.

Adalah adaptor yang bisa mengubah tegangan *DC* yang besar jadi tegangan *DC* yang kecil. Contohnya tegangan 12 Volt *DC* jadi 6 Volt *DC*.

2. Adaptor *Step up sert Step Down*.

Adaptor *step up* yaitu adaptor yang bisa mengubah tegangan *AC* yang kecil jadi tegangan *AC* yang besar. Contohnya tegangan 110V jadi tegangan 220V. Adaptor *step down* yaitu adaptor yang bisa mengubah tegangan *AC* yang besar jadi tegangan *AC* yang kecil. Contohnya tegangan 220 Volt menjadi tegangan 110 Volt.

3. Adaptor *Power Supply*.

Adalah adaptor yang bisa mengubah tegangan listrik *AC* yang besar jadi tegangan *DC* yang kecil. Contohnya tegangan 220 Volt *AC* jadi tegangan 6 Volt, 9 Volt atau 12 Volt *DC*.

2.2.20 Kabel Jumper

Kabel *jumper* adalah kabel elektrik untuk menghubungkan antar komponen di *breadboard* tanpa memerlukan solder. Kabel *jumper* umumnya memiliki *connector* atau pin di masing-masing ujungnya. *Connector* untuk menusuk disebut *male connector*, dan *connector* untuk ditusuk disebut *female connector* kabel *jumper* dibagi menjadi 3 yaitu : *Male to Male*, *Male to Female* dan *Female to Female*. [27]

Kabel yang digunakan sebagai penghubung antar komponen yang digunakan dalam membuat perangkat *prototype*. Kabel *jumper* bisa dihubungkan ke *controller* seperti *raspberry pi*, *arduino* melalui *bread board*. Kabel *jumper* akan ditancapkan pada pin *GPIO* di *raspberry pi*. [27]

Karakteristik dari kabel *jumper* ini memiliki panjang antara 10 sampai 20 cm. Jenis kabel *jumper* ini jenis kabel serabut yang bentuk *housingnya* bulat. Dalam merancang sebuah desain rangkaian elektronik, maka dibutuhkan sebuah kabel yang digunakan untuk menghubungkannya. [27]



Gambar 2. 11 Kabel *Jumper*

BAB III

METODOLOGI PENELITIAN

3.1. Prosedur Penelitian

3.1.1. Rencana / *Planning*

Rencana/ *Planning* yang dilakukan adalah dengan melakukan observasi pada pertanian bawang merah. Melihat dan memahami apa saja yang dibutuhkan tanaman bawang merah agar tetap terjaga dari awal tanam sampai panen. Setelah melihat lalu memahami, maka muncul suatu ide atau gagasan untuk menunjang dan membantu memudahkan petani dalam hal pengairan lahan pertanian bawang merah yaitu dengan membuat alat atau sistem, yang dimana alat atau sistem yang sudah ada dengan mengembangkan cara-cara yang masih manual agar menjadi otomatis dan menghemat waktu petani bawang merah dan meminimalisir resiko kekeringan pada lahan pertanian. Sistem ini dibuat dengan kelebihan otomatisasi, dengan mempertimbangkan tingkat ketetapan dalam pengaliran air, sehingga cara cara manual terdahulu tidak perlu lagi digunakan untuk mengairi ladang tanaman bawang merah. Ketika waktu sudah ditetapkan untuk mengalir air maka alat atau sistem secara langsung akan memberikan informasi bahwa ladang pertanian bawang akan otomatis teraliri air dengan sendirinya untuk memenuhi kebutuhan air pada ladang tanaman bawang tersebut.

3.1.2. Data Analisis

Pada tahap analisis ini akan diuraikan permasalahan yang dihadapi dengan maksud agar dapat mengidentifikasi dan mengevaluasi permasalahan yang dibutuhkan agar lebih efektif.

Maka sistem yang dibangun mampu melakukan beberapa hal berikut :

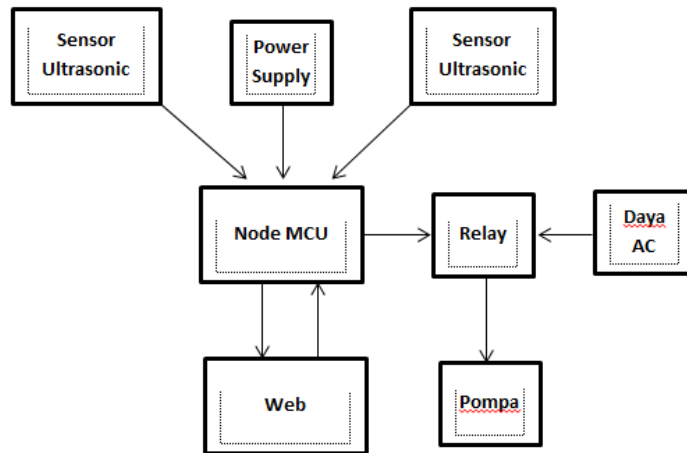
1. membaca ketersediaan air pada sawah.
2. mengirimkan permintaan atau order ketika sawah membutuhkan air.
3. nodemcu akan mengirim data ke *website*.

Dalam pembuatan *Sistem control ketinggian air berbasis IOT*. Berdasarkan data analisa diatas, diperlukan *software Arduino IDE* untuk merancang semua jenis *input- output* terhadap alat yang akan digunakan. *Software IDE arduino* diinstal pada laptop yang mempunyai *processor Intel Core i3, RAM 2,00 GB*.

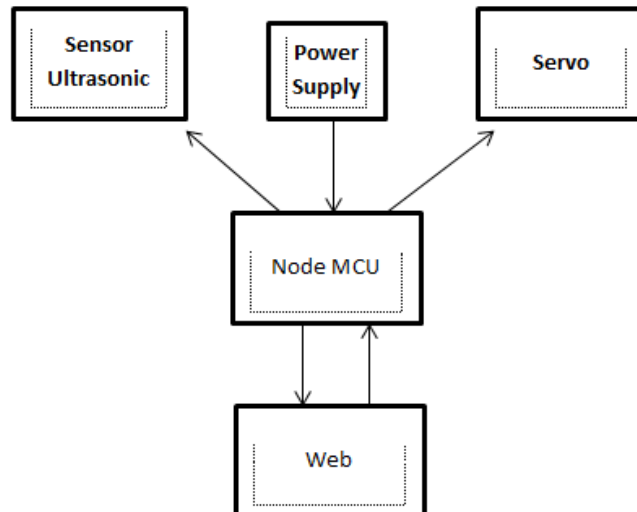
Hasil analisa diatas terdapat permasalahan yang diselesaikan yaitu bagaimana agar *monitoring* dapat dilakukan dari jarak jauh dan proses pengairan dilakukan secara otomatis.

3.1.3. Rancangan dan Desain

Pada perancangan ini akan membahas rangkaian skematik dari setiap komponen serta modul serta koneksi dari setiap *port* modul tersebut. Pembahasan difokuskan pada desain skematik seperti pada *block diagram* alat.



Gambar 3. 1 Blok Diagram Sistem (Pompa Air Otomatis)



Gambar 3. 2 Blok Diagram Sistem (*Monitoring*)

Blok diagram akan dijelaskan masing-masing fungsi sebagai berikut:

1. *power Supply* digunakan sebagai penyuplai arus listrik.
2. *servo* digunakan untuk membuka tutup saluran air / pipa.
3. *sensor Ultrasonic* digunakan untuk mendeteksi ketersediaan air baik di sungai atau di ladang tanaman bawang merah.
4. *relay* digunakan sebagai kontak saklar

5. *pompa* digunakan untuk memindahkan air dari satu tempat ke tempat yang lainnya.
6. *web* digunakan untuk memonitoring ketersediaan air di sungai dan di ladang tanaman bawang merah.

3.1.4. Implementasi

Hasil dari penelitian ini akan diuji cobakan secara *real* dalam bentuk *prototype* / rancang bangun untuk menilai seberapa baik produk Sistem kontrol ketinggian air pada sawah bawang merah berbasis *IOT* menggunakan *Sensor Ultrasonic* dan dengan Sistem Kontrol Melalui *Website* yang telah dibuat serta memperbaiki bila ada kesalahan-kesalahan yang terjadi. Kemudian hasil dari uji coba tersebut akan diimplementasikan.

3.2. Metode Pengumpulan Data

3.2.1 Observasi

Metode pengumpulan data melalui pengamatan yang meliputi lokasi, tanaman bawang merah, dan alat-alat yang digunakan dalam pembuatan Sistem monitoring irigasi otomatis berbasis *IOT*, serta meninjau secara langsung lokasi yang akan diobservasi.



Gambar 3. 3 Ladang Tanaman Bawang Merah



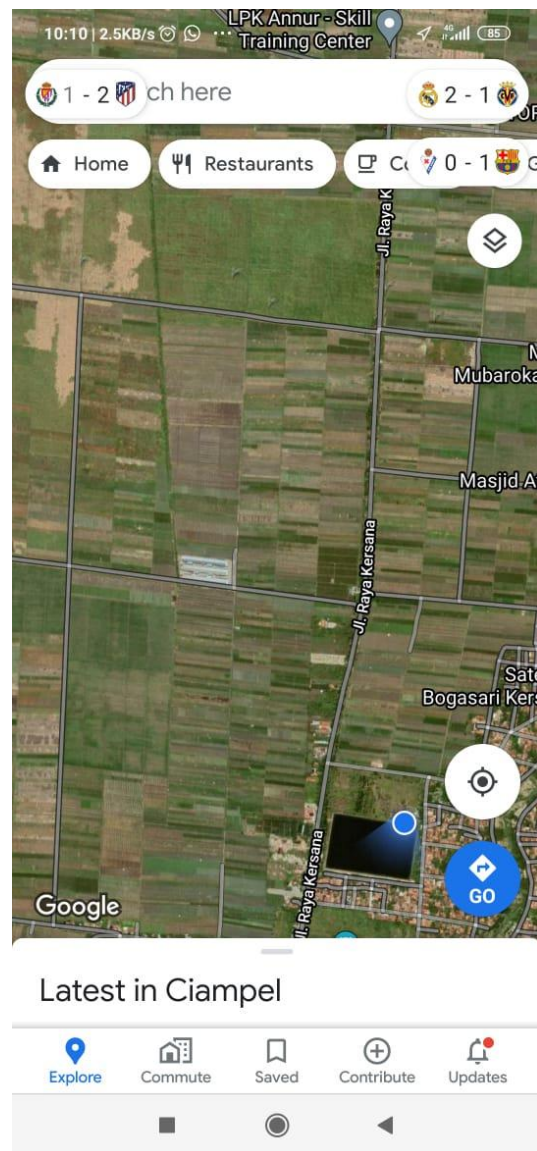
Gambar 3. 4 Jalur keluar masuk air di embung



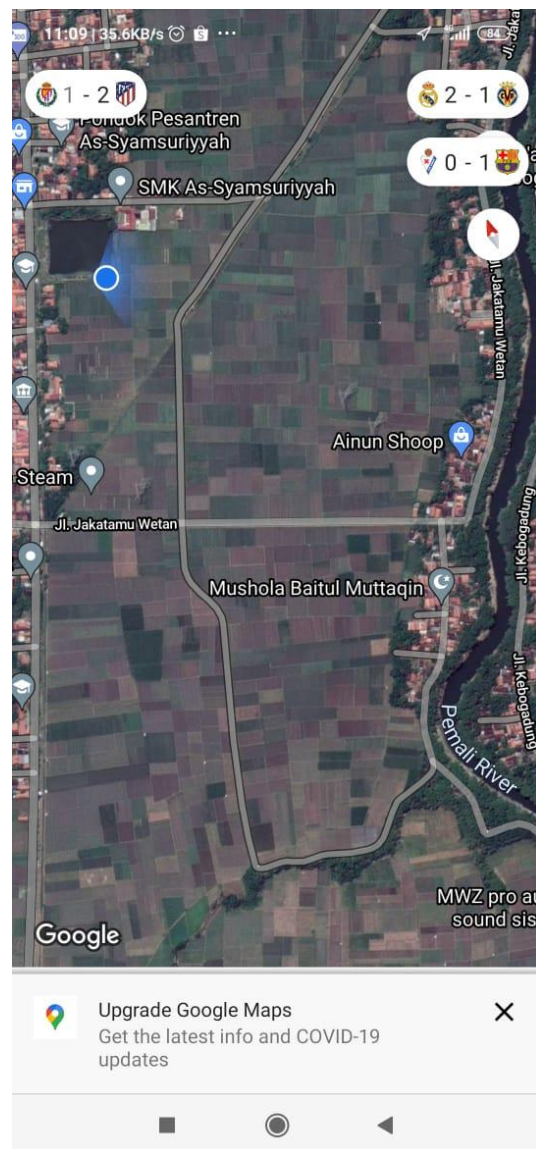
Gambar 3. 5 Kondisi air di embung yang rendah



Gambar 3. 6 Jalur irigasi menuju sawah / lahan tanaman bawang merah



Gambar 3. 7 Lokasi Embung Observasi 1



Gambar 3. 8 Lokasi Embung Observasi 2

3.2.2 Wawancara

Wawancara adalah salah satu metode pengumpulan data yang dilakukan melalui wawancara, yaitu suatu kegiatan untuk mendapatkan informasi secara langsung dengan memberikan pertanyaan-pertanyaan kepada responden. Dalam hal ini wawancara

dilakukan kepada petani bawang merah di desa jagalempeni, kecamatan wanasari, kabupaten Brebes.



Gambar 3. 9 Wawancara dengan petani yang sedang menanam bawang



Gambar 3. 10 Wawancara dengan petani yang sedang memompa air

3.3. Tools

1. Hardware
 - a. Node mcu
 - b. Relay 2 ch 5v
 - c. Psu
 - d. Motor *Servo*
 - e. Sensor *Ultrasonic*
 - f. Kabel Jumper
2. Software
 - a. Arduino IDE
 - b. Xampp

3.4. Waktu dan Tempat Penelitian

3.4.1 Waktu Penelitian

Waktu yang digunakan peneliti untuk penelitian ini dilaksanakan sejak bulan Desember 2020 dalam kurun waktu kurang lebih 5 (lima) bulan, 3 bulan pengumpulan data dan 2 bulan pengolahan data yang meliputi penyajian dalam bentuk tugas akhir serta proses bimbingan berlangsung.

3.4.2 Tempat Penelitian

Tempat Pelaksanaan Penelitian ini adalah di Desa kersana, Kecamatan kersana, Kabupaten Brebes dan Desa Jagalempeni, Kecamatan Wanasari, Kabupaten Brebes.

BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa Permasalahan

Bawang merah adalah salah satu komoditas unggulan di beberapa daerah di Indonesia, yang digunakan sebagai bumbu masakan dan memiliki kandungan beberapa zat yang bermanfaat bagi kesehatan, dan khasiatnya sebagai zat anti kanker dan pengganti antibiotik, penurunan tekanan darah, kolestrol serta penurunan kadar gula darah.

Tanaman bawang merah membutuhkan kondisi air tanah yang baik, yaitu air tanah dalam keadaan kapasitas lapang (lembab, tetapi tidak becek) sejak tumbuh hingga pembentukan umbi dan perkembangan umbinya. Kekeringan pada saat pertumbuhan vegetatif dapat menghambat pertumbuhan tanaman, sedangkan kekeringan pada saat pembentukan umbi dapat menggagalkan panen. Sebaliknya, tanah yang becek akan memudahkan berjangkitnya penyakit busuk umbi.

Adanya pemanasan global mengakibatkan terjadinya perubahan iklim yang tidak menentu beberapa tahun belakangan ini. Hal ini merupakan ancaman yang dihadapi dalam usaha pengembangan usaha budidaya bawang merah. Musim kemarau yang berkepanjangan menimbulkan adanya kekeringan di banyak lahan pertanian yang berujung pada gagal panen akibat kekurangan air. Berubahnya waktu musim hujan juga menyulitkan petani untuk menentukan masa tanam yang tepat. Demikian juga adanya

perubahan cuaca yang eksterm mengakibatkan pertumbuhan tanaman terganggu sehingga menimbulkan dampak menurunnya kualitas dan produktivitas lahan.

Berdasarkan masalah yang dipaparkan pada BAB I dapat disimpulkan bahwa pemantauan ketersediaan air pada ladang tanaman bawang merah masih menggunakan manual, sehingga dianggap belum maksimal. Untuk itu perlu dibuat sebuah alat dan *Hardware* yang akan dirancang dan dibangun yaitu sistem kontrol ketinggian air pada sawah bawang merah otomatis berbasis *IOT*. Dengan mengembangkan teknologi alat kontrol ketinggian air pada sawah bawang merah otomatis berbasis *IOT* ini, bermaksud agar kegiatan *monitoring* dan penyedotan air dari embung ke lahan tanaman bawang merah ini dapat berjalan lebih cepat dan menghemat waktu dalam pemeliharaan tanaman.

4.2 Analisa Kebutuhan Sistem

4.2.1 Analisa Kebutuhan *Hardware*

Kebutuhan *hardware* yang dimaksud yaitu perangkat keras yang digunakan untuk membuat sistem kontrol ketinggian air pada sawah bawang merah otomatis berbasis *IOT*. Adapun perangkat keras yang dibutuhkan, diantaranya sebagai berikut:

1. node mcu
2. sensor *Ultrasonic*
3. psu

4. motor *Servo*

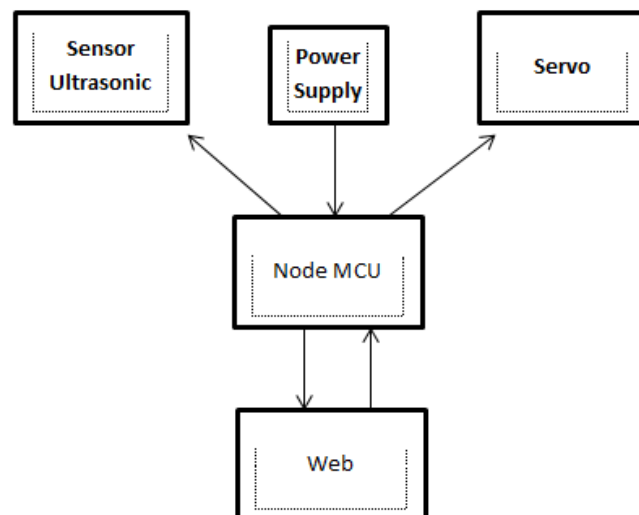
4.2.2 Analisa Kebutuhan *Software*

Kebutuhan *software* yaitu perangkat lunak yang digunakan untuk membuat Sistem *control* ketinggian air pada sawah bawang merah berbasis *IOT* yaitu Arduino IDE (*Integrated Development Environment*), Xampp dan Notepad++.

4.3 Perancangan Sistem

4.3.1 Perancangan Diagram Blok Sistem

Perancangan diagram blok merupakan gambaran ringkas antara masukan dan keluaran dari suatu sistem. Perancangan diagram blok untuk alat yang akan ditampilkan sebagai berikut :



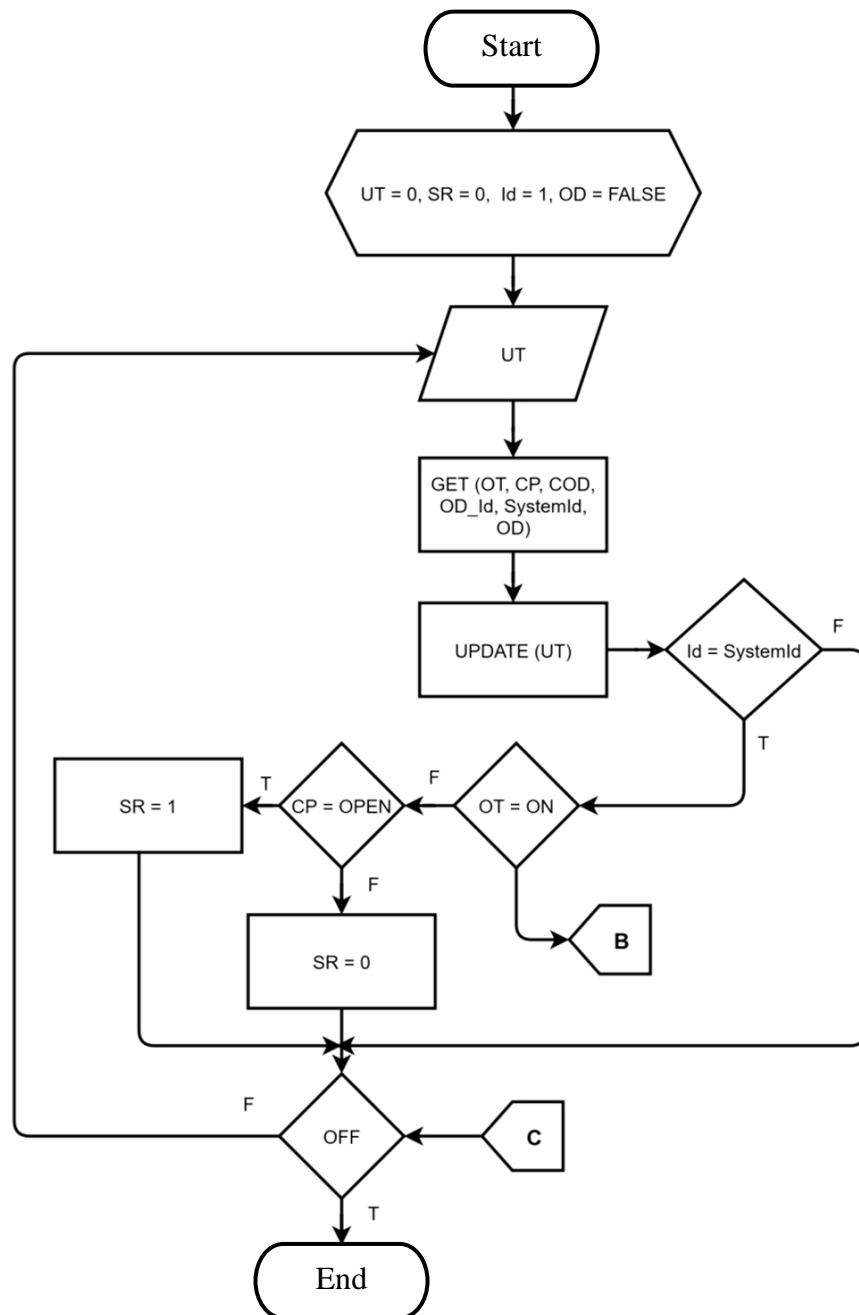
Gambar 4. 1 Diagram blok Rangkaian

Dari blok diagram rangkaian dapat dijelaskan sebagai berikut :

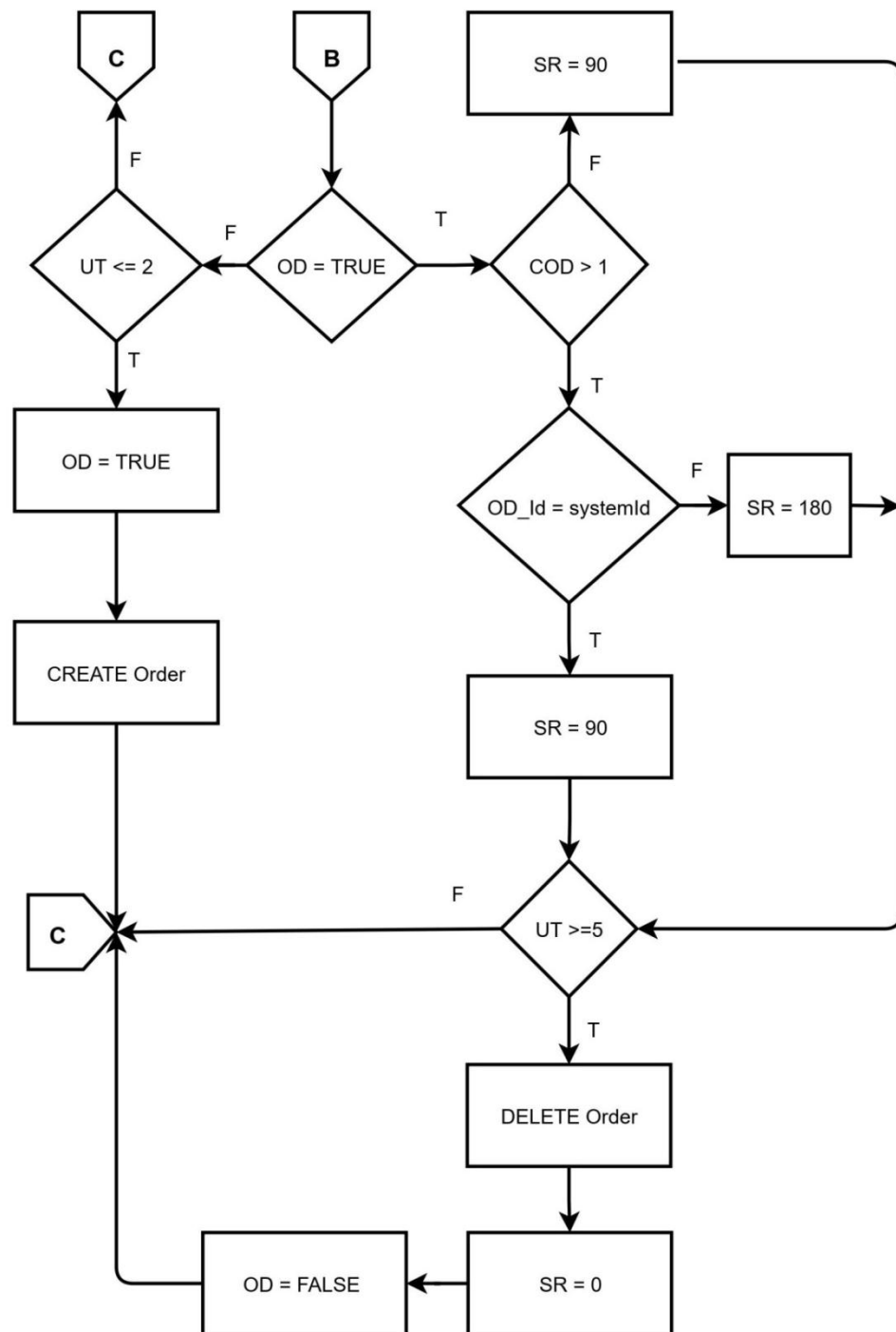
1. nodemcu ESP 8266 Berfungsi sebagai pengontrol dan pengolah data dari perangkat *input* dan *output* sensor serta sebagai pengirim data ke *Web Server*.
2. motor *servo* berfungsi sebagai penggerak buka tutup pintu air sawah.
3. sensor *ultrasonic* berfungsi untuk mendeteksi level ketinggian air pada sawah.
4. *power supply* sebagai penyuplai daya.

4.3.2 Flowchart

Flowchart adalah bagan alir yang menggambarkan tentang urutan langkah jalannya suatu program dalam sebuah bagan dengan simbol – simbol yang sudah ditentukan seperti berikut :



Gambar 4. 2 *Flowchart* Sistem kontrol ketinggian air sawah



Gambar 4. 3 *Flowchart* sistem kontrol ketinggian air sawah

Keterangan *Flowchart*:

1. T artinya *True* / Benar

2. *F* artinya *False* / Salah
3. *UT* adalah sebuah variabel yang digunakan untuk menampung hasil pembacaan nilai dari sensor *ultrasonic* yang ditempatkan di sawah, $UT = 0$ artinya dalam kondisi awal atau sawah dalam kondisi kosong tidak ada air.
4. *OD* adalah sebuah variabel yang digunakan sebagai input untuk sistem $OD = 0$ dalam kondisi awal atau tidak ada sawah yang membutuhkan aliran air.
5. *OT* adalah sebuah variable yang digunakan sebagai input, untuk sistem $OT = ON$ artinya dalam kondisi awal atau otomatis sistem dalam kondisi aktif atau *ON*.
6. *SR* (*Motor Servo*) yaitu untuk menggerakkan atau mengatur buka tutup pintu air, $SR = 0$ artinya dalam kondisi awal atau pintu air menutup air ke sawah.
7. *CP* adalah sebuah variabel yang digunakan sebagai input untuk sistem, $CP = OPEN$ artinya dalam mode manual *user* menginstruksikan kepada sistem untuk mengairi sawah
8. *GET* (*OT, CP, COD*) artinya sistem sedang mengambil data dari seb server kemudian data tersebut disimpan di variable *OT, CP, COD*.
9. *UPDATE* (*UT*) artinya sistem sedang menjalankan proses update nilai sensor *UT*.

10. COD artinya sistem membuat atau mengirim permintaan kepada *web server* untuk mengisi air.
11. SystemId sebagai validasi apakah Id sistem sudah terdaftar di *web server*.
12. SR = 90 pintu air terbuka namun hanya di arahkan ke sawah tersebut.
13. CP = ON validasi apakah *user* mengontrol.
14. SR = 180 artinya pintu air terbuka namun sawah dibelakang masih bisa mengisi air.
15. UT <= 2 artinya sawah sedang membutuhkan air.
16. COD > 1 artinya order lebih dari 1 atau sawah yang membutuhkan di aliri air lebih dari 1.
17. UT >= 5 artinya air sudah terisi penuh.
18. *Delete Order* artinya sistem menghapus permintaan pengisian air ke *web server*.
19. *Create Order* artinya sistem membuat atau mengirim permintaan kepada *web server* untuk mengisi air.
20. OD = TRUE artinya sistem sudah mengirim permintaan kepada *web server* untuk mengisi air.
21. OD = FALSE artinya tidak ada permintaan kepada *web server* untuk mengisi air.

4.3.3 Desain Input Output

Rangkaian komponen sistem kontrol ketinggian air pada sawah

bawang merah berbasis *IoT* guna mengurangi kekeringan akibat kekurangan air pada sawah bawang merah adalah sebagai berikut :

1. rangkaian *Nodemcu ESP8266*

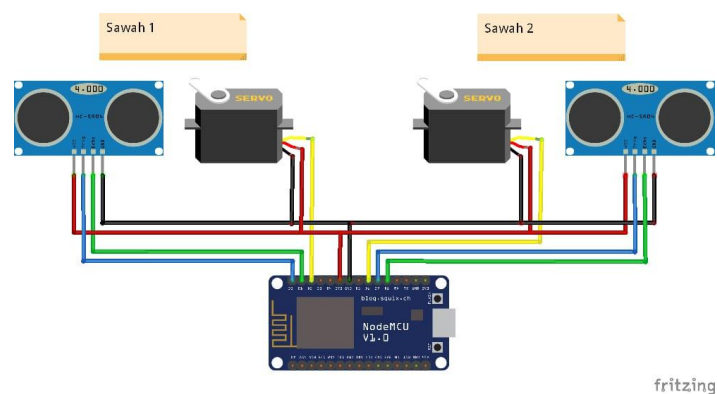
Komponen ini merupakan pusat rangkaian yang berfungsi sebagai pengendali komponen utama, *Nodemcu ESP8266* ini memiliki prosesor *Tensilica 32bit RISC CPU Xtensa LX106*, 7-12 V *input* tegangan, 16 Pin *Digital* (DIO), 1 Pin *Analog* (ADC), *UARTs* 1, *SPIs* 1, *I2Cs* 1, kecepatan memori 4 MB, *sRAM* 64 kb, kecepatan *clock* 80 MHz

2. rangkaian ultrasonic

Rangkaian ini dipasang untuk mendeteksi ketersediaan air, rangkaian ini akan dihubungkan ke *Nodemcu ESP8266* melalui pin D8 dan D7.

3. rangkaian Servo

Rangkaian ini dipasangkan sebagai *motor* penggerak buka tutup pintu air sawah, rangkaian ini dihubungkan ke *nodemcu* melalui Pin di D6 dan D1.



Gambar 4. 4 Rangkaian Alat / Komponen

1. Pin Vin digunakan sebagai input arus DC 5V
2. Pin D8 dihubungkan ke trig sensor *ultrasonic A*
3. Pin D7 dihubungkan ke Echo sensor *ultrasonic A*
4. Pin D3 dihubungkan ke trig sensor *ultrasonic B*
5. Pin D2 dihubungkan ke Echo sensor *ultrasonic B*
6. Pin D6 digunakan untuk *servo A*
7. Pin D1 digunakan untuk *servo B*

Setelah perancangan sistem secara blok per blok ditentukan, maka perancangan terakhir akan digambarkan secara keseluruhan. Rangkaian keseluruhan sistem ini akan memperlihatkan keterkaitan seluruh sistem yang ada, mulai dari *Nodemcu ESP8266* sebagai pusat dari pengendali utama sampai Sensor ultrasonic sebagai inputan untuk mendeteksi ketersediaan air dan *Motor servo* penggerak pintu aliran air sawah sebagai output, dan *website* sebagai *monitoring* dari alat tersebut.

BAB V

IMPLEMENTASI SISTEM

5.1 Implementasi Sistem

Implementasi sistem adalah prosedur-prosedur yang dilakukan dalam menyelesaikan konsep desain sistem yang telah dirancang sebelumnya. Agar sistem dapat beroperasi sesuai yang diharapkan, maka sebelumnya diadakan rencana implementasi atau uji coba dimaksudkan untuk mengatur biaya, waktu yang dibutuhkan, alat-alat yang dibutuhkan dan menguji sistem yang digunakan. Tahap ini merupakan tahap penerapan alat sistem kontrol otomatis agar siap untuk dioperasikan dan dapat digunakan sebagai pengembangan teknologi dalam hal pengairan atau manajemen air.

5.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan suatu proses instalasi alat atau perakitan alat yang digunakan dalam membangun sistem *prototype*.

5.1.2 Implementasi Perangkat Lunak dan Instalasi Aplikasi

Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula melakukan pemrograman. Sebelum dijual dipasaran IC Mikrokontroler arduino telah ditanamka suatu program bernama *bootloader* yang berfungsi sebagai penengah antara *compiler* arduino dengan mikrokontroler.

Arduino *IDE* dibuat dari bahasa pemrograman *JAVA*, arduino *IDE* juga dilengkapi dengan *Library C/C++* yang bisa disebut *Iwiring* yang membuat operasi input dan output menjadi lebih mudah.

5.2 Tahap Instalasi

Agar sistem ini berjalan sesuai rencana, maka ada tahap – tahap yang dilalui dalam pembuatan alat sistem pompa air otomatis ini.

5.2.1 Perancangan

Alat-alat yang digunakan dalam perancangan sistem control otomatis ini adalah berupa *software* dan *hardware* sebagai berikut :

1. install Arduiono *IDE* Bahasa pemrograman Arduino (*Sketch*) sudah dilakukan perubahan untuk memudahkan pemula melakukan pemrograman. Sebelum dijual dipasaran *IC* Mikrokontroler arduino telah ditanamka suatu program bernama *bootloader* yang berfungsi sebagai penengah antara *compiler* arduino dengan mikrokontroler. Arduino *IDE* dibuat dari bahasa pemrograman *JAVA*
2. *nodeMcu* berfungsi sebagai pengontrol dan pengolah data dari perangkat *input output*, serta sebagai pengirim data ke *webservice*.
3. *motor servo* berfungsi sebagai penggerak untuk membuka dan menutup pintu air sawah.
4. sensor *ultrasonic* berfungsi untuk mendeteksi ketersediaan / volume air dalam sawah.

5. kabel sebagai media penghubung untuk mengoneksikan pin yang harus dihubungkan agar alat berfungsi

Adapun komponen yang akan dijadikan sebagai sistem otomatisasi seperti pada tabel 5.1 dibawah ini.

Tabel 5. 1 Alat / Komponen yang digunakan

No	Nama Komponen	Keterangan
1	<i>Nodemcu</i>	Pengontrol dan pengolah data
2	Sensor <i>Ultrasonic</i>	Pendeteksi ketersediaan air
3	Motor <i>Servo</i>	Penggerak Pintu Air Sawah

Pada setiap pin *Nodemcu* dihubungkan dengan semua komponen, adapun pin-pin yang digunakan sesuai program yang dibuat. Untuk lebih jelas nya lihat tabel 5.2.

Tabel 5. 2 Pembagian Pin *Nodemcu*

No	Nama Pin	Keterangan
1	Pin Vin	Digunakan Sebagai input arus DC 5 V
2	Pin D8	Dihubungkan ke trig sensor <i>Ultrasonic A</i>
3	Pin 7	Dihubungkan ke Echo sensor <i>Ultrasonic A</i>
4	Pin 3	Dihubungkan ke trig sensor <i>Ultrasonic B</i>
5	Pin 2	Dihubungkan ke Echo sensor <i>Ultrasonic B</i>
6	Pin 6	Dihubungkan ke <i>Servo A</i>
7	Pin 1	Dihubungkan ke <i>Servo B</i>

5.2.2 Pembuatan

Langkah pembuatan Alat sistem kontrol otomatis ini terbagi menjadi 2 yaitu, pembuatan *hardware* (proses pembuatan atau merakit alat) dan Pembuatan *software* atau program nya yaitu dengan menggunakan *coding* yang kemudian di *compaile* menjadi *sketch* dan menanamkan ke dalam arduino uno dengan menggunakan mikrokontroler ESP8266.

5.2.3 Perakitan

Perakitan adalah suatu proses penyusunan / penggabungan beberapa bagian komponen menjadi satu alat atau mesin yang mempunyai fungsi tertentu atau sebagai langkah terakhir yang dilakukan sehingga menjadi produk jadi.



Gambar 5. 1 *Prototype* alat sistem kontrol ketinggian air pada sawah



Gambar 5. 2 *Prototype* alat sistem kontrol ketinggian air pada sawah

5.3 Hasil dan Pembahasan

5.3.1 Pengujian Sistem

Pengujian sistem dimaksud untuk menguji semua komponen-komponen perangkat keras seperti *nodemcu*, *servo*, sensor *ultrasonic* apakah sudah sesuai yang diharapkan tanpa adanya *error*.

Tabel 5. 3 Penjelasan Pengujian Sistem

Kelas Uji	Butir Uji	Alat Uji
Pengujian <i>input</i>	Pembacaan ketersediaan air	Sensor Ultrasonic
Pengujian <i>output</i>	Penampilan data ke <i>Website</i>	<i>Website Monitoring</i>
	Buka tutup pintu air sawah	<i>Servo</i>

Tabel 5. 4 Hasil pengujian alat

No	Pengujian	Yang di harapkan	Hasil 1	Hasil 2	Hasil 3
1	Sensor Ultrasonic	Mendeteksi volume dan ketersediaan air.	Fungsi	Fungsi	Fungsi
2	Servo	Membuka / menutup pintu air sawah	Fungsi	Fungsi	Fungsi

Sedangkan hasil pengujian unjuk kerja keseluruhan alatnya seperti ditampilkan pada tabel 5.5.

Tabel 5. 5 Hasil pengujian unjuk kerja keseluruhan alat

No	Level Air	Ultrasonic	Servo	Ket	Tanggal
1	Sawah 1 <=1 Sawah 2 >=3	Mengecek volum air pada sawah1 dan sawah2 dan membuat permintaan ke Webserver untuk mengisi air pada sawah1.	Servo 1= pintu terbuka dengan sudut 90 derajat. Servo2 = Menutup	Sawah1 teraliri air	02/06/2021 08:38:05
2	Sawah 1 >=3 Sawah 2 <=1	Mengecek volum air pada sawah1 dan sawah2 dan membuat permintaan ke Webserver untuk mengisi air pada sawah2.	Servo 1= Menutup Servo2 = pintu terbuka dengan sudut 90 derajat.	Sawah2 teraliri air	02/06/2021 08:40:47
3	Sawah 1 <=1 Sawah 2 <=1	Mengecek volum air pada sawah1 dan sawah2 dan membuat permintaan ke Webserver	Servo1= pintu terbuka dengan sudut 180 derajat.	Jika sawah 2 sudah terisi penuh maka pintu	

No	Level Air	Ultrasonic	Servo	Ket	Tanggal
		untuk mengisi air pada sawah1 dan sawah2	Servo2 = pintu terbuka dengan sudut 90 derajat	pada sawah2 (servo2 = 0) akan menutup dan sudut pintu sawah1 (servo1) akan berubah menjadi 90 derajat.	02/06/2021 08:42:44
4	Sawah 1 >=4 Sawah 2 >=4	Mengecek volum air pada sawah1 dan sawah2 dan mengirimkan data informasi ke webserver	Servo1 = 0, Servo2 = 0	Tidak ada kegiatan pengisian air ke sawah 1 maupun sawah 2	02/06/2021 08:45:00

Hasil pengujian alat rancang bangun sistem kontrol ketinggian air sawah bawang merah berbasis *IoT* diatas menunjukkan beberapa keadaan diantaranya yaitu:

1. pengujian dilakukan beberapa kali dengan air.
2. setelah *Ultrasonic* mendeteksi level air dan akan mengirim data ke *website monitoring*.
3. *website* akan menampilkan data yang di kirim dari sensor.
4. untuk servo akan berputar posisi membuka ketika irigasi sedang ada aliran air dan kondisi level air sawah rendah atau membutuhkan air dan otomatis menutup ketika level air pada

lahan sawah sudah cukup atau terpenuhi.

5.3.2 Cara Penggunaan Alat

Sistem kontrol ketinggian air pada sawah bawang merah berbasis *IoT* ini digunakan untuk meminimalisir terjadinya kekeringan pada lahan pertanian bawang merah yang dapat mengakibatkan kegagalan panen.

Alat ini akan mengirimkan data *volume* air ke *database web server*. Jika ada sawah yang membutuhkan air maka sensor akan mengirim data order ke *web server*.

Berikut ini adalah cara menggunakan alat pada *project* sistem *monitoring* irigasi otomatis pada sawah bawang merah :

1. *nodeMCU ESP8266* sambungkan kabel *USB* dari *NodeMCU* ke laptop, jika arduino sudah berhasil terhubung ke laptop maka akan muncul *port* yang akan digunakan pada pin *vin* dan *dng* pada *NodeMCU* disambungkan dengan *power supply* *vin* dengan kabel *positif* dan *gnd* dengan kabel *negative*
2. sensor *Ultrasonic Monitoring* air di sawah disambungkan kabel *jumper* dari sensor *ultrasonic* ke *NodeMCU ESP8266*, sesuaikan kabel pada masing-masing kaki sensor *ground*, *vcc*, *trig* dan *echo*. Pin *trig* yang digunakan adalah D8 dan pin *Echo* yang digunakan adalah D7.
3. *motor Servo* sambungkan kabel *jumper* ke *vcc*. Pin kabel positif dari *power supply*, lalu kabel *jumper* dihubungkan ke *ground*

dan hubungkan dengan kabel negative dari *power supply*. Setelah itu sambung kabel jumper dan *pin trig* yang digunakan adalah D1.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. rancang bangun sistem kontrol ketinggian air pada sawah bawang merah berbasis *IOT* untuk mengurangi terjadinya kekeringan pada lahan sawah bawang merah menggunakan *Nodemcu ESP8266* telah berhasil dirancang.
2. hasil pengujian menunjukkan *website* dapat menampilkan data level air atau ketinggian air yang dikirimkan dari sensor *ultrasonic*.
3. pengiriman data ke *web real time* memerlukan waktu beberapa detik agar data bisa tampil ke *web*.

6.2 Saran

Untuk pengembangan selanjutnya diperlukan masukan yang berupa saran agar nantinya produk hasil penelitian akan semakin baik dari segi bentuk maupun sistem untuk mencapai kesempurnaan dalam memenuhi kebutuhan, saran yang diperoleh adalah *prototype* dibuat semirip mungkin agar terlihat seperti miniatur sawah tanaman bawang merah.

DAFTAR PUSTAKA

- [1] Efriyani Sumastuti, Nuswantoro Setyadi Pradono, "DAMPAK PERUBAHAN IKLIM PADA TANAMAN PADI DI JAWA," *Jurnal Of Economic Education*, vol. Vol. 5, pp. 31-38, 2016.
- [2] Muhammad Salman Ibnu Chaer, Sirajuddin H. Abdullah, sih Priyati, "APLIKASI MIKROKONTROLER ARDUINO PADA SISTEM IRIGASI TETES UNTUK TANAMAN SAWI (*Brassica juncea*)," *urnal Ilmiah Rekayasa Pertanian dan Biosistem*, vol. Vol. 2, pp. 228-238, 2016.
- [3] Hariyanto, "ANALISIS PENERAPAN SISTEM IRIGASI UNTUK PENINGKATAN HASIL PERTANIAN DI KECAMATAN CEPU KABUPATEN BLORA," *Reviews In Civil Engineering*, vol. Vol. 4, pp. 29-34, 2018.
- [4] Linda Tri Wira Astuti, Arief Daryanto, Yusman Syaukat, Heny K Daryanto, "ANALISIS RESIKO PRODUKSI USAHATANI BAWANG MERAH PADA MUSIM KERING DAN MUSIM HUJAN DI KABUPATEN BREBES," *Jurnal Ekonomi Pertanian dan Agribisnis*, vol. Vol. 3, pp. 840-852, 2019.
- [5] Nanan Ariska, Diah Rachmawati, "PENGARUH KETERSEDIAAN AIR BERBEDA TERHADAP PERTUMBUHAN DAN HASIL TIGA KULTIVAR BAWANG MERAH (*Allium cepa* L.)," *Agrotek Lestari*, vol. Vol. 4, pp. 42-50, 2017.
- [6] Luluun Nuri Zamaniah, Tuty Handayani, Ratna Saraswat, "PENGARUH HUJAN EKSTREM TERHADAP PRODUKTIVITAS BAWANG MERAH DI KABUPATEN PROBOLINGGO JAWA TIMUR," *Prosiding Seminar Nasional Pendidikan Geografi FKIP UMP*, pp. 173-183, 2018.
- [7] Hendri Nurdin, Hasanuddin, Irzal, "OPTIMALISASI PEMANFAATAN MESIN POMPA UNTUK MENSUPLAI KEBUTUHAN AIR SAWAH TADAH HUJAN DI NAGARI SUMANI," *SEMINAR NASIONAL PENGABDIAN MASYARAKAT LPM UNIMED*, pp. 104-109, 2017.
- [8] SUMARDI SADI, ILHAM SYAH PUTRA, "RANCANG BANGUN MONITORING KETINGGIAN AIR DAN SISTEM KONTROL PADA PINTU AIR BERBASIS ARDUINO DAN SMS GATEWAY," *Jurnal*

- Teknik: Universitas Muhammadiyah Tangerang*, vol. Vol. 7, pp. 77-91, 2018.
- [9] Subianto, Paulus Lucky Tirma Irawan, Shenata Hanadam Shienjaya, "Rancang Bangun Sistem Monitoring Level Air Bendungan Untuk Pengendalian Banjir," *SMATIKA Jurnal* , vol. Vol. 19, pp. 39-44, 2019.
- [10] Umam, Kurniawan Gigig L, *Smart Kandang Ayam Petelur Berbasis Internet Of Things*, 2018.
- [11] N. Documentation, "www.nodemcu.com," 17 april 2016. [Online].
- [12] D. Naista, "Codeigniter Vs Laravel," Yogyakarta, CV. Lokomedia, 2017.
- [13] A. Kadir, *Buku Pintar Pemrograman Arduino*, Melaka: Mediakom, 2014.
- [14] M. F. Payuda, "Rancang Bangun Sistem Informasi Penjualan Pada PT. Duta Perfume Berbasis Web Menggunakan Metode Sekuensial," *Jurnal Algoritma, Logika dan Komputasi*, vol. Vol. III (No.1), pp. 229-237, 2020.
- [15] K. Y, *Aplikasi Web Database dengan PHP dan MySQL*, Jakarta: PT Elex Media Komputindo, 2002.
- [16] R. Astamal, *Mastering Code HTML*, 2002.
- [17] B. A. N. D. F. Fery Sofian Efendi, *Aplikasi Tempat Kos di Kota Kediri Berbasis*, Kediri, 2015.
- [18] J. W. Gilmore, *Beginning PHP and*, New York: Appres, 2006.
- [19] Z. A. & S. Community, *36 Menit Belajar Komputer: PHP dan MySql*, Jakarta: Elex Media Komputindo, 2008.
- [20] A.-B. b. Ladjamudin, *Analisis dan Desain Sistem Informasi*, Yogyakarta: Graha Ilmu, 2005.
- [21] M. M. d. Oktafianto, *Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML*, Yogyakarta: Andi Offset, 2015.
- [22] A. A. A. Soni, ""Distance Measurement of an Object by using Ultrasonic Sensors with Arduino and GSM Module," *International Journal of Science Technology & Engineering*, vol. Vol.4 No.11, pp. 23-28, 2018.

- [23] C. Y. d. Andani, "Sistem Kendali Servo Posisi dan Kecepatan Motor dengan Programmable Logic Control (PLC)," *Jurnal Ilmiah Foristek*, Vols. Vol.11, No.2, 2011.
- [24] J. Susilo, Aplikasi on/off pompa air otomatis berbasis sensor ultrasonic., 2015.
- [25] R. Basic. [Online]. Available: www.physics.unlv.edu/bill/PHYS483/relay.pdf. [Accessed 15 03 2016].
- [26] M. H. Rashid, Power Electronics, Printice Hall International, 1993.
- [27] A. Razor, "ALDYRAZOR.COM," 2020. [Online]. Available: <https://www.aldyrazor.com/2020/04/kabel-jumper-arduino.html>. [Accessed 20 06 2020].

LAMPIRAN

Lampiran 1 Surat Kesediaan Membimbing TA Pembimbing I

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini :

Nama : Mohammad Humam, M.Kom
NIDN : 0618117901
NIPY : 12.002.007
Jabatan Struktural : Kepala Bagian Pengembangan Bisnis
Jabatan Fungsional : Asisten Ahli

Dengan ini menyatakan bersedia untuk menjadi pembimbing I pada Tugas Akhir mahasiswa berikut :

No	Nama	NIM	Program Studi
1	Zakaria	18041094	DIII Teknik Komputer

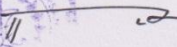
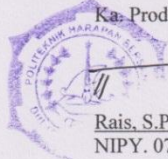
Judul TA : "RANCANG BANGUN SISTEM MONITORING IRIGASI OTOMATIS PADA SAWAH BAWANG MERAH BERBASIS IOT (*INTERNET of THINGS*)".

Demikian pernyataan ini dibuat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, April 2021

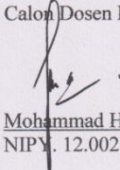
Mengetahui,

Ka. Prodi DIII Teknik Komputer

Rais, S.Pd., M.Kom.
NIPY. 07.011.083

Calon Dosen Pembimbing I,



Mohammad Humam, M.Kom
NIPY. 12.002.007

Lampiran 2 Surat Kesiediaan Membimbing TA Pembimbing II

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan dibawah ini :

Nama : Abdul Basit S.Kom., MT
NIDN : -
NIPY : 01.015.198
Jabatan Struktural : Koordinator Kemahasiswaan
Jabatan Fungsional : Koordinator Kemahasiswaan

Dengan ini menyatakan bersedia untuk menjadi pembimbing 1 pada Tugas Akhir Mahasiswa berikut :

NO	NAMA	NIM	Program Studi
1	Zakaria	18041094	DIII Teknik Komputer

Judul TA : "RANCANG BANGUN SISTEM MONITORING IRIGASI OTOMATIS PADA SAWAH BAWANG MERAH BERBASIS IOT (*INTERNET of THINGS*)".

Demikian pernyataan ini di buat agar dapat dilaksanakan sebagaimana mestinya.

Tegal, Juni 2021

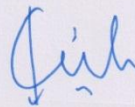
Mengetahui,

Ka. Prodi DIII Teknik Komputer




Basit, S.PD., M.Kom.
NIPY. 07.011.083

Calon Dosen Pembimbing 2,



Abdul Basit S.Kom., MT.
NIPY. 01.015.198

Lampiran 3 Surat Keterangan Observasi

**PEMERINTAH KABUPATEN BREBES**
DINAS PENGELOLAAN SUMBER DAYA AIR
DAN PENATAAN RUANG
Jl. Hasanuddin (Work shop) Komp. RSS Gandasuli, Brebes - 52215
Telepon / Fax .(0283) 6174144

Brebes, 16 Juni 2021

Nomor : 032.03 / 1189 / 2021
Sifat :
Perihal : **Balasan Ijin Observasi**

Kepada
Yth. Ka. Prodi DIII Teknik Komputer
Politeknik Harapan Bersama
di
TEGAL


Dengan ini Kepala Dinas Pengelolaan Sumber Daya Air dan Penataan Ruang Kabupaten Brebes menerangkan bahwa mahasiswa di bawah ini :

No	Nama	NIM	No. HP
1	ZAKARIA	18041094	081902246401
2	SHOFIYUN	17041094	085747705788
3	RAUDHOTUL JHANNAH	18040219	089658019061

Telah melaksanakan Observasi pengambilan data di Dinas Pengelolaan Sumber Daya Air dan Penataan Ruang Kabupaten Brebes untuk memenuhi mata kuliah tugas akhir. Demikian surat keterangan ini kami buat untuk dapat digunakan sebagaimana mestinya.

Jl. No. 100	Pesert	Tgl
Staf. 1000/100	<i>[Signature]</i>	16/6/21
Ka. Brebes		
Ka. / Ka. Brebes	<i>[Signature]</i>	16/6/21

**Kepala Dinas Pengelolaan Sumber Daya Air
Dan Penataan Ruang
Kabupaten Brebes**



AGUS AS'ARI, BAE, ST, MT.
Pembina Utama Muda
NIP. 19630810 198709 1 002

Tembusan :
1. Arsip.

Lampiran 4 Foto Kegiatan Observasi



Jalur masuk air irigasi ke sawah



Proses kegiatan penanaman bawang merah

Lampiran 5 Foto Kegiatan Observasi

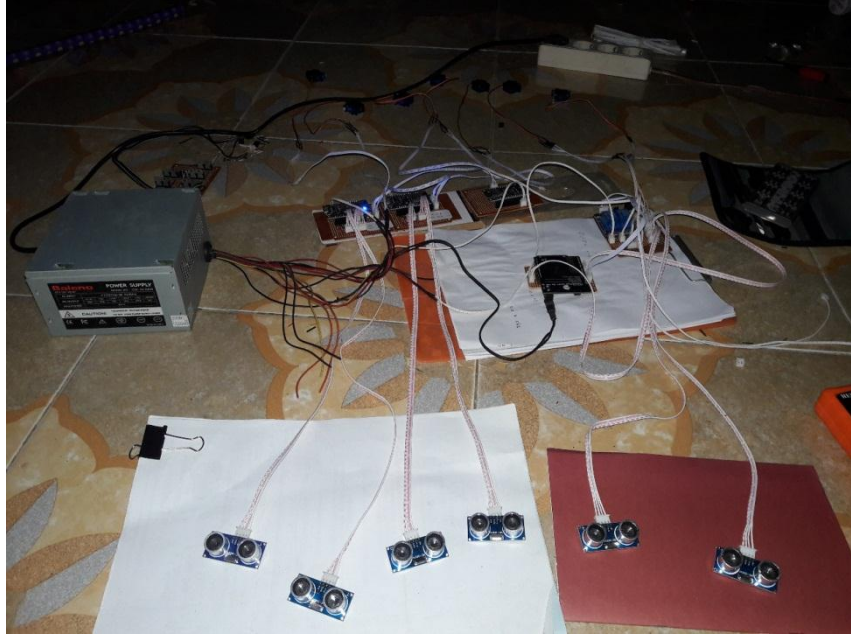


Jalur masuk air irigasi ke sawah yang tertutup sampah

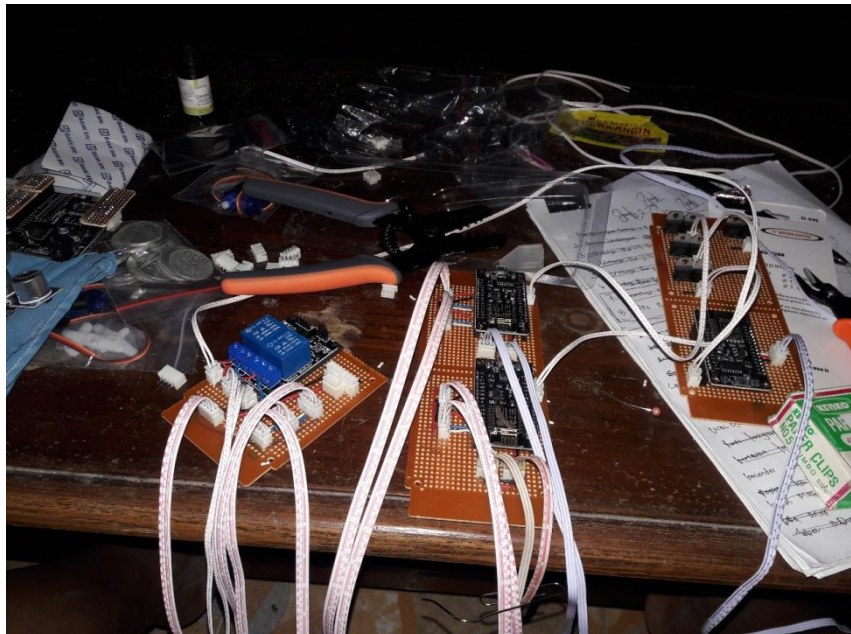


Sawah yang memiliki instalasi listrik mandiri

Lampiran 6 Foto Perakitan



Sensor ultrasonic



Part sebelum dirakit

Lampiran 7 Foto Perakitan



Prototype sawah



Prototype embung dan jalur irigasi sawah

Lampiran 8 Source Code

```
// Include Library
#include <Servo.h>
#include <ESP8266WiFi.h>
#include <ArduinoJson.h>
#include <ESP8266HTTPClient.h>

Servo Servo_A;
Servo Servo_B;

/*=====
 * Deklarasi Variabel
 *=====
 * Note :
 * UT_A = Variabel untuk menampung input dari sensor
 * ultra sonic yang di tempatkan sawah A
 * UT_B = Variabel untuk menampung input dari sensor
 * ultra sonic yang di tempatkan sawah A
 * OT_A = Otomatis
 *
 *
 *=====
 */
int OD_A = 0, OD_B = 0, pintu1 = 180, pintu2 = 180;
float UT_A=0, UT_B=0;
int OT_A=0, OT_B=0, CP_A=0, CP_B=0, COD=0, OD_Id = 0;
int ID_A = 1, ID_B = 2, SystemId_A = 0, SystemId_B = 0;
//-----

// Configurasi Password Dan SSID Wifi
const char* ssid = "AndroidAP";
const char* password = "fhy140995";

void setup () {

  Serial.begin(9600);
  WiFi.begin(ssid, password);

  // Setup PIN Mode Dan PIN GPIO
  Servo_B.attach(5);
  Servo_A.attach(2);

  Servo_A.write(180);
  Servo_B.write(180);

  pinMode(12, OUTPUT); // TRIG Sensor UT A
  pinMode(13, INPUT); // ECHO Sensor UT A
  pinMode(4, OUTPUT); // TRIG Sensor UT B
  pinMode(0, INPUT); // ECHO Sensor UT B

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
}
```

```

/*
=====
*| Method Sensor |
=====
* Note :
* Method ini di gunakan untuk mengecek level
* Air. Method ini mempunyai dua parameter
* yaitu trig dan echo semuanya bertipe
* integer Trig adalah pin Triger
* untuk sensor Ultra Sonic dan
* Echo adalah pin Echo pada
* Sensor Ultra Sonic
*-----
*/
void sensor(int trig, int echo){

    float duration, level;

    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(100);
    digitalWrite(trig, LOW);

    duration = pulseIn(echo, HIGH);
    level = (duration/2) / 29.1;

    switch(trig){
    case 12:
        UT_A = 9 - level;
        Serial.print("1. Update Level AIR A : ");
        Serial.println(UT_A);
        break;
    case 4:
        UT_B = 8 - level;
        Serial.print("2. Update Level AIR B : ");
        Serial.println(UT_B);
        break;
    }
    delay(2000);
}
//-----

/*
=====
=====
*| Method GET Data |
=====
=====
* Note :
* Method ini berfungsi untuk mendapatkan data dari Web Api
Services

```



```

* data yang di dapat di sini adalah data Sistem kontrol yang akan
* di gunakan untuk mengontrol system monitoring sawah.
*-----
-----
*/
void getData(String systemId) {

    HTTPClient http;

    http.begin("http://irigasiotomatis.com/Laravel/public/api/monitoring/"+systemId);
    int httpCode = http.GET();

    if(httpCode > 0){
        char json[500];
        String payload = http.getString();
        payload.toCharArray(json, 500);
        StaticJsonDocument<200> doc;
        deserializeJson(doc, json);

        Serial.print("Http Code Response : ");
        Serial.println(httpCode);
        Serial.println("=====");
        Serial.println();

        if(httpCode == 200){

            /*=====
            | Parsing data JSON yang di dapat dari WEB API Sevices
            =====*/
            if(systemId == "1"){
                Serial.println("A. Data Sawah A");
                SystemId_A = doc["data"]["system_id"];
                OT_A = doc["data"]["otomatis"];
                CP_A = doc["data"]["pintu_air"];
                OD_A = doc["data"]["status"];
                COD = doc["count_order"];
                OD_Id = doc["order_id"];

                Serial.print("system_Id : ");
                Serial.println(SystemId_A);
                Serial.print("otomatis : ");
                Serial.println(OT_A);
                Serial.print("Control Pintu : ");
                Serial.println(CP_A);
                Serial.print("status : ");
                Serial.println(OD_A);
                Serial.print("Count Order : ");
                Serial.println(COD);
                Serial.print("Order Id : ");
                Serial.println(OD_Id);

            }else{

```

```

Serial.println();
Serial.println("B. Data Sawah B");
SystemId_B = doc["data"]["system_id"];
OT_B = doc["data"]["otomatis"];
CP_B = doc["data"]["pintu_air"];
OD_B = doc["data"]["status"];
COD = doc["count_order"];
OD_Id = doc["order_id"];

Serial.print("system_Id : ");
Serial.println(SystemId_B);
Serial.print("otomatis : ");
Serial.println(OT_B);
Serial.print("Control Pintu : ");
Serial.println(CP_B);
Serial.print("status : ");
Serial.println(OD_B);
Serial.print("Count Order : ");
Serial.println(COD);
Serial.print("Order Id : ");
Serial.println(OD_Id);

}
}
}
http.end();
delay(500);

}

/*
=====
*| Methode POST Order |
=====
* Note :
* Method ini berfungsi untuk membuat sebuah permintaan ke system
* pompa irigasi untuk mengisi sawah. Method ini mempunyai
* parameter bertipe string yaitu systemId. SystemId merupakan
* identitas dari system monitoring sawah tersebut.
*-----
-----
*/
void postOrder(String systemId) {
    HTTPClient http;

    String Data = "system_id=" + systemId ;

    http.begin("http://irigasiotomatis.com/Laravel/public/api/monitoring");

```

```

    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
    int httpCode = http.POST(Data);

    switch(httpCode){
    case 200:
    Serial.print("Http Code Response : ");
    Serial.println(httpCode);
    Serial.println(" Send Permintaan Pengisian Air Sucess!");
    break;
    default:
    Serial.println(Data);
    Serial.print("Http Code Response : ");
    Serial.println(httpCode);
    Serial.println(" Send Permintaan Pengisian Air Failed!");
    break;
    }

    http.end();
    delay(500);

}

/*
*=====
*=====
*| Method PUT Sensor |
*=====
*=====
*Note :
* Method ini di gunakan untuk mengupdate data sensor yang ada di
database, Untuk
* Method sensor ini memiliki Dua parameter yaitu Sensor dan
systemId Kedua
* parameter tersebut bertipe String. Sensor di sini berisi data
volume
* Air yang ada di Sawah. Nilai Volume air tersebut di dapat dari
hasil
* pembacaan dari sensor ultra sonic yang di tempatkan sawah,
Sedangkan
* SystemId merupakan identitas atau Id dari system monitoring
sawah tersebut.
*-----
*-----
*/
void putSensor(String sensor, String systemId){
    HTTPClient http;

    String Data = "sensor=" + sensor+"&system_id=" + systemId;

    http.begin("http://irigasiotomatis.com/Laravel/public/api/monitori
ng");

```

```

    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
    int httpCode = http.PUT(Data);

    switch(httpCode){
    case 200:
    Serial.print("Http Code Response : ");
    Serial.println(httpCode);
    Serial.println(" Update Sensor dan status order Success");
    break;
    default:
    Serial.print("Http Code Response : ");
    Serial.println(httpCode);
    Serial.println(" Update Sensor dan status order Failed");
    break;
    }

    http.end();
    delay(500);
}
//-----
-----

/*
*=====
=====
*| Method Delete permintaan ke pompa |
*=====
=====
*Note :
* Method ini di gunakan untuk mendelete permintaan pengisian air
dari system pompa
* air irigasi otomatis,ketika sawah sudah selesai mengisi air
dengan penuh.
* ataupun ketika sedang mengisi air dari embung dan kemudian air
di
* saluran irigasi aktif. Method ini mempunyai 1 Parameter
systemId.
* systemId ini bertipe String dan systemId ini merupakan identias
* dari id sawah tersebut.
*-----
-----
*/
void deleteOrder(String systemId){
    HTTPClient http;

    http.begin("http://irigasiotomatis.com/Laravel/public/api/monitori
ng/destroy/" + systemId);
    int httpCode = http.GET();

```

```

switch(httpCode) {
case 200:
Serial.print("Http Code Response : ");
Serial.println(httpCode);
Serial.println("Delete permintaan Success!");
break;
default:
Serial.print("Http Code Response : ");
Serial.println(httpCode);
Serial.println("Delete permintaan Failed");
break;
}

http.end();
delay(500);
}
//-----
-----

/*
*
=====
=====
* | Method Pintu Close
*
=====
=====
*/
void pintuClose(int systemId) {
switch(systemId) {
case 1 :
switch(pintu1) {
case 0:
Servo_A.write(pintu1);
delay(1000);
Servo_A.write(90);
delay(1000);
Servo_A.write(180);
delay(1000);
pintu1 = 180;
break;
case 90:
Servo_A.write(pintu1);
delay(1000);
Servo_A.write(180);
delay(1000);
pintu1 = 180;
break;
}
break;
case 2 :
switch(pintu2) {
case 0:
Servo_B.write(pintu2);

```

```

delay(1000);
Servo_B.write(90);
delay(1000);
Servo_B.write(180);
delay(1000);
pintu2 = 180;
break;
case 90:
Servo_B.write(pintu2);
delay(1000);
Servo_B.write(180);
delay(1000);
pintu2 = 180;
break;
}
break;
}
}

//-----
-----

/*
*
=====
=====
* | Method Pintu Open
*
=====
=====
*
*
*/
void pintuOpen(int sudut, int systemId){
switch(systemId){
case 1:
switch(sudut){
case 1:
switch(pintu1){
case 180:
Servo_A.write(pintu1);
delay(1000);
Servo_A.write(90);
delay(1000);
pintu1 = 90;
break;
case 0:
Servo_A.write(pintu1);
delay(1000);
Servo_A.write(90);
delay(1000);
pintu1 = 90;
break;
}
break;
}
}
}

```

```

case 2:
switch(pintu1){
case 180:
Servo_A.write(pintu1);
delay(1000);
Servo_A.write(90);
delay(1000);
Servo_A.write(0);
delay(1000);
pintu1 = 0;
break;
case 90:
Servo_A.write(pintu1);
delay(1000);
Servo_A.write(0);
delay(1000);
pintu1 = 0;
break;
}
break;
}
break;
case 2:
switch(sudut){
case 1:
switch(pintu2){
case 180:
Servo_B.write(pintu2);
delay(1000);
Servo_B.write(90);
delay(1000);
pintu2 = 90;
break;
case 0:
Servo_B.write(pintu2);
delay(1000);
Servo_B.write(90);
delay(1000);
pintu2 = 90;
break;
}
break;
case 2:
switch(pintu2){
case 180:
Servo_B.write(pintu2);
delay(1000);
Servo_B.write(90);
delay(1000);
Servo_B.write(0);
delay(1000);
pintu2 = 0;
break;
case 90:

```

```

Servo_B.write(pintu1);
delay(1000);
Servo_B.write(0);
delay(1000);
pintu2 = 0;
break;
}
break;
}
break;
}
}
//-----
-----

void loop() {
  Serial.println("=====");
  Serial.println(" Process Method Sensor");
  Serial.println("=====");
  Serial.println();
  sensor(12, 13); // Mengecek Level Air Sawah A
  Serial.println("-----");
  Serial.println();
  Serial.println();

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("=====");
    Serial.println(" Process GET DATA");
    Serial.println("=====");
    Serial.println();
    getData(String(ID_A));
    Serial.println();
    Serial.println("-----");
    Serial.println();
    Serial.println();

    Serial.println("=====");
    Serial.println(" Process Method PUT Sensor ");
    Serial.println("=====");
    Serial.println();
    putSensor(String(UT_A), String(ID_A));
    Serial.println();
    Serial.print("1. Volume Air Sawah A : ");
    Serial.println(UT_A);
    Serial.println();
    Serial.println("-----");
    Serial.println();
    Serial.println();

    // =====
    // Pengecekan Sysytem Sawah A
    /* =====
    * Note :
    * Servo_A.write(0) == Pintu AIR CLOSE
    * Servo_A.write(90) == Pintu AIR OPEN Air Hanya

```



```

* di alirkan ke sawah itu saja
* Servo_A.write(180) == Pintu AIR OPEN Air di
* alirkan ke sawah tersebut namun sawah di
* depan bisa mengairi
* OD_A = 1
* OD_A = 0
* ===== */
// Control Flow Otomatis System Irigasi

Serial.println("=====
=====");
Serial.println(" Test System Sawah A ");

Serial.println("=====
=====");
Serial.println();
Serial.print("system Id : ");

if(SystemId_A == ID_A){
Serial.println("Terdaftar");
Serial.print("Status : ");
switch(OT_A){
case 1:
Serial.println("Otomatis ON");
// Control Flow Status Order
Serial.print(" 1. Status Permintaan Pengisian AIR : ");
switch(OD_A){
case 1:
Serial.println("BENAR");
Serial.print(" 1. Jumlah sawah yang membutuhkan air : ");
Serial.println(OD_Id);
if(COD > 1){
Serial.println(" a. Kondisi : COD > 1");
Serial.println(" a. Sawah yang membutuhkan pengisian air lebih
dari 1");
Serial.print(" a. Order Id sawah paling Ujung : ");
Serial.println(OD_Id);
if(OD_Id == ID_A){
Serial.println(" ~ Kondisi : OD_Id == ID_A");
Serial.println(" ~ ID Order AIR sawah paling ujung adalah id
sawah ini");
Serial.print(" + Pintu Terbuka Dengan sudut 90 Derajat");
pintuOpen(1, ID_A);
}else{
Serial.println(" ~ Kondisi : OD_Id != ID_A");
Serial.println(" ~ ID Order AIR sawah paling ujung BUKAN id sawah
ini");
Serial.print(" + Pintu Terbuka Dengan sudut 180 Derajat");

```

```

pintuOpen(2, ID_A);
}
}else{
Serial.println(" a. Kondisi : COD == 1");
Serial.println(" a. Sawah yang membutuhkan pengisian air hanya
1");
Serial.print(" + Pintu Terbuka Dengan sudut 90 Derajat");
pintuOpen(1, ID_A);
}
Serial.println();
Serial.println();
Serial.print(" 2. Progres Pengisian Air : ");
Serial.println(UT_A);
if(UT_A >= 3){
Serial.println(" a. Kondisi : UT_A >= 3");
Serial.println(" b. Proses Pengisian Air Sudah Selesai!! . OD =
0/FALSE ");
Serial.println(" c. System Menghapus Permintaan Pengisian air Ke
Web Server ");
OD_A = 0;
deleteOrder(String(ID_A));
pintuClose(ID_A);
Serial.println(" d. System Menutup Pintu AIR");
}
break;
case 0:
Serial.println("SALAH");
Serial.print(" 2. System Mengecek Volume Air Di sawah : ");
Serial.println(UT_A);
if(UT_A <= 1){
Serial.println(" a. Kondisi : UT_A <= 1");
Serial.println(" b. Sawah Sudah Harus Di isi : OD : 1/BENAR");
Serial.println(" c. System Membuat permintaan ke WEB Server Untuk
mengisi air");
OD_A = 1;
postOrder(String(ID_A));
pintuOpen(1, ID_A);
}
break;
}
break;
case 0:
Serial.println("Otomatis OFF");
switch(CP_A){
case 0:
Serial.println("1. Pintu Close");
pintuClose(ID_A);
break;
case 1:
Serial.println("1. Pintu OPEN");
pintuOpen(1, ID_A);

```

```

break;
case 2:
Serial.println("1. Pintu Open Full");
pintuOpen(2, ID_A);
break;
}
break;
}
}else{
Serial.println("Tidak Terdaftar");
}

// =====
Serial.println();
Serial.println("-----");
Serial.println();
Serial.println();
delay(500);

// =====
// Controlflow Sysytem Monitoring Sawah B
/* =====
* Note :
* Servo_A.write(0) == Pintu AIR CLOSE
* Servo_A.write(90) == Pintu AIR OPEN
* ===== */
Serial.println("=====");
Serial.println(" Process Method Sensor");
Serial.println("=====");
Serial.println();
sensor(4, 0); // Mengecek Level Air Sawah B
Serial.println("-----");
Serial.println();
Serial.println();

Serial.println("=====");
Serial.println(" Process GET DATA");
Serial.println("=====");
Serial.println();
getData(String(ID_B));
Serial.println();
Serial.println("-----");
Serial.println();
Serial.println();

Serial.println("=====");
Serial.println(" Process Method PUT Sensor ");
Serial.println("=====");
Serial.println();
putSensor(String(UT_B), String(ID_B));
Serial.println();
Serial.print("2. Volume Air Sawah B : ");
Serial.println(UT_B);
Serial.println();

```

```

Serial.println("-----");
Serial.println();
Serial.println();

Serial.println("=====
=====");
Serial.println(" Test System Sawah B ");

Serial.println("=====
=====");

Serial.println();
Serial.print("system Id : ");

if(SystemId_B == ID_B){
Serial.println("Terdaftar");
Serial.print("Status : ");
switch(OT_B){
case 1:
Serial.println("Otomatis ON");
// Control Flow Status Order
Serial.print(" 1. Status Permintaan Pengisian AIR : ");
switch(OD_B){
case 1:
Serial.println("BENAR");
Serial.print(" 1. Jumlah sawah yang membutuhkan air : ");
Serial.println(OD_Id);
if(COD > 1){
Serial.println(" a. Kondisi : COD > 1");
Serial.println(" a. Sawah yang membutuhkan pengisian air lebih
dari 1");
Serial.print(" a. Order Id sawah paling Ujung : ");
Serial.println(OD_Id);
if(OD_Id == ID_B){
Serial.println(" ~ Kondisi : OD_Id == ID_B");
Serial.println(" ~ ID Order AIR sawah paling ujung adalah id
sawah ini");
Serial.print(" + Pintu Terbuka Dengan sudut 90 Derajat");
pintuOpen(1, ID_B);
}else{
Serial.println(" ~ Kondisi : OD_Id != ID_B");
Serial.println(" ~ ID Order AIR sawah paling ujung BUKAN id sawah
ini");
Serial.print(" + Pintu Terbuka Dengan sudut 180 Derajat");
pintuOpen(2, ID_B);
}
}
}
}

```

```

}else{
  Serial.println(" a. Kondisi : COD == 1");
  Serial.println(" a. Sawah yang membutuhkan pengisian air hanya
1");
  Serial.print(" + Pintu Terbuka Dengan sudut 90 Derajat");
  pintuOpen(1, ID_B);
}
Serial.println();
Serial.println();
Serial.print(" 2. Progres Pengisian Air : ");
Serial.println(UT_B);
if(UT_B >= 3){
  Serial.println(" a. Kondisi : UT_B >= 3");
  Serial.println(" b. Proses Pengisian Air Sudah Selesai!! . OD =
0/FALSE ");
  Serial.println(" c. System Menghapus Permintaan Pengisian air Ke
Web Server ");
  OD_B = 0;
  deleteOrder(String(ID_B));
  pintuClose(ID_B);
  Serial.println(" d. System Menutup Pintu AIR");
}
break;
case 0:
  Serial.println("SALAH");
  Serial.print(" 2. System Mengecek Volume Air Di sawah : ");
  Serial.println(UT_B);
  if(UT_B <= 1){
    Serial.println(" a. Kondisi : UT_B <= 1");
    Serial.println(" b. Sawah Sudah Harus Di isi : OD : 1/BENAR");
    Serial.println(" c. System Membuat permintaan ke WEB Server Untuk
mengisi air");
    OD_B = 1;
    postOrder(String(ID_B));
    pintuOpen(1, ID_B);
  }
  break;
}
break;
case 0:
  Serial.println("Otomatis OFF");
  switch(CP_B){
  case 0:
    Serial.println("1. Pintu Close");
    pintuClose(ID_B);
    break;
  case 1:
    Serial.println("1. Pintu OPEN");
    pintuOpen(1, ID_B);
    break;
  case 2:

```

```
Serial.println("1. Pintu Open Full");
pintuOpen(2, ID_B);
break;
}
break;
}
}else{
Serial.println("Tidak Terdaftar");
}
//
=====
===

}
Serial.println();
Serial.println("-----");
Serial.println();
Serial.println();
Serial.println();
delay(500);
```