



TUGAS AKHIR - EE184801

**Sistem Monitoring dan Kontrol pada Irigasi Tetes Berbasis IoT
Menggunakan LoRa**

Mabarrus Sholeh
NRP 0711154000015

Dosen Pembimbing
Dr. Muhammad Rivai, ST.MT.
Muhammad Attamimi B.Eng., M.Eng., Ph.D.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR - EE184801

**Sistem Monitoring dan Kontrol pada Irigasi Tetes Berbasis IoT
Menggunakan LoRa**

Mabarrus Sholeh
NRP 0711154000015

Dosen Pembimbing
Dr. Muhammad Rivai, ST.MT.
Muhammad Attamimi B.Eng., M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



FINAL PROJECT - EE184801

Monitoring and Controlling System on Drip Irrigation Based on IoT with LoRa

Mabarrus Sholeh
NRP 0711154000015

Supervisor
Dr. Muhammad Rivai, ST.MT.
Muhammad Attamimi B.Eng., M.Eng., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Sistem Monitoring dan Kontrol pada Irigasi Tetes Berbasis IoT Menggunakan LoRa” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 20 Juli 2019

Mabarrus Sholeh
NRP. 0711 15 40000 015

.....*Halaman ini sengaja dikosongkan*.....

**SISTEM MONITORING DAN KONTROL
PADA IRIGASI TETES BERBASIS IOT
MENGUNAKAN LORA**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I



Dr. Muhammad Rivai, ST., MT.
NIP. 196904261994031003

Dosen Pembimbing II



Muhammad Aitaraqini B.Eng., M.Eng., Ph.D.
NIP. 1985201711039



.....*Halaman ini sengaja dikosongkan*.....

Sistem Monitoring dan Kontrol pada Irigasi Tetes Berbasis IoT Menggunakan LoRa

Nama : Mabarrus Sholeh
Pembimbing : Dr. Muhammad Rivai, ST.MT.
Muhammad Attamimi B.Eng., M.Eng., Ph.D.

ABSTRAK

Irigasi adalah salah satu instrumen penting dalam mendukung kualitas hasil pertanian. Fungsi dari irigasi ini adalah untuk menyuplai air yang memiliki pengaruh besar dalam unsur fisik tanah. Akan tetapi dalam aplikasinya, sistem irigasi yang ada memiliki tingkat efisiensi yang rendah dan tidak berdasarkan kebutuhan tanah. Berdasarkan kondisi tersebut, dibuat alat yang dapat melakukan pengontrolan dan monitoring kelembaban tanah. Alat ini terdiri dari dua bagian yaitu pengirim dan penerima. Bagian pengirim menggunakan mikrokontroler Arduino Nano untuk membaca 4 *capacitive soil moistures sensor* dan waktu dari RTC DS1307 yang hasilnya diproses menggunakan fuzzy logic untuk mengatur lama waktu pompa menyala. Data kelembaban yang telah terbaca dikirim menggunakan modul Transmitter LoRa SX1278 menuju Receiver LoRa SX1278 yang berada di rumah pemilik lahan. Bagian penerima ini terdiri dari Receiver LoRa SX1278 dengan pemrosesannya menggunakan Arduino Uno yang selanjutnya dihubungkan secara serial dengan NODEMCU ESP8266 untuk menampilkan data pada software Blynk menggunakan wifi. Alat ini telah dibuat yang terdiri dari 2 bagian yaitu bagian pengirim dan penerima. Hasil dari penelitian ini adalah sistem dapat menjaga kelembaban tanah dengan error 0 – 5 dari setpoint yang ditentukan yakni hasil pada kelembaban tanah kering 27 – 32 %, kelembaban tanah sedang 55 – 62 % dan kelembaban tanah basah 81 – 88 %. Sistem monitoring yang dibuat telah dapat memantau kelembaban tanah menggunakan aplikasi blynk.

Kata kunci: Irigasi, Kelembaban Tanah, LoRa

.....*Halaman ini sengaja dikosongkan*.....

Monitoring and Controlling System on Drip Irrigation Based on IoT with LoRa

Name : Mabarrus Sholeh
Supervisor : Dr. Muhammad Rivai, ST.MT.
Muhammad Attamimi B.Eng., M.Eng., Ph.D.

ABSTRACT

Irrigation is one of the important instruments in supporting the quality of agricultural products. Irrigation has function to supply water which has a large influence on the physical elements of the soil. In the application, existing irrigation systems have a low level of efficiency and are not based on land requirements. Based on these conditions, this projects are made that can control and monitor soil moisture. There are two parts in this projects, they are transmitter and receiver. Transmitter uses the Arduino Nano to read 4 capacitive soil moistures sensors and time from RTC DS1307. The results of reading data are processed using fuzzy logic which control the length of time the pump turning on. Data of moisture are sent using the LoRa SX1278 Transmitter module to the Receiver LoRa SX1278 which is in the home of the land owner. This receiver consists of a Receiver LoRa SX1278 with processing using Arduino Uno which is then connected serially with NODEMCU ESP8266 to display data in the Blynk software using wifi. The results of this study are that the system can maintain soil moisture with an error of 0 - 5%, namely dry soil moisture 27 - 32%, moderate soil moisture 55 - 62% and wet soil moisture 81 - 88% and applications made have been able to monitor soil moisture .

Keywords: Irrigation, Mositure Soil, LoRa

.....*Halaman ini sengaja dikosongkan*.....

KATA PENGANTAR

Puji dan syukur kepada Allah SWT atas berkat dan hikmat yang diberikan, penulis dapat menyelesaikan laporan tugas akhir dengan judul **“Sistem Monitoring dan Kontrol pada Irigasi Tetes Berbasis IoT Menggunakan LoRa”**, sebagai salah satu persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.

Dalam penyusunan dan penyelesaian laporan Tugas Akhir ini penulis mendapatkan banyak sekali doa, bantuan dan dukungan dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih yang sebesar – besarnya kepada:

1. Bapak Dr. Muhammad Rivai, ST.MT. dan Muhammad Attamimi B.Eng., M.Eng., Ph.D. selaku dosen pembimbing atas gagasan topik tugas akhir serta bimbingan dan arahan untuk penulis selama mengerjakan tugas akhir.
2. Bapak Ir. Tasripan, MT., Astria Nur Irfansyah, ST., M.Eng., Ir. Haris Pringadi, MT. dan Ir. Totok Mujiono, M.Kom. selaku dosen penguji tugas akhir Saya
3. Bapak Dr. Eng. Ardyono Priyadi, ST., M.Eng. selaku Kepala Departemen Teknik Elektro.
4. Seluruh dosen bidang studi elektronika Departemen Teknik Elektro ITS.
5. Orang tua, yang tak henti-hentinya memberikan semangat dan kasih sayang yang luar biasa kepada penulis.
6. Teman-teman bidang studi elektronika yang tidak dapat disebutkan satu-persatu yang telah membantu dan memberikan semangat kepada penulis selama menjalani perkuliahan di Departemen Teknik Elektro ITS.

Penulis menyadari bahwa masih banyak yang dapat dikembangkan pada tugas akhir ini. Oleh karena itu penulis menerima setiap masukan dan kritik yang diberikan. Semoga tugas akhir ini dapat memberikan manfaat bagi banyak pihak.

Surabaya, 20 Juli 2019

Mabarrus Sholeh

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR ISI

PERNYATAAN KEASLIAN.....	i
HALAMAN PENGESAHAN.....	iii
ABSTRAK.....	v
ABSTRACT.....	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan.....	4
1.7 Relevansi.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Irigasi Tetes.....	7
2.2 Kelembaban Tanah.....	8
2.3 Fuzzy Logic.....	9
2.4 Internet of Things.....	10
2.5 Sensor Capacitive Soil Moisture.....	11
2.6 LoRa SX 1278.....	12
2.7 Submersible Electric Pump 3 Volt.....	13
2.8 Buck Boost Converter DC-DC.....	13
2.9 Baterai LiPo (Lithium Polymer).....	14
2.10 Arduino.....	15
2.11 RTC DS1307.....	16
2.12 Blynk.....	17
2.13 Tinjauan Pustaka.....	18
2.13.1 Low-Cost Wireless Monitoring and Decision.....	18
2.13.2 Smart Irrigation based on Fuzzy Logic.....	18
BAB III PERANCANGAN SISTEM.....	19
3.1 Diagram Sistem.....	21
3.2 Flow Chart Sistem.....	25
3.3 Perancangan Perangkat Keras.....	26
3.3.1. Mikrokontroler.....	26

3.3.2.	Capacitive Soil Moisture Sensor.....	28
3.3.3.	RTC DS1307.....	29
3.3.4.	LCD I2C	29
3.3.5.	LoRa SX1278	30
3.3.6.	<i>Board</i> Mikrokontroler Terpadu	31
3.4	Perancangan Perangkat Lunak	33
3.4.1	Pembacaan Capacitive Soil Moisture Sensor.....	33
3.4.2	Pembacaan RTC DS1307	34
3.4.3	Pengiriman Data Menggunakan SX1278.....	35
3.5	Fuzzy Logic.....	36
3.5.1	Fuzifikasi	37
3.5.2	Pembentukan Rule	39
3.5.3	Defuzzifikasi.....	40
3.6	Perancangan Blynk.....	40
BAB IV	PENGUJIAN DAN ANALISIS	43
4.1.	Pengujian Karakteristik <i>Soil Moisture Sensor</i>	43
4.1.1.	Penentuan LSB dan MSB ADC Sensor	43
4.1.2.	Pengujian Output Tegangan Sensor	48
4.1.3.	Pengujian Operational Voltage Sensor.....	49
4.1.4.	Kalibrasi Kelembaban Tanah	49
4.1.4.1.	Kelembaban Tanah Kering	49
4.1.4.2.	Kelembaban Tanah Sedang.....	50
4.1.4.3.	Kelembaban Tanah Basah.....	51
4.2.	Pembacaan RTC	52
4.3.	Pengujian Pengiriman Data menggunakan SX1278.....	53
4.4.	Pengujian Blynk.....	55
4.5.	Respon Kelembaban Tanah Terhadap Air	56
4.6.	Pengujian Fuzzy Logic.....	58
4.6.1.	Kelembaban Tanah Kering.....	58
4.6.2.	Kelembaban Tanah Sedang.....	59
4.6.3.	Kelembaban Tanah Basah	59
4.7.	Pengujian Alat di Tempat Sesungguhnya	60
BAB V	PENUTUP	63
5.1.	Kesimpulan	63
5.2.	Saran.....	63
	DAFTAR PUSTAKA.....	65
	LAMPIRAN A Program Pengirim.....	69
	LAMPIRAN B Program Penerima	79
	LAMPIRAN C PROGRAM BLYNK.....	81

LAMPIRAN D DATASHEET SX1278	83
LAMPIRAN E <i>DATASHEET CAPACITIVE SOIL MOISTURE SENSOR</i>	87
LAMPIRAN F DATASHEET RTC DS1307	89
BIODATA PENULIS	93

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR GAMBAR

Gambar 2.1	Sistem Irigasi Tetes	8
Gambar 2.2	<i>Fuzzy Logic</i>	10
Gambar 2.3	<i>Internet of Things</i>	11
Gambar 2.4	Capacitive Soil Moisture Sensor	12
Gambar 2.5	Diagram Blok SX1278.....	13
Gambar 2.6	<i>Submersible Electric Pump 3V</i>	13
Gambar 2.7	Buck Boost Converter	14
Gambar 2.8	Battery LiPo	15
Gambar 2.9	Arduino	16
Gambar 2.10	Skema Diagram RTC DS1307	16
Gambar 2.11	Blynk.....	17
Gambar 3.1	Skema Sistem Keseluruhan.....	19
Gambar 3.2	Diagram Sistem Pengirim.	20
Gambar 3.3	Diagram Sistem Penerima.....	20
Gambar 3.4	Diagram Blok Sistem.....	22
Gambar 3.5	Arsitektur Sistem Pengirim.....	23
Gambar 3.6	Arsitektur Sistem Penerima	24
Gambar 3.7	Flow Chart Sistem	25
Gambar 3.8	Skematik Arduino Nano.....	27
Gambar 3.9	Skema Nodemcu NODEMCU ESP8266	27
Gambar 3.10	Antarmuka Arduino Uno dan Nodemcu ESP8266....	27
Gambar 3.11	Batas Capacitive Soil Moisture Sensor	28
Gambar 3.12	Antarmuka Capacitive Soil Moisture Sensor	29
Gambar 3.13	Antarmuka RTC DS1307.....	29
Gambar 3.14	Antarmuka I2C dengan Arduino	30
Gambar 3.15	Antarmuka SX1278	30
Gambar 3.16	Desain PCB Pengirim	31
Gambar 3.17	Layout Blok Pengirim.....	32
Gambar 3.18	Layout Blok Penerima	32
Gambar 3.19	Flowchart Pembacaan Sensor	33
Gambar 3.20	Flowchart Pembacaan RTC.....	34
Gambar 3.21	Flowchart Modul Pengirim SX1278.....	35
Gambar 3.22	Flowchart Modul Penerima SX1278.....	36
Gambar 3.23	Flowchart Fuzzy Logic	37
Gambar 3.24	Fuzifikasi Kelembaban Tanah Kering.....	38
Gambar 3.25	Fuzifikasi Kelembaban Tanah Sedang	38
Gambar 3.26	Fuzifikasi Kelembaban Tanah Basah	39

Gambar 3.27	Fuzifikasi Output.....	39
Gambar 3.28	Flowchart Penampilan Blynk	41
Gambar 4.1	Pengujian Kelembaban Tanah pada Udara	43
Gambar 4.3	Pengukuran Sensor Kelembaban Tanah B pada Udara.....	44
Gambar 4.4	Pengukuran Sensor Kelembaban Tanah C pada Udara.....	45
Gambar 4.5	Pengukuran Sensor Kelembaban Tanah D pada Udara	45
Gambar 4.6.	Pengujian Kelembaban Tanah pada Air.....	46
Gambar 4.7	Pengukuran Sensor Kelembaban Tanah A pada Air	46
Gambar 4.8	Pengukuran Sensor Kelembaban Tanah B pada Air	47
Gambar 4.9	Pengukuran Sensor Kelembaban Tanah C pada Air	47
Gambar 4.10	Pengukuran Sensor Kelembawban Tanah D pada Air	48
Gambar 4.11	Kalibrasi Kelembaban Tanah Kering.....	50
Gambar 4.12	Kalibrasi Kelembaban Tanah Sedang.....	51
Gambar 4.13	Kalibrasi Kelembaban Tanah Basah.....	51
Gambar 4.14	Pengujian RTC	52
Gambar 4.15	Pembacaan RTC	53
Gambar 4.16	Pengujian SX1278.....	54
Gambar 4.17	Pengiriman Data SX1278.....	54
Gambar 4.18	Penerimaan Pengiriman Data SX1278	55
Gambar 4.20	Tampilan Blynk	56
Gambar 4.18	Pengujian Fuzzy Logic.....	58
Gambar 4.19	Pengujian Alat di Tempat Sesungguhnya	60
Gambar 4.20	Hasil Fuzzy Logic di Lokasi Sesungguhnya	61

DAFTAR TABEL

Tabel 2.1	Kelembaban Tanah Tiap Tanaman.....	9
Tabel 3.1	Spesifikasi Alat	21
Tabel 4.1.	Pengujian Output Tegangan Sensor.....	49
Tabel 4.2	Pengujian <i>Operational Voltage</i> Sensor.....	49
Tabel 4.3	Tingkat Kelembaban Tanah Kering.....	50
Tabel 4.4	Tingkat Kelembaban Tanah Sedang	50
Tabel 4.5	Tingkat Kelembaban Tanah Basah	51
Tabel 4.6	Respon Kelembaban Tanah terhadap Air	57
Tabel 4.7	Pengujian Fuzzy Logic pada Kelembaban Tanah Kering..	59
Tabel 4.8	Pengujian Fuzzy Logic pada Kelembaban Tanah Sedang .	59
Tabel 4.9	Pengujian Fuzzy Logic pada Kelembaban Tanah Basah ...	60

.....*Halaman ini sengaja dikosongkan*.....

BAB I

PENDAHULUAN

Tugas akhir merupakan suatu penelitian yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Topik yang akan dibahas pada tugas akhir ini mengenai monitoring dan kontrol pada irigasi tetes berbasis IoT menggunakan LoRa.

Pada bab ini membahas mengenai hal-hal yang mendahului pelaksanaan tugas akhir. Hal tersebut meliputi latar belakang, perumusan masalah, tujuan penulisan, batasan masalah, metodologi penelitian, sistematika penulisan, dan relevansi.

1.1 Latar Belakang

Air adalah salah satu komponen penting pada tumbuhan yang mempengaruhi pertumbuhan dan perkembangan. Kebutuhan air tiap tanaman berbeda satu sama lain tergantung masa pertumbuhan dan perkembangan yang dialami. Pada musim kemarau, tumbuhan banyak yang mengalami water stress karena kekurangan dan laju evapotranspirasi yang melebihi laju absorpsi air. Sedangkan pada musim hujan, tanaman mengalami kondisi jenuh air [1].

Untuk mengendalikan jumlah air yang ada, petani menggunakan irigasi. Irigasi yang digunakan pada umumnya adalah irigasi konvensional yang mengandalkan gravitasi sehingga tempat yang paling tinggi akan mendapatkan air terlebih dahulu. Hal ini mengakibatkan persebaran air tidak merata sehingga pertumbuhan dan perkembangan tumbuhan akan terganggu. Terganggunya pertumbuhan dan perkembangan akan mempengaruhi hasil produktivitas dari petani [2].

Sistem irigasi konvensional yang dikembangkan sekarang hanya akan mengeluarkan air pada musim tertentu bukan berdasarkan kebutuhan tanah saat itu. Oleh karena itu, kadar air tidak terkontrol sehingga suplai unsur hara akan berkurang. Keadaan seperti ini akan merugikan petani yang berimbas pada penurunan hasil produksi [3].

Selain itu, sistem pertanian yang dikembangkan sekarang belum dapat mengetahui tingkat kelembaban tanah secara real time. Bahkan cenderung tidak memperhatikan dan hanya berdasarkan intuisi dari petani. Hal ini terjadi dikarenakan alat yang tidak tersedia di kalangan para petani. Kondisi seperti ini secara tidak langsung menyebabkan

penurunan tingkat kesuburan dalam bertahap. Pengaruh yang diakibatkan memang tidak akan terasa secara langsung, tetapi dalam waktu yang lama [4].

Berdasarkan permasalahan yang ada, maka perlu adanya sebuah sistem irigasi yang inovatif. Pada penelitian ini akan dibuat sistem monitoring dan kontrol pada irigasi tetes berbasis IoT dengan menggunakan LoRa. Sistem yang ditawarkan ini akan dapat mengetahui kadar kelembaban tanah dari jarak jauh. Selain itu, dari parameter kelembaban tanah akan digunakan untuk mengatur aliran irigasi sehingga dapat menyesuaikan kebutuhan tanah. Jenis irigasi yang digunakan adalah irigasi tetes bertujuan untuk maksimalisasi penyerapan air ke dalam akar. Oleh karena itu, penelitian ini diharapkan dapat mengatasi permasalahan yang telah dijelaskan di atas.

1.2 Perumusan Masalah

Berdasar latar belakang diatas, dapat dirumuskan masalah sebagai berikut:

1. Bagaimana cara monitoring kelembaban tanah dari jarak jauh?
2. Bagaimana cara mengontrol air yang mengalir sesuai kebutuhan tanaman?
3. Bagaimana cara monitoring kelembaban tanah menggunakan smartphone?

1.3 Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut:

1. Mengirim data kelembaban tanah dari transceiver ke receiver menggunakan LoRa.
2. Mengontrol aliran air yang mengalir pada irigasi menggunakan fuzzy logic.
3. Menggunakan aplikasi Blynk yang memuat informasi data kelembaban tanah sehingga bisa dilihat dalam Smartphone.

1.4 Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Komunikasinya hanya satu arah.
2. Parameter yang digunakan hanya kelembaban tanah dan fase tanam.
3. Analisis yang dilakukan tidak mempertimbangkan aspek eksternal yang mungkin ditimbulkan seperti hujan.

4. Pembahasan yang dilakukan hanya melibatkan satu node sensor saja.

1.5 Metodologi Penelitian

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

1. Studi literatur

Studi literatur berisi kegiatan pengumpulan dan pengkajian dasar teori yang terpercaya untuk menunjang penulisan tugas akhir ini. Literatur dapat bersumber dari paper, jurnal, artikel, buku, maupun website yang bertaraf nasional dan internasional, serta dari hasil konsultasi dengan dosen pembimbing.

2. Observasi dan Analisa Masalah

Pada tahap ini dilakukan pengkajian terhadap sistem irigasi dan alat yang akan dibuat. Analisa terhadap kekurangan-kekurangan sensor kelembaban tanah guna menyempurnakan alat tersebut. Observasi dan analisa masalah dilakukan dengan mengkaji paper, jurnal, dan isu-isu terkini di website terkait efisiensi sistem irigasi. Salah satu masalah yang ditemukan saat pengkajian berbagai sumber adalah respon dari perubahan kelembaban tanah yang lambat. Selanjutnya melakukan inovasi yang implementatif untuk mengatasi masalah tersebut.

3. Persiapan Alat dan Bahan

Tahap ini merupakan tahap pencarian informasi mengenai konsep yang dibutuhkan untuk merancang alat ini, yang bisa didapatkan dari studi literatur dan bimbingan dosen pembimbing. Kemudian dilakukan pengumpulan alat dan bahan yang dibutuhkan. Alat dan bahan yang diperlukan adalah seperangkat sensor kelembaban tanah, modul transceiver LoRa, Arduino, Nodemcu, LCD Display, dan perangkat lainnya.

4. Perancangan Alat

Perancangan bertujuan untuk mendapatkan desain dan mekanisme yang optimal dengan memperhatikan data-data yang diperoleh sebelumnya. Perancangan disini termasuk perancangan software dan hardware. Software dirancang dengan pembuatan

source code yang meliputi membaca sensor kelembaba tanah, fuzzy logic, pengiriman data, penerimaan data, dan pembuatan webserver.

5. Pembuatan Alat

Perancangan bertujuan untuk mendapatkan desain dan mekanisme yang optimal dengan memperhatikan data-data yang diperoleh sebelumnya. Perancangan disini termasuk perancangan software dan hardware. Software dirancang dengan pembuatan source code yang meliputi membaca sensor kelembaba tanah, fuzzy logic, pengiriman data, penerimaan data, dan pembuatan webserver. Desain hardware meliputi arsitektur elektronik, interface dan desain mekanik penunjang lainnya.

6. Tahap Pengujian

Pengujian dilakukan secara bertahap, dari pengujian software terhadap masing-masing komponen hingga pengujian hardware yang sudah tertanam software yang telah dibuat. Bagian-bagian tersebut meliputi sistem pembacaan sensor kelembaban tanah, fuzzy logic dan komunikasi LoRa. Tahap pengujian selanjutnya yaitu memasang alat pada prototype irigasi yang telah didesain.

7. Analisa dan Evaluasi

Analisa dilakukan terhadap hasil pengujian sehingga karakteristik software dan hardware dapat diketahui. Analisa dilakukan pada respon pembacaan sensor, pengiriman data, fuzzy logic, dan pembuatan webserver. Apabila karakteristik dari dua bagian utama tersebut belum sesuai, maka perlu dilakukan evaluasi pada sistem untuk dirancang dan diuji kembali.

8. Penyusunan Laporan Tugas Akhir

Tahap penyusunan laporan merupakan tahap terakhir dari proses pengerjaan tugas akhir ini. Laporan berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, studi literatur, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan analisa, serta penutup.

1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai

berikut:

- **BAB I: Pendahuluan**

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

- **BAB II: Tinjauan Pustaka**

Pada bab ini berisi mengenai teori yang mendasari penyusunan laporan tugas akhir secara umum khususnya yang berhubungan komponen yang akan digunakan.

- **BAB III: Perancangan Sistem**

Bab ini menjelaskan tentang perencanaan sistem yang meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) untuk pembuatan alat ini.

- **BAB IV: Pengujian dan Analisis**

Pada bab ini menguraikan tentang pengujian alat pada pesawat dan analisa hasil pengujian.

- **BAB V: Penutup**

Bab ini berisi tentang kesimpulan yang diperoleh dari pembuatan alat serta saran untuk pengembangan lebih lanjut.

1.7 Relevansi

Dengan dibuatnya tugas akhir ini akan membantu proses irigasi pada suatu wilayah pertanian. Sistem irigasi yang dibuat ini akan memantau tingkat kelembaban tanah sesuai kebutuhan tanaman dan akan dikontrol menggunakan fuzzy logic. Selain itu, data kelembaban tanah yang telah diambil dapat dipantau dari lokasi yang jauh sehingga petani dapat memantaunya setiap saat tanpa langsung ke lokasi pertanian.

.....*Halaman ini sengaja dikosongkan*.....

BAB II

TINJAUAN PUSTAKA

Suatu penelitian memerlukan teori-teori yang sudah ada sebelumnya untuk dikaji lebih dalam memperkuat argumen penulis. Teori tersebut digunakan untuk membantu penulis dan sebagai dasar dalam membuat suatu penelitian.

Pada bab ini terdapat teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian ini terdapat tinjauan pustaka tentang komponen yang akan digunakan untuk membuat alat pada penelitian ini.

2.1 Irigasi Tetes

Irigasi tetes adalah salah satu jenis microirrigation atau trickle irrigation yang merupakan kelompok irigasi dengan efisiensi tinggi. Skema sistem yang digunakan berupa pipa kecil yang diletakkan di atas tanah sepanjang tanaman. Setiap sisi yang berdekatan dengan tanaman akan diberi lubang sebagai tempat keluarnya air. Dengan susunan seperti itu, sistem ini dapat mengalirkan air langsung ke daerah akar sehingga memiliki tingkat efisiensi 90 % - 95 % dalam mengalirkan air [5].

Irigasi Tetes (drip irrigation) mulai berkembang akhir akhir ini seiring berkembangnya penggunaan pipa plastic pada tahun 1960-an. Irigasi tetes dibedakan menjadi dua yaitu irigasi tetes menggunakan pompa dan irigasi tetes dengan gaya gravitasi. Irigasi tetes adalah metode yang efisien untuk menyediakan air irigasi langsung ke tanah di zona akar tanaman dan dengan demikian, meminimalkan kerugian konvensional seperti perkolasi dalam, limpasan dan erosi tanah. Tidak seperti irigasi permukaan, irigasi tetes lebih cocok dan ekonomis jika digunakan di daerah yang langka air yang memiliki topografi yang bergelombang, tanah yang dangkal dan berpasir dan untuk tanaman bernilai tinggi yang tersebar luas. Manajemen kandungan N melalui fertigasi dengan irigasi tetes dapat mengurangi pupuk dan meminimalkan merugikan dampak lingkungan [6]–[8]. Salah satu contoh dari irigasi tetes bisa dilihat pada gambar 2.1.



Gambar 2.1 Sistem Irigasi Tetes

2.2 Kelembaban Tanah

Kelembaban tanah dipengaruhi oleh besarnya kandungan air dalam tanah. Kelembaban tanah merupakan salah satu parameter yang digunakan sebagai indikasi tingkat kekeringan. Menurut Asdak, kelembaban tanah umumnya terbentuk melalui tiga proses:

- a. Kelembaban higroskopis adalah kelembaban yang terjadi karena air terikat pada lapisan tipis butir-butir tanah. Air terikat ini tidak dapat bergerak dan oleh karenanya tidak dapat dimanfaatkan oleh tanaman.
- b. Kelembaban kapiler adalah kelembaban tanah yang terjadi oleh adanya gaya tarik-menarik antara butir-butir tanah. Air yang dihasilkan dapat dimanfaatkan oleh tanaman.
- c. Kelembaban gravitasi adalah kelembaban yang terjadi sebagai akibat adanya gaya tarik bumi, yaitu air dalam posisi peralihan menuju ke pori-pori tanah yang lebih besar.

Masing masing tanaman memiliki tingkat kelembaban ideal yang berbeda. Ada tanaman yang mulai dari masa tanam memiliki tingkat kelembaban tanah yang sama. Ada juga tanaman yang memiliki tingkat kelembaban tanah sesuai masa tanamnya misalkan tanaman padi. Tanaman padi mempunyai 4 fase tanaman dengan tingkat kelembaban tanah yang berbeda. Kelembaban tanah pada level basah untuk fase awal dan vegetatif sangat penting untuk tanaman khususnya daerah perakaran dalam menyediakan air yang cukup untuk pertumbuhan akar, batang dan daun. Kemudian, air irigasi dapat dikurangi untuk menjaga kelembaban

tanah pada level agak basah di fase tengah musim untuk menghindari dan mengurangi jumlah bulir yang tidak produktif. Pada fase pertumbuhan akhir musim, kelembaban tanah pada level kering dapat diterapkan untuk menghemat air irigasi ketika pada fase ini kebutuhan air tanaman minimal. Kelembaban tanah adalah sumber air untuk atmosfer melalui proses yang mengarah ke evapotranspirasi dari darat, yang meliputi terutama transpirasi tanaman dan penguapan tanah kosong [9]–[11]. Pada tabel 2.1 disajikan data kelembaban tanah ideal tiap tanaman

Tabel 2.1 Kelembaban Tanah Tiap Tanaman

(Sumber : [https://www.javamas.com/kelembaban-tanah-ideal/\[30\]](https://www.javamas.com/kelembaban-tanah-ideal/[30]))

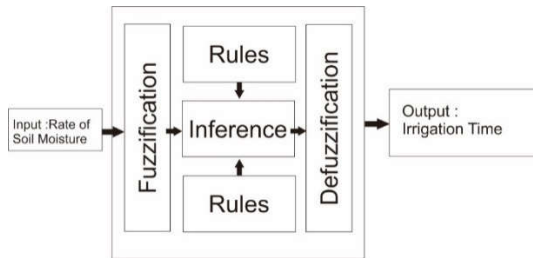
Jenis Tanaman	Kelembaban Tanah Ideal
Cabai	60 – 80 %
Terong	40 – 60 %
Jagung	30 – 50 %
Melon	50 – 70 %
Semangka	50 – 70 %
Kubis	40 – 60 %
Kacang Tanah	30 – 50 %

2.3 Fuzzy Logic

Teori fuzzy logic pertama kali diperkenalkan oleh Lotfi Zadeh pada tahun 1965. Fuzzy logic adalah Teknik control yang bagus sebagai metodologi dalam pemecahan beberapa jenis sistem control. Sistem ini merupakan metode komputasi untuk memodelkan dan memproses informasi yang tidak pasti berdasarkan penggabungan fakta numerik dan pengalaman manusia. Teori kontrol logika fuzzy terinspirasi oleh kemampuan manusia yang bertindak berdasarkan informasi yang diperoleh dari mereka pengalaman. Logika fuzzy bekerja berdasarkan aturan linguistik dari penalaran dan pengambilan keputusan. Kontrol logika fuzzy memiliki beberapa tahap yang fuzzification, inferensi, dan defuzzifikasi [12], [13].

Fuzzyfication terdiri dari proses transformasi nilai-nilai ke dalam tingkatan keanggotaan untuk istilah-istilah linguistik input set fuzzy. Fungsi keanggotaan digunakan untuk mengaitkan nilai untuk setiap istilah linguistik. Inferensi fuzzy adalah bagian inti dari fuzzy sistem yang menggabungkan fakta - fakta yang diperoleh dari fuzzyfication dengan

basis aturan dan melakukan fuzzy proses penalaran. Aturan fuzzy biasanya adalah pernyataan if-then yang menggambarkan tindakan yang harus diambil sebagai tanggapan terhadap berbagai hal masukan fuzzy. Dalam defuzzification, semua nilai output fuzzy berubah fungsi keanggotaan output masing-masing. Salah satu yang Teknik paling banyak defuzzifikasi yang umum digunakan disebut pusat gravitasi (COG) atau metode centroid. Dalam metode ini, masing-masing fungsi keanggotaan keluaran di atas nilai yang ditunjukkan oleh masing-masing output fuzzy dipotong dan digabungkan kemudian COG dihitung [14]. Arsitektur *Fuzzy Logic* bisa dilihat pada gambar 2.2.



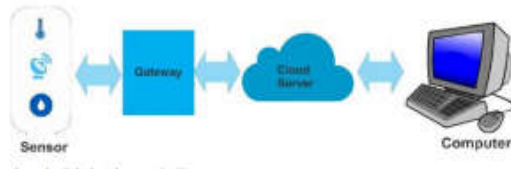
Gambar 2.2 *Fuzzy Logic*.

2.4 Internet of Things

Secara umum Internet of Things (IoT) adalah suatu konsep menghubungkan perangkat pintar dengan perangkat pintar yang lain baik digunakan untuk bertukar data maupun untuk keperluan pengendalian melalui internet. Saat ini konsep IoT sudah banyak digunakan pada objek di sekitar kita. Diperkirakan pada tahun 2020 dalam 50 miliar objek yang terhubung dengan Internet. Pada IoT setiap benda harus memiliki sebuah IP address. IP address adalah sebuah identitas dalam jaringan yang membuat benda tersebut bisa diperintahkan dari peralatan lain dalam jaringan yang sama. IP address dalam setiap peralatan tersebut akan dikoneksikan ke jaringan internet. IOT membawa suatu paradigma pergeseran dalam layanan, infrastruktur, dan industri konsumen. Sementara pergeseran paradigma ini terjadi, kepercayaan dan keamanan adalah persyaratan yang diperlukan untuk mengatasi berbagai jenis serangan, ancaman, malfungsi, dan dampak buruk bagi masyarakat [15]–[17].

Karakteristik dari Internet of Things (IoT) adalah bahwa perangkat keduanya terhubung jaringan dan tertanam di lingkungan. Tidak seperti

teknologi server jaringan tradisional yang memiliki sedikit koneksi dengan lingkungan sehingga perangkat IoT dirancang dan diposisikan untuk merespons dan mempengaruhi lingkungan mereka. Namun, jika pengguna mengakses perangkat IoT untuk mengukur suhu ruangan, sensor dan perangkat harus ditempatkan di ruangan tertentu[18]. Skema sederhana dari IoT bisa dilihat pada gambar 2.3.



Gambar 2.3 *Internet of Things.*

2.5 Sensor Capacitive Soil Moisture

Sensor yang digunakan pada Tugas Akhir ini adalah Capacitive Soil Moisture Sensor. Jenis sensor ini mengukur kelembaban tanah menggunakan prinsip sensing kapasitif. Bahan yang digunakan dalam pembuatannya merupakan bahan yang tidak mudah terkena korosi sehingga memberikan performa yang prima. Modul sensor ini mencakup regulator sehingga memberikan tegangan operasi antara 3,3 sampai 5 V [19].

Pengukuran kapasitif memiliki beberapa keunggulan, tidak hanya menghindari korosi pada probe tetapi juga memberikan pembacaan yang lebih baik dari kadar air tanah dibandingkan dengan pengukuran resistensi. Pengukuran resistansi ini tidak mengukur kelembaban (karena air adalah penghantar arus yang buruk), melainkan mengukur ion yang terlarut dalam uap air. Ketika ditambahkan pupuk akan mengurangi ketahanan tanah, walaupun tidak ada air yang ditambahkan. Pengukuran kapasitif pada dasarnya mengukur dielektrik yang dibentuk oleh tanah dan air[20]. Gambar 2.4 menunjukkan sensor kelembaban tanah yang digunakan.



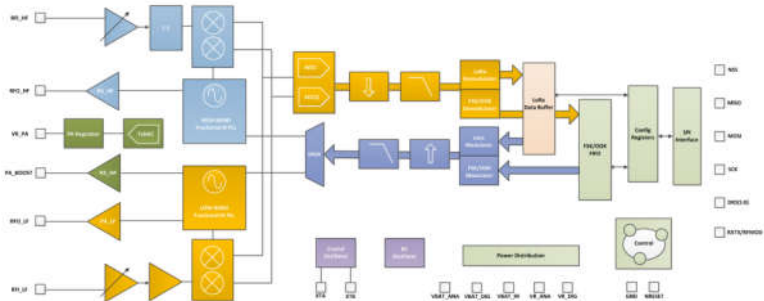
Gambar 2.4 Capacitive Soil Moisture Sensor

2.6 LoRa SX 1278

Istilah LoRa adalah singkatan dari Long Range berupa teknologi frekuensi Radio nirkabel yang diperkenalkan oleh perusahaan bernama Semtech. Teknologi LoRa ini dapat digunakan untuk mengirimkan informasi dua arah ke jarak jauh tanpa menghabiskan banyak daya. Alat ini dapat digunakan untuk mengirim data pengukuran sensor dengan konsumsi daya kecil. Konsumsi daya yang kecil ini berdampak pada besarnya bandwidth yang kecil sehingga data yang dikirimkan tidak memiliki ukuran besar. Oleh karena itu, teknologi ini tidak bisa mengirim audio, video, ataupun data besar lainnya[21].

LoRa yang digunakan pada penelitian ini adalah *Transceiver LoRa SX1278*. *Transceiver LoRa SX1278* adalah sebuah modul pengiriman yang menyediakan komunikasi persebaran spektrum ultra panjang dan ketebalan interferensi sehingga meminimalkan konsumsi arus. Teknik modulasi yang digunakan menghasilkan sensitivitas lebih dari -148 dBm dengan material kristal yang memiliki harga murah. Modul pengiriman dan penerimaan ini menggunakan tegangan sumber 1.8 Volt sampai 3.7 Volt dan suhu operasi antara -40° C sampai 85° C[22].

SX1278 adalah transceiver setengah dupleks dan IF rendah. Sinyal RF yang diterima pertama kali akan diperkuat oleh LNA. Berdasarkan input LNA, konversi ke diferensial dibuat untuk meningkatkan linearitas orde kedua dan penolakan harmonik. Sinyal kemudian dikonversi ke komponen inphase dan quadrature (I&Q) pada frekuensi menengah (IF) oleh tahap mixer. Sepasang ADC sigma delta kemudian melakukan konversi data, dengan semua pemrosesan sinyal berikutnya dan demodulasi dilakukan dalam domain digital [23]. Arsitektur penyusun SX1278 dapat dilihat pada gambar 2.5.



Gambar 2.5 Diagram Blok SX1278

2.7 Submersible Electric Pump 3 Volt

Komponen ini merupakan motor pompa celup ukuran kecil dengan biaya rendah dan dapat dioperasikan dari catu daya 2,5 ~ 6V. Pompa ini bisa memakan waktu hingga 120 liter per jam dengan konsumsi arus yang sangat rendah yaitu 220mA. Cara pemakaian cukup menyambungkan pipa tabung ke outlet motor, rendam pompa dalam air dan nyalakan dengan sumber tegangan sesuai karakteristiknya. Ketinggian air harus dikondisikan selalu lebih tinggi dari motor. Kondisi yang terlalu kering dapat merusak motor karena pemanasan dan juga akan menghasilkan suara[24]. Gambar 2.6 menunjukkan pompa yang digunakan pada pembuatan alat ini.

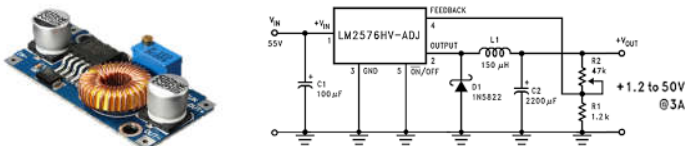


Gambar 2.6 Submersible Electric Pump 3V

2.8 Buck Boost Converter DC-DC

Tegangan masukan dari Arduino Nano adalah maksimal 5.5 Volt. Sedangkan sumber tegangan untuk menyuplai adalah 9 Volt. Oleh karena itu, diperlukan sebuah bagian untuk menyesuaikan tegangan

supply dengan tegangan input mikrokontroler. Buck boost converter digunakan untuk menyelesaikan permasalahan ini. Modul ini menerapkan sistem Switching Mode Power Supply. Sistem ini mengubah daya listrik secara efisien dengan menggabungkan pengaturan switching. Buck boost yang digunakan adalah IC XL4005 dengan tegangan masukan antara 5 volt dan 30 volt. Tegangan keluaran yang dihasilkan yaitu 0.8 volt sampai 30 volt. Penyesuaian tegangan tersebut dilakukan dengan mengatur multituran pada modul tersebut. Pada gambar 2.7 ditunjukkan alat dan skema rangkaian Buck Boost yang digunakan.



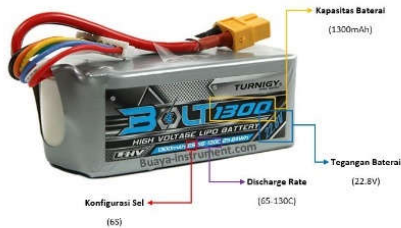
Gambar 2.7 Buck Boost Converter

2.9 Baterai LiPo (Lithium Polymer)

Baterai LiPo seperti pada gambar.10 digunakan sebagai sumber dari rangkaian yang sudah dibuat seperti Arduino dan sensor-sensor lainnya. Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastic film tipis. Lapisan ini tersusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Metode ini dapat membuat baterai LiPo dibentuk dengan berbagai macam ukuran. Terdapat kekurangan dari baterai LiPo ini yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada charging dan discharging rate. Dapat diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini tidak dapat diaplikasikan pada keadaan sehari-hari karena dapat menimbulkan bahaya[25].

Ada tiga kelebihan dari baterai LiPo daripada baterai jenis lain :

- Baterai LiPo memiliki berat lebih ringan dan tersedia dalam berbagai bentuk atau ukuran.
 - Baterai LiPo memiliki kapasitas penyimpanan lebih besar.
 - Baterai LiPo memiliki tingkat discharge rate energy yang tinggi.
- Pada gambar 2.8 ditunjukkan bagian dari baterai LiPo.



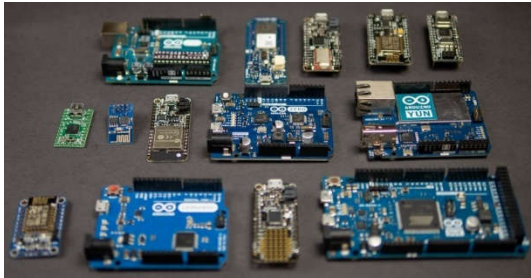
Gambar 2.8 Battery LiPo

Selain keuntungan yang disebutkan diatas, baterai ini memiliki beberapa kekurangan antara lain :

- Harga dari baterai LiPo relatif lebih mahal.
- Memiliki umur yang lebih pendek dari baterai lain.
- Dari segi keamanan baterai LiPo kurang begitu aman karena menggunakan bahan elektrolit yang mudah terbakar

2.10 Arduino

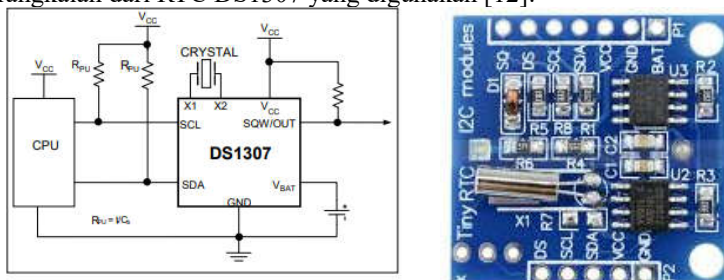
Arduino merupakan pengendali mikro *single-board* yang bersifat *open-source* yang didalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan ATMEL dan menggunakan *software* dan bahasa sendiri. Chip tersebut dapat diprogram sesuai kebutuhan menggunakan komputer dengan Bahasa C. Karena Arduino bersifat *open source* pengguna dapat mengembangkannya secara mandiri sesuai dengan kebutuhan. Hal ini membuat arduino dapat berkembang pesat karena sifatnya yang *open source*. Pada Arduino terdapat produk tambahan seperti *X-bee shield*, *Ethernet shield*, *microSD shield* dan produk lainnya yang dapat dipasang dan digunakan dengan mudah pada Arduino. Terdapat beberapa jenis Arduino antara lain Arduino PRO MINI, Arduino NANO, Arduino UNO, Arduino MEGA dan produk Arduino lainnya[26]. Pada gambar 2.9 ditampilkan berbagai jenis Arduino.



Gambar 2.9 Arduino

2.11 RTC DS1307

Seri real-time clock (RTC) DS1307 adalah jam / kalender desimal (BCD) berkekuatan biner penuh. ditambah 56 byte NV SRAM. Alamat dan datanya ditransfer secara serial melalui I2 C, bis dua arah. Jam / kalender menyediakan detik, menit, jam, informasi hari, tanggal, bulan, dan tahun. Akhir dari tanggal bulan secara otomatis disesuaikan dengan bulan dengan kurang dari 31 hari, termasuk koreksi untuk lompatan tahun. Jam beroperasi dalam 24-jam atau 12- format jam dengan indikator AM / PM. DS1307 memiliki sebuah built-in power-sense circuit yang mendeteksi kegagalan daya dan secara otomatis beralih ke persediaan cadangan. Operasi ketepatan waktu berlanjut sementara bagian beroperasi dari persediaan cadangan. Pada gambar 2.10 disajikan komponen dan rangkaian dari RTC DS1307 yang digunakan [12].



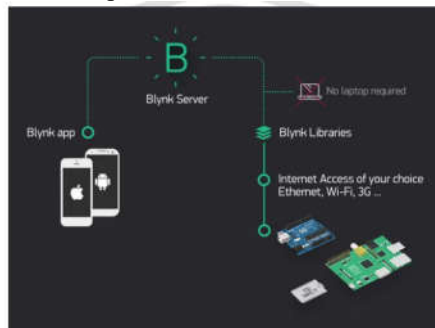
Gambar 2.10 Skema Diagram RTC DS1307

2.12 Blynk

Blynk adalah platform aplikasi iOS dan Android untuk mengontrol Arduino, Raspberry Pi dan sejenisnya melalui Internet. Aplikasi ini adalah fitur digital yang dapat membangun antarmuka grafis untuk sebuah proyek dengan mudah. Keunggulan dari aplikasi ini adalah sangat sederhana. Blynk tidak terikat pada sebuah device tertentu. Sebaliknya, aplikasi ini mendukung perangkat keras sesuai pilihan baik Arduino atau Raspberry Pi yang dapat terhubung ke Internet melalui Wi-Fi, Ethernet atau Chip ESP8266[27]. Pada gambar 2.11 disajikan skema kerja dari Blynk.

Blynk dirancang untuk *Internet of Things* yang memungkinkan dapat mengontrol perangkat keras dari jarak jauh, dapat menampilkan data sensor, menyimpan data, dan memvisualisasikannya. Ada tiga komponen utama dalam platform[28]:

- Blynk App: - Memungkinkan Anda untuk membuat antarmuka untuk setiap proyek menggunakan berbagai widget yang telah disediakan.
- Blynk Server: - perangkat yang bertanggung jawab untuk semua komunikasi antara smartphone dan perangkat keras. Pengguna dapat menggunakan Blynk Cloud atau menjalankan server Blynk pribadi secara lokal.
- Blynk Libraries: - Memungkinkan komunikasi, untuk semua platform perangkat keras yang populer dengan server dan memproses semua perintah masuk dan keluar.



Gambar 2.11 Blynk

2.13 Tinjauan Pustaka

Tinjauan pustaka bertujuan untuk membandingkan perangkat yang telah ada dan dikembangkan sebelumnya dengan perangkat yang dirancang pada tugas akhir ini. Berikut merupakan judul paper atau proyek yang dibandingkan dengan proyek pada tugas akhir ini.

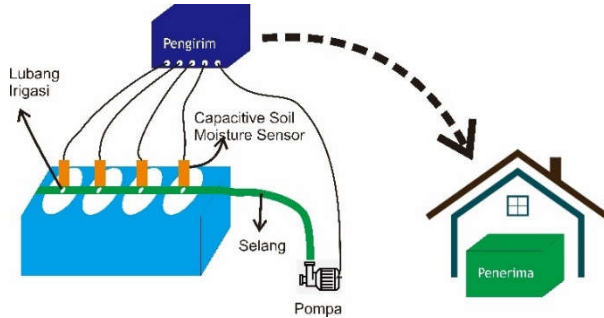
2.13.1 Low-Cost Wireless Monitoring and Decision Support for Water Saving in Agriculture

Sistem ini merupakan sebuah sistem irigasi otomatis yang dapat mengontrol kelembaban tanah. Pengaturan kelembaban tanah yang digunakan adalah dengan metode *fuzzy logic* dan *wireless sensor node*. Sensor yang digunakan pada penelitian ini adalah sensor kelembaban tanah dan temperatur. Sensor tersebut ditempatkan pada beberapa titik agar bias mencakup area yang luas. Tiap penempatan juga terdapat mikrokontroler untuk mengolah data dari sensor. Di bagian lain, ada sebuah mikrokontroler yang berfungsi untuk mendapatkan data dari tiap tiap node sensor yang dikirimkan. Data yang telah diperoleh akan digunakan untuk mengatur kelembaban tanah menggunakan *fuzzy logic*. Keluaran dari *fuzzy logic* berupa lama waktu pompa menyala[29].

2.13.2 Smart Irrigation Decision Support based on Fuzzy Logic using Wireless Sensor Network

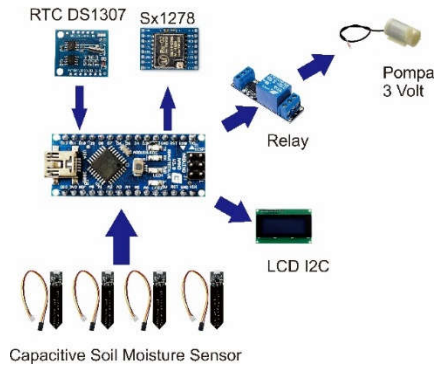
Penelitian ini juga memiliki metode dan sistem yang sama dengan tinjauan pustaka pertama. Sistem ini merupakan sebuah sistem irigasi otomatis yang dapat mengontrol kelembaban tanah. Pengaturan kelembaban tanah yang digunakan adalah dengan metode *fuzzy logic* dan *wireless sensor node*. Sensor yang digunakan pada penelitian ini adalah sensor kelembaban tanah dan temperatur. Sensor tersebut ditempatkan pada beberapa titik agar bias mencakup area yang luas. Tiap penempatan juga terdapat mikrokontroler untuk mengolah data dari sensor. Di bagian lain, ada sebuah mikrokontroler yang berfungsi untuk mendapatkan data dari tiap tiap node sensor yang dikirimkan. Data yang telah diperoleh akan digunakan untuk mengatur kelembaban tanah menggunakan *fuzzy logic*. Keluaran dari *fuzzy logic* berupa lama waktu pompa menyala[13].

BAB III PERANCANGAN SISTEM



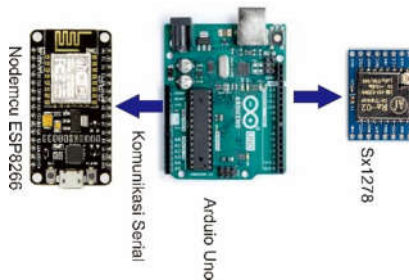
Gambar 3.1 Skema Sistem Keseluruhan

Pada bab ini dijelaskan tentang gambaran keseluruhan alat yang akan dibuat. Alat yang akan dibuat terdiri dari 2 blok yaitu pengirim dan penerima dengan visualisasi seperti yang ada di gambar 3.1. Blok pengirim yang ditunjukkan pada gambar 3.2 terdiri dari beberapa bagian. Bagian tersebut yaitu *capacitive soil moisture sensor*, RTC DS1307, Mikrokontroler Arduino Nano, Relay, *Submersible electric pump 3 Volt*, dan modul Transceiver SX1278. *Capacitive soil moisture sensor* untuk mengukur kelembaban tanah. Mikrokontroler Arduino Nano untuk melakukan proses data. RTC DS1307 digunakan untuk mengetahui masa tanam dari suatu tanaman. Relay untuk mengontrol lama waktunya pompa sesuai hasil pemrosesan Mikrokontroler Arduino Nano. *Submersible electric pump* sebagai aktuator mengalirkan air. Modul transceiver SX1278 sebagai transmitter digunakan untuk mengirimkan data yang dibaca. Untuk mengontrol lamanya *Submersible Electric Pump* menyala digunakan metode logika fuzzy yang diproses dalam Mikrokontroler Arduino Nano.



Gambar 3.2 Diagram Sistem Pengirim.

Sedangkan dalam blok penerima pada gambar 3.3 terdiri dari Transceiver SX1278, Mikrokontroler Arduino Uno, dan NODEMCU ESP8266. Transceiver SX1278 difungsikan sebagai receiver untuk menerima data yang telah dikirim oleh Transmitter SX1278. Mikrokontroler Arduino Uno digunakan untuk pemrosesan data yang telah diterima. NODEMCU ESP8266 dihubungkan secara serial dengan Arduino Uno untuk menampilkan data di aplikasi Blynk. Spesifikasi alat secara keseluruhan dapat dilihat pada tabel 3.1.



Gambar 3.3 Diagram Sistem Penerima.

Tabel 3.1 Spesifikasi Alat

Karakteristik	Blok Pengirim	Blok Penerima
Dimensi	(15 x 11 x 5) cm	(5 x 5 x 3) cm
Berat Total	125 gram	25
Mikrokontroler	Arduino Nano	Arduino Uno, Nodemcu ESP8266
Jumlah Sensor	4 buah <i>Capacitive Soil Moisture Sensor</i>	-
Modul Lain	RTC DS1307, SX127	SX1278
Output	Relay, <i>Submersible Pump 3 Volt</i>	-
Tegangan Masukan	5 Volt	5 Volt
Tegangan Keluaran	3.3 Volt dan 5 Volt	3.3 Volt dan 5 Volt

3.1 Diagram Sistem

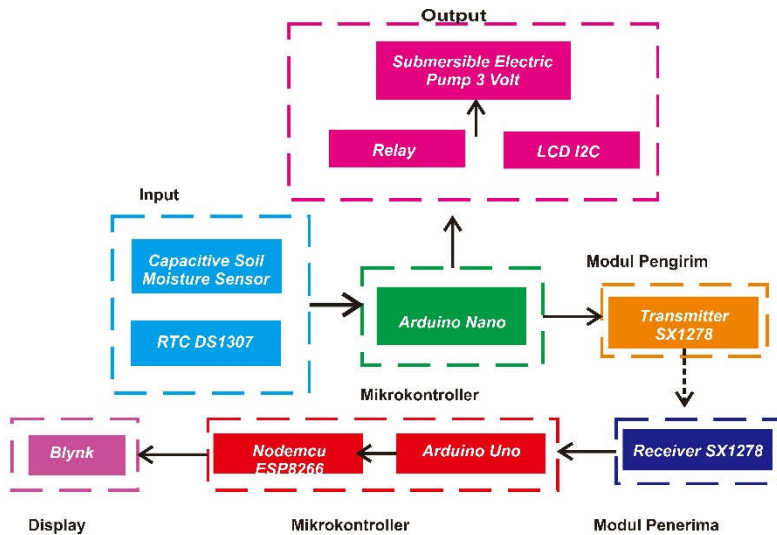
Pada tugas akhir ini, mikrokontroler yang digunakan sebagai unit pengolah data yaitu Arduino baik pada blok penerima maupun pengirim. Diagram sistem keseluruhan bisa dilihat pada gambar 3.4. Pada bagian penerima, terdapat mikrokontroler lain yang digunakan yaitu Nodemcu NODEMCU ESP8266. Nodemcu ini digunakan sebagai penghubung ke web server. Arduino pada penerima dihubungkan serial dengan Nodemcu untuk mengambil data yang dikirim oleh *transmitter*.

Sensor yang digunakan adalah *Capacitive Soil Moisture Sensor*. Sensor Capacitive Soil Moisture Sensor diletakkan di dalam tanah untuk mengukur tingkat kelembabannya. Ada 4 sensor yang digunakan pada penelitian ini. Selain Capacitive Soil Moisture Sensor, alat ini juga menggunakan RTC untuk mengetahui lamanya waktu tanam. Hasil dari pengukuran sensor dan RTC akan ditampilkan dalam LCD 20 x 4. Sumber tegangan dari mikrokontroler dan sensor didapat dari baterai LiPo yang telah diturunkan tegangannya terlebih dahulu menggunakan *buck converter*.

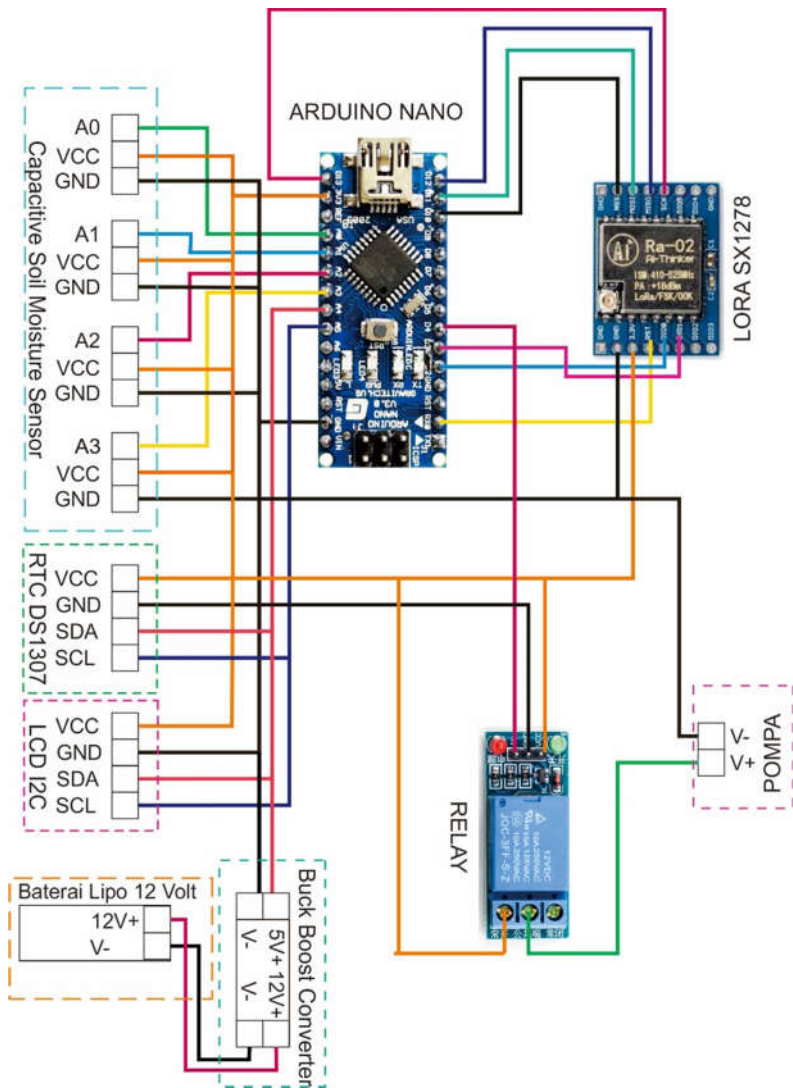
Sensor Capacitive Soil Moisture Sensor dibaca oleh mikrokontroler Arduino Nano. Hasil pembacaan dari sensor dan RTC ini akan diproses dalam Mikrokontroler Arduino menggunakan logika fuzzy. Parameter yang digunakan dalam proses control ini yaitu kelembaban tanah dan waktu masa tanam. Sistem irigasi ini dilengkapi dengan submersible

electric pump sebagai actuator yang dikontrol.

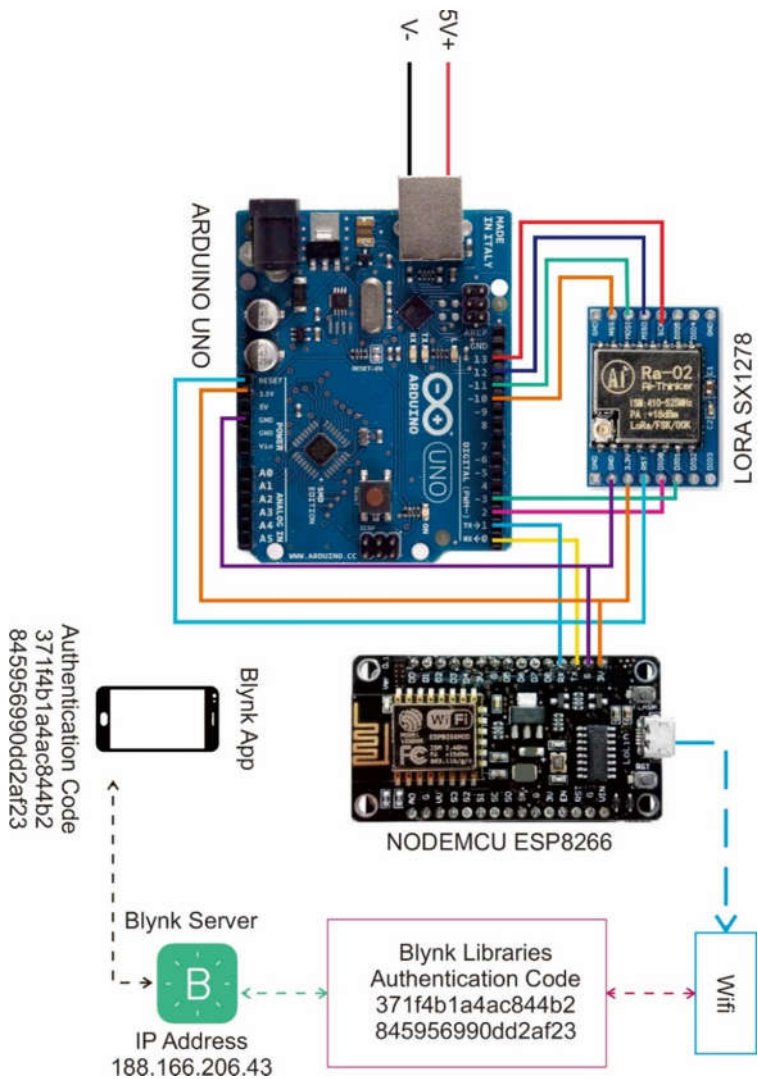
Hasil pemrosesan dari sensor, RTC dan fuzzy logic akan dikirim menggunakan modul Transceiver LoRa SX1278. Data yang dikirim akan diterima oleh Receiver LoRa SX1278 dan diproses oleh Mikrokontroler Arduino. Jarak antara transmitter dan receiver bervariasi maksimal 15 Km. Data yang telah diterima akan dikirimkan ke blynk server. Penjelasan sisten yang lebih lengkap bisa dilihat pada gambar 3.5 untuk arsitektur pengirim dan gambar 3.6 untuk arsitektur penerima.



Gambar 3.4 Diagram Blok Sistem



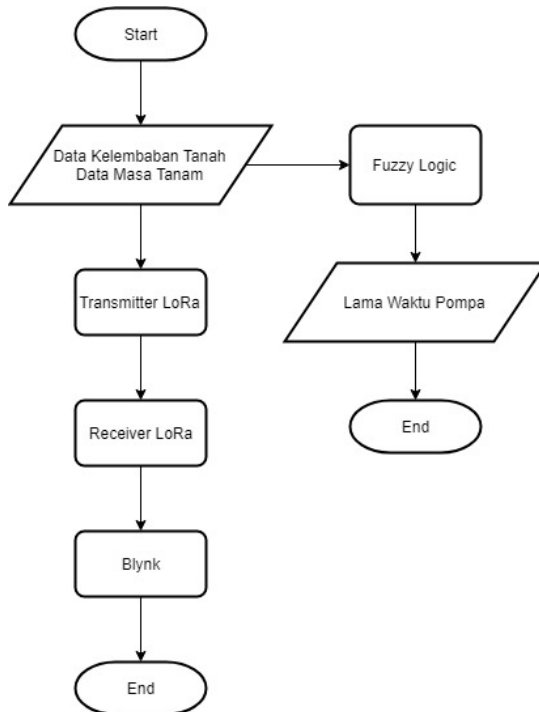
Gambar 3.5 Arsitektur Sistem Pengirim



Gambar 3.6 Arsitektur Sistem Penerima

3.2 Flow Chart Sistem

Pada tugas akhir ini, sensor membaca kelembaban tanah secara real time dan RTC membaca waktu pengoperasian alat tersebut. Data kelembaban tanah dan waktu akan diproses oleh Arduino menggunakan fuzzy logic untuk menentukan keluaran dari sistem tersebut. Keluaran yang dihasilkan berupa lama waktu pompa menyala. Data kelembaban tanah dan waktu juga akan dikirim menggunakan SX1278. Data yang dikirim diterima oleh *receiver* menggunakan SX1278 dengan mikrokontroler Arduino. Setelah data telah diterima oleh Arduino, data dikirim serial ke Nodemcu. Nodemcu akan mengirim data ke web server local. Pada gambar 3.7 ditunjukkan flow chart yang digunakan pada sistem ini.



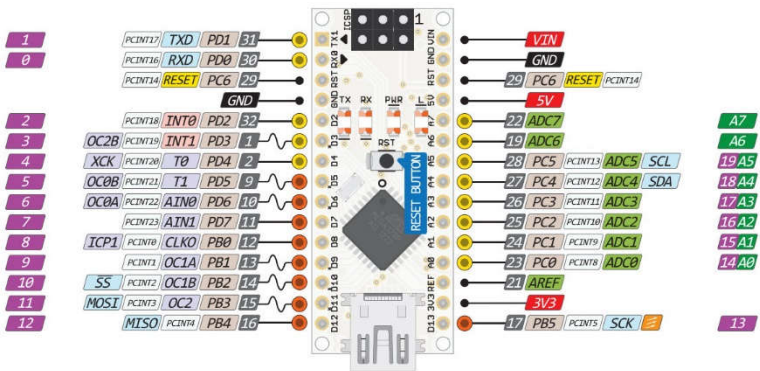
Gambar 3.7 Flow Chart Sistem

3.3 Perancangan Perangkat Keras

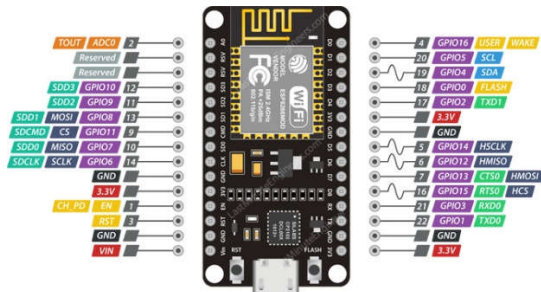
Perangkat keras pada tugas akhir yaitu desain elektronik. Desain elektronik meliputi penyusunan *board* penerima dan pengirim. Board pengirim terdiri dari Arduino, Capacitive Soil Moisture Sensor, RTC, Transmitter LoRa SX1278 dan submersible electric pump. Sedangkan blok penerima terdiri dari Arduino, Receiver LoRa SX1278, dan Modul Wifi.

3.3.1. Mikrokontroler

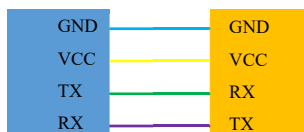
Pada tugas akhir ini mikrokontroler yang digunakan adalah Arduino Nano dan Uno. Pada gambar 3.8 dijelaskan pin out dari Arduino Nano. Pemilihan jenis mikrokontroler ini dikarenakan fitur yang cukup memadai untuk pemrosesan dalam alat ini. Arduino Nano digunakan untuk pemrosesan hasil sensor yang telah terukur menggunakan *Fuzzy Logic*. Output dari *Fuzzy Logic* berupa lama waktu penyiraman. Data sensor yang telah diukur juga akan dikirimkan menggunakan modul LoRa SX1278. Tegangan kerja mikrokontroler Arduino Nano yaitu 3,3 V. Pada sistem ini digunakan pin digital sebanyak 4 pin (pin digital 1,2,3, dan 10), pin analog sebanyak 4 pin (pin A0, A1, A2, A3), pin I2C (SDA dan SCL), dan pin SPI (MISO, MOSI, dan SCK). Selain itu pin 3,3V dan GND digunakan untuk input dari mikrokontroler Arduino Nano. *Board* mikrokontroler didapatkan secara open source. Arduino Uno digunakan untuk penerimaan data yang dikirim oleh pengirim menggunakan modul LoRa SX1278. Pada mikrokontroler ini, pin yang digunakan berupa pin digital sebanyak pin (pin digital 2,3, dan 10), pin SPI (MISO, MOSI, dan SCK), pin komunikasi serial (RX dan TX), dan pin 3,3 V serta GND sebagai input dari mikrokontroler Arduino Uno. Mikrokontroler dihubungkan secara serial seperti pada gambar 3.10 dengan Nodemcu untuk proses penampilan di web server dari data yang telah diterima. Mikrokontroler Nodemcu memiliki modul wifi sehingga dapat digunakan mengakses wifi dan blynk. Pinout dari Nodemcu ESP8266 ditunjukkan pada gambar 3.9.



Gambar 3.8 Skematik Arduino Nano.



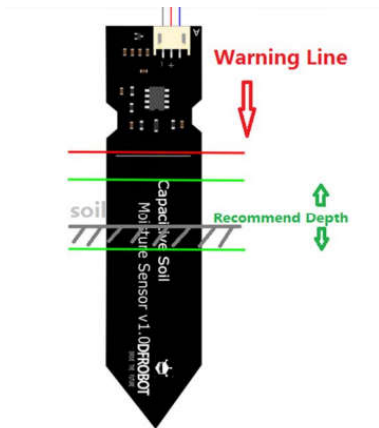
Gambar 3.9 Skema Nodemcu NODEMCU ESP8266



Gambar 3.10 Antarmuka Arduino Uno dan Nodemcu ESP8266

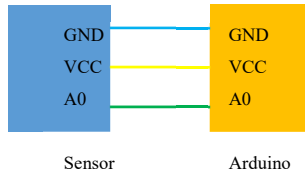
3.3.2. Capacitive Soil Moisture Sensor

Capacitive Soil Moisture Sensor merupakan sensor yang digunakan untuk mengukur kadar kelembaban tanah. Dimensi ukuran sensor ini yaitu 3.86 x 0.905 inches dengan massa 15 gram. Tegangan masukan dari sensor ini adalah 3.3 Volt. Sensor diletakkan didalam tanah dengan kedalaman yang tepat. Kedalaman yang salah dapat mempengaruhi nilai pembacaan sensor tersebut. Terdapat area yang tidak boleh masuk ke dalam tanah dan ada bagian yang boleh masuk ke dalam tanah. Batas sensor yang masuk ke tanah ditunjukkan pada gambar 3.11.



Gambar 3.11 Batas Capacitive Soil Moisture Sensor

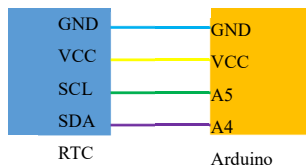
Antarmuka antara *Capacitive soil moisture sensor* dan mikrokontroler Arduino nano yaitu menggunakan pin A0 sampai A3. Selain pin tersebut, pin 3.3 Volt dan Gnd juga digunakan pada antarmuka ini. Jumlah pin analog yang digunakan sebanyak 4 dikarenakan sensor yang digunakan pada sistem ini berjumlah 4. Antarmuka *Capacitive Soil Moisture Sensor* seperti pada Gambar 3.12.



Gambar 3.12 Antarmuka Capacitive Soil Moisture Sensor

3.3.3. RTC DS1307

RTC adalah sebuah modul yang digunakan untuk menjalankan fungsi waktu dan kalender. Modul ini digunakan untuk mengetahui waktu masa tanam. Dimensi modul RTC ini yaitu 27 x 28 x 8.4mm. Tegangan masukan yang digunakan untuk supply modul ini adalah 3.3 Volt. Modul ini akan dijadikan dalam satu board dengan mikrokontroller. Antarmuka pada modul RTC ini yaitu menggunakan pin I2C yaitu SCL dan SDA atau pin A4 dan pin A5. Tegangan kerja dari RTC adalah 3.3 Volt. Skematik dan antarmuka RTC dengan mikrokontroller Arduino Nano ditunjukkan pada Gambar 3.13.

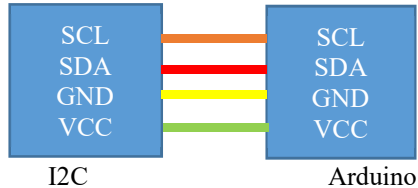


Gambar 3.13 Antarmuka RTC DS1307

3.3.4. LCD I2C

LCD pada tugas akhir ini digunakan untuk menampilkan data kelembaban tanah, waktu tanam, dan kondisi tandon. Rangkaian LCD ini menggunakan modul I2C LCD agar tidak memakan banyak pin di mikrokontroller Arduino Nano. Pin Gnd, VCC, VEE, RS, R/W, EN, DB0, DB1, DB2, DB3, DB4, DB5, DB6, DB7, Led+ dan Led- pada LCD dan Modul I2C LCD masing masing dihubungkan. Sedangkan pin SCL, SDA, VCC dan Gnd pada modul I2C dihubungkan ke Mikrokontroller Arduino Nano di pin SCL, SDA, VCC, dan Gnd. LCD yang digunakan adalah

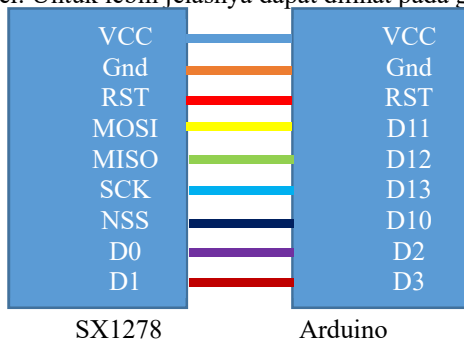
LCD 20 x 4 dikarenakan LCD ukuran tersebut dapat memuat data yang diperlukan. Skematik antara Arduino dan LCD I2C ditunjukkan pada gambar 3.14.



Gambar 3.14 Antarmuka I2C dengan Arduino

3.3.5. LoRa SX1278

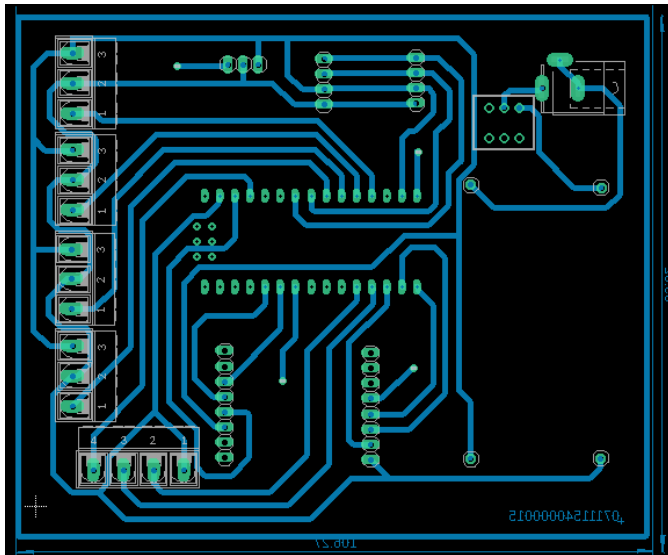
LoRa SX1278 merupakan sebuah modul pengiriman yang digunakan. Penggunaan modul dikarenakan jangkauan pengiriman yang jauh. Modul pengiriman ini dapat mencapai jarak 10 Km. Modul yang digunakan sebanyak 2 buah yaitu sebagai Transmitter dan Receiver. Modul ini dihubungkan ke mikrokontroler baik untuk penerima maupun pengirim. Antarmuka antara SX1278 dan mikrokontroler menggunakan pin MISO, MOSI, SCK, NSS, VCC, Gnd, D0, D1, dan Reset. Untuk MISO, MOSI, SCK, VCC, Gnd dan Reset pada SX1278 dihubungkan ke mikrokontroler. Untuk lebih jelasnya dapat dilihat pada gambar 3.15.



Gambar 3.15 Antarmuka SX1278

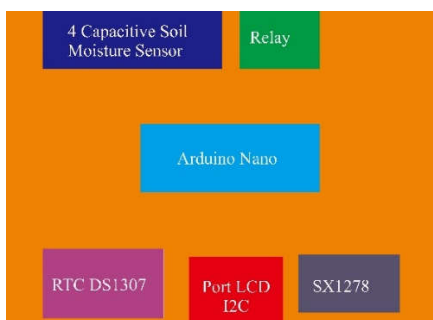
3.3.6. Board Mikrokontroler Terpadu

Board mikrokontroler terpadu merupakan gabungan dari komponen-komponen yang sudah dijelaskan sebelumnya. Ada 2 *Board* yang digunakan yaitu *Board* pengirim dan penerima. *Board* terpadu pertama terdiri dari mikrokontroler Arduino Nano, *Capacitiver Soil Moisture Sensor*, modul LoRa SX1278, LCD I2C dan Relay. Untuk *Capacitiver Soil Moisture Sensor* diletakkan di luar karena fungsi sensor ini adalah mengukur kelembaban tanah. Sedangkan *Board* penerima terdiri dari mikrokontroler Arduino Nano, Nodemcu, dan modul LoRa SX1278. *Board* ini menggunakan catu daya 3.3 Volt dan 5 Volt. Skematik *board* ini dibuat menggunakan *software* EAGLE. Setelah dibuat skematik dari komponen-komponen yang djelaskan sebelumnya adalah menata letak dari komponen-komponen tersebut agar didapatkan desain yang efisien dengan dimensi yang kecil. Tahap selanjutnya adalah penyolderan komponen ke *board* tersebut. Skema dari *Board* ini ditunjukkan pada Gambar 3.16.

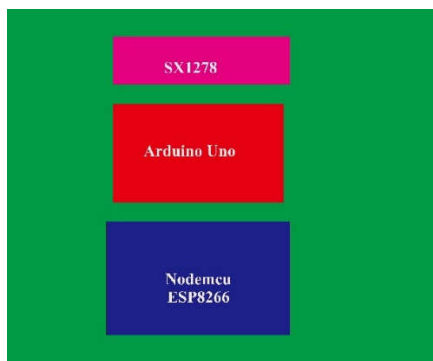


Gambar 3.16 Desain PCB Pengirim

Hasil dari desain *board* mikrokontroler terpadu dicetak dalam bentuk PCB. Dimensi board setelah dipasang semua komponennya memiliki dimensi 11 x 87 x 20 mm. PCB ini memiliki spesifikasi *single layer*, dan *through hole*. *Single layer* merupakan PCB yang hanya memiliki satu lajur pada salah satu sisinya. *Through hole* adalah lobang PCB yang memiliki jalur elektrik sehingga kedua sisinya pada lubang tersebut terhubung menjadi satu jalur. Setelah komponen telah tersolder, proses selanjutnya adalah memasukkan program yang telah dibuat ke dalam mikrokontroler. Layout blok pengirim dapat dilihat pada gambar 3.17 dan blok penerima pada gambar 3.18.



Gambar 3.17 Layout Blok Pengirim



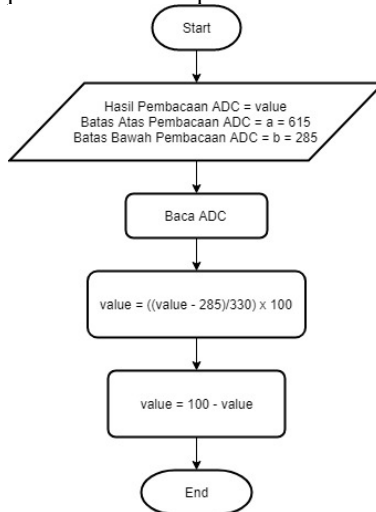
Gambar 3.18 Layout Blok Penerima

3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak terintegrasi ke dalam mikrokontroler Arduino Nano. Perancangan algoritma pemrograman digunakan untuk melakukan beberapa proses di antaranya pembacaan *Capacitive Soil Moisture Sensor*; LCD I2C, modul LoRa SX1278, relay, dan *fuzzy logic*.

3.4.1 Pembacaan Capacitive Soil Moisture Sensor

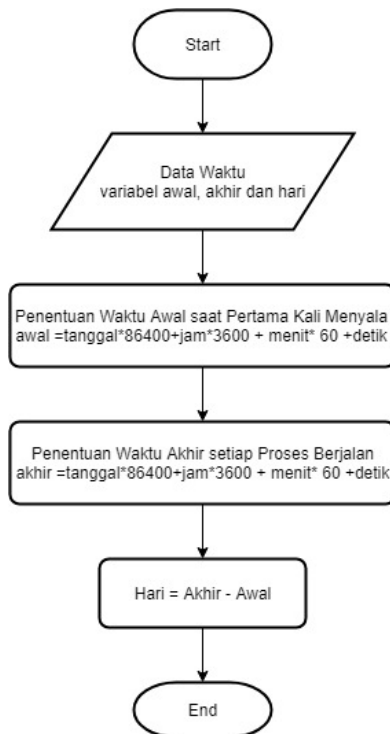
Pembacaan *Capacitive Soil Moisture Sensor* menggunakan pembacaan analog dari Arduino Nano. Pembacaan analog dari Arduino Nano merupakan tegangan yang dikeluarkan oleh sensor. Tegangan akan berubah-ubah sesuai dengan perubahan kelembaban tanah. Pembacaan analog ini memiliki skala 0 sampai 1023. Akan tetapi, hasil pada pengukuran ini diskalakan ulang antara 285 – 615. Hal ini berdasarkan pengukuran kelembaban tanah di air dan udara sebagai batas atas dan bawah hasil pengukuran analog Arduino Nano. Skala 285 – 615 yang telah ditentukan akan dimapping menjadi 0 sampai 100. Hasil mapping tersebut menunjukkan tingkat persentase dari kelembaban tanah. Pada gambar 3.19 ditampilkan flow chart pembacaan sensor.



Gambar 3.19 Flowchart Pembacaan Sensor

3.4.2 Pembacaan RTC DS1307 dan Penentuan Hari Masa Tanam

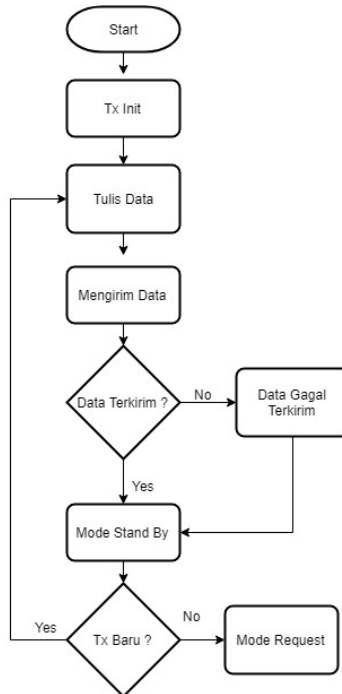
Pembacaan RTC menggunakan library yang ada di Arduino yaitu "RTClib.h". RTC akan menunjukkan waktu secara real time saat alat beroperasi. Dari hasil waktu yang didapat akan digunakan fase masa tanam dari tanaman tersebut. Alat ini akan menentukan waktu awal ketika pertama kali dioperasikan dan mencatat waktu secara real time. Waktu real time akan dikurangkan dengan waktu awal sehingga didapatkan hasil waktu masa tanam dari tanaman tersebut. Flow chart pembacaan RTC bisa dilihat pada gambar 3.20.



Gambar 3.20 Flowchart Pembacaan RTC

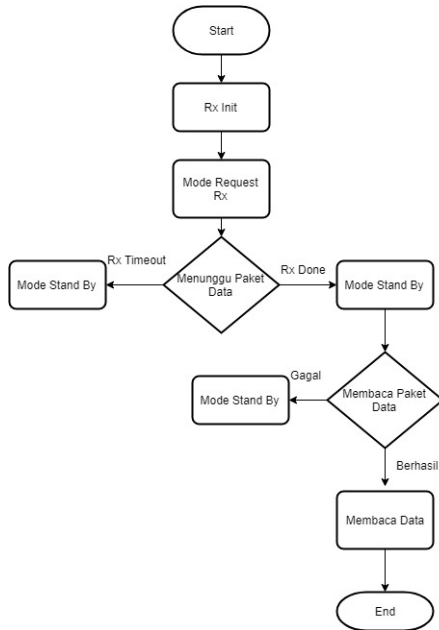
3.4.3 Pengiriman Data Menggunakan SX1278

Data sensor yang telah didapatkan akan dikirim menggunakan modul LoRa SX1278. Pengiriman data dari modul SX1278 menggunakan Library yang ada di Arduino yaitu “LoRaLib.h”. Flowchart pengiriman data menggunakan SX1278 ditunjukkan pada gambar 3.21.



Gambar 3.21 Flowchart Modul Pengirim SX1278

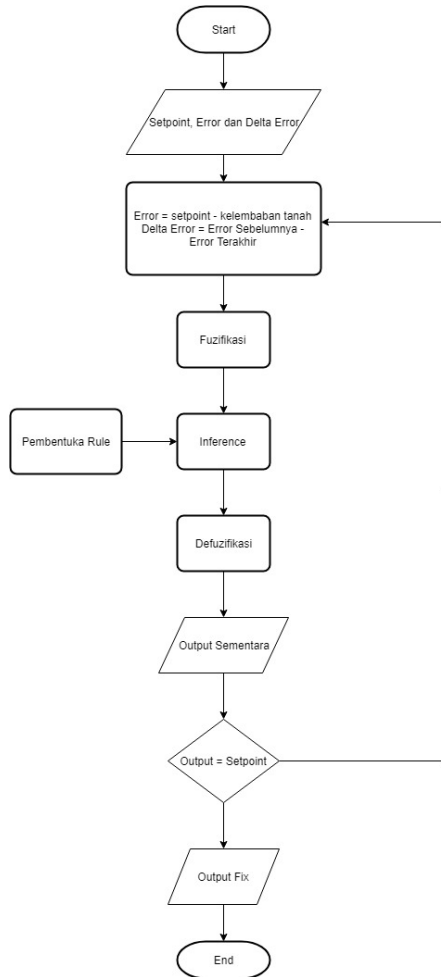
Data yang telah dikirim oleh modul LoRa SX1278 akan diterima oleh modul yang sama yaitu modul LoRa SX1278. Hal ini bisa digunakan karena modul ini merupakan modul Transceiver. Oleh karena modul ini bisa menjadi pengirim dan menerima. Pada gambar 3.22 menunjukkan flowchart penerimaan data SX1278.



Gambar 3.22 Flowchart Modul Penerima SX1278

3.5 Fuzzy Logic

Data sensor yang telah didapatkan akan dikontrol untuk mendapatkan hasil kelembaban tanah yang sesuai. Metode control yang digunakan adalah fuzzy logic. Fuzzy logic memiliki 3 bagian yaitu fuzzifikasi, rule dan defuzzifikasi. Setiap bagian memiliki fuzzifikasi, rule dan defuzzifikasi yang berbeda. Pada gambar 3.23 menunjukkan flowchart proses fuzzy logic.



Gambar 3.23 Flowchart Fuzzy Logic

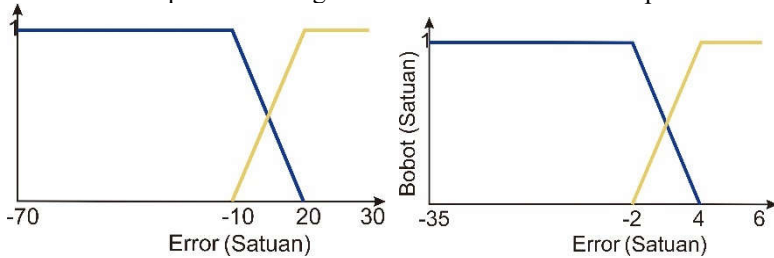
3.5.1 Fuzifikasi

Fuzzifikasi adalah penentuan anggota himpunan dari fuzzy logic. Pembentukan rule adalah aturan yang diterapkan pada kondisi yang telah ditetapkan. Sedangkan defuzifikasi adalah pengolahan input terhadap rule

sehingga dihasilkan keluaran yang dibutuhkan. Ada 2 himpunan masukan yang digunakan pada fuzzy logic ini yaitu error kelembaban tanah terhadap setpoint dan perbedaan besar error real dengan sebelumnya. Dikarenakan setiap waktu masa tanam harus memiliki kelembaban yang berbeda, maka anggota himpunan keanggotaan fuzzy logic dibagi menjadi yaitu kering, sedang, dan basah. Ketika masa tanam fase 1 dan 2, kelembaban tanah yang digunakan adalah kelembaban tanah basah. Kelembaban tanah sedang digunakan untuk masa tanam fase 3. Sedangkan fase tanam 4 menggunakan kelembaban tanah kering.

a. Masa Tanam Fase ke 4 (Kelembaban Kering)

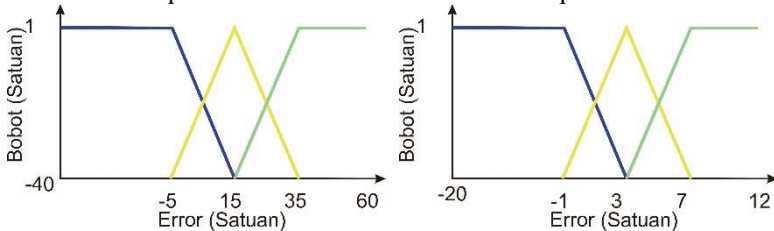
Pada gambar 3.24 menunjukkan fuzifikasi pada masa tanam ke 4 atau kelembaban kering. Pada fuzifikasi ini dilihat bahwa rentang error adalah -70 sampai 30. Sedangkan delta error antara -35 sampai 6.



Gambar 3.24 Fuzifikasi Kelembaban Tanah Kering

b. Masa Tanam Fase Ke 3 (Kelembaban Sedang)

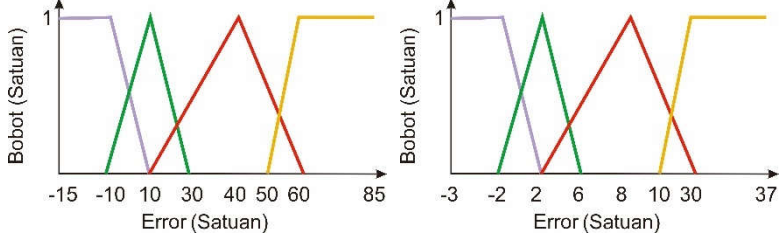
Pada gambar 3.25 menunjukkan fuzifikasi pada masa tanam ke 3 atau kelembaban sedang. Pada fuzifikasi ini dilihat bahwa rentang error adalah -40 sampai 60 dan delta error antara -20 sampai 12.



Gambar 3.25 Fuzifikasi Kelembaban Tanah Sedang

c. Masa Tanam Fase ke 1 dan 2 (Kelembaban Basah)

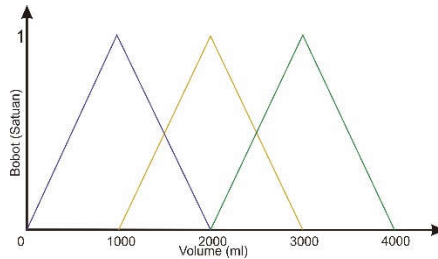
Pada gambar 3.26 menunjukkan fuzifikasi pada masa tanam ke 1 dan 2 atau kelembaban basah. Pada fuzifikasi ini dilihat bahwa rentang error adalah -15 sampai 85 dan delta error antara -3 sampai 37.



Gambar 3.26 Fuzifikasi Kelembaban Tanah Basah

d. Output

Pada gambar 3.27 menunjukkan fuzifikasi output lama pompa menyala. Pada fuzifikasi ini dilihat bahwa pembagian keanggotaan dibagi menjadi 3 yaitu sedikit, sedang dan banyak. Untuk anggota sedikit antara 0 sampai 200, sedang antara 1000 sampai 3000, dan banyak antara 2000 sampai 4000.



Gambar 3.27 Fuzifikasi Output

3.5.2 Pembentukan Rule

a. Masa Tanam Fase ke 1 dan 2 (Kelembaban Tanah Basah)

Rule yang digunakan pada fase tanam 1 dan 2 atau kelembaban tanah basah ditunjukkan pada tabel 3.2

Tabel 3.2 Rule Kelembaban Tanah Basah

	Δ Error 1	Δ Error 2	Δ Error 3	Δ Error 4
Error 1	Stop	Stop	Sedikit	Sedang
Error 2	Stop	Sedikit	Sedang	Sedang
Error 3	Sedikit	Sedikit	Sedang	Lama
Error 4	Sedikit	Sedang	Lama	Lama

b. Masa Tanam Fase Ke 3 (Kelembaban Tanah Sedang)

Rule yang digunakan pada fase tanam 3 atau kelembaban tanah sedang ditunjukkan pada tabel 3.3

Tabel 3.3 Rule Kelembaban Tanah Sedang

	Δ Error 1	Δ Error 2	Δ Error 3
Error 1	Stop	Stop	Sedikit
Error 2	Stop	Sedikit	Sedang
Error 3	Sedikit	Sedang	Sedang

c. Masa Tanam Fase Ke 4 (Kelembaban Tanah Kering)

Rule yang digunakan pada fase tanam 4 atau kelembaban tanah kering ditunjukkan pada tabel 3.4

Tabel 3.4 Rule Kelembaban Tanah Kering

	Δ Error 1	Δ Error 2
Error 1	Stop	Stop
Error 2	Stop	Sedikit

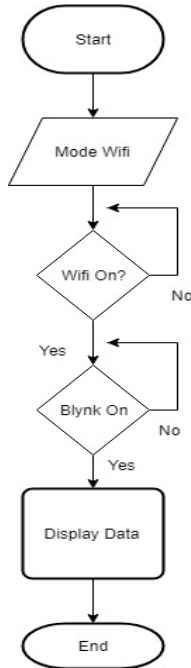
3.5.3 Defuzzifikasi

Defuzzifikasi adalah menghubungkan antara input data, anggota fuzifikasi, dan rule. Metode yang digunakan pada defuzzifikasi ini adalah metode sugeno. Metode ini menggunakan nilai rata-rata dari korelasi input, anggota fuzifikasi dan rule. Nilai dari defuzzifikasi didapatkan dengan mengambil nilai tengah antara hubungan ketiga tiganya.

3.6 Perancangan Blynk

Pada tahap ini adalah proses desain tampilan di aplikasi blynk. Proses desain ini menggunakan widget yang sudah disediakan oleh

aplikasi blynk sendiri. Widget yang digunakan pada tahap ini berupa Value Display dan Superchart. Value display digunakan untuk menampilkan data yang telah terbaca oleh SX1278. Sedangkan Superchart untuk menampilkan grafik perubahan rata rata kelembaban yang terbaca oleh sensor. Berikut diagram alir dari perancangan program Blynk sesuai pada gambar 3.28.



Gambar 3.28 Flowchart Penampilan Blynk

.....*Halaman ini sengaja dikosongkan*.....

BAB IV

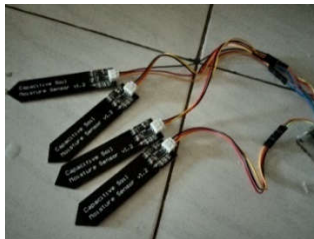
PENGUJIAN DAN ANALISIS

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang pada bab sebelumnya. Bab ini bertujuan untuk mendapatkan data yang kemudian dilakukan analisa pada masing-masing pengujian. Pengujian yang dilakukan meliputi pengujian sensor

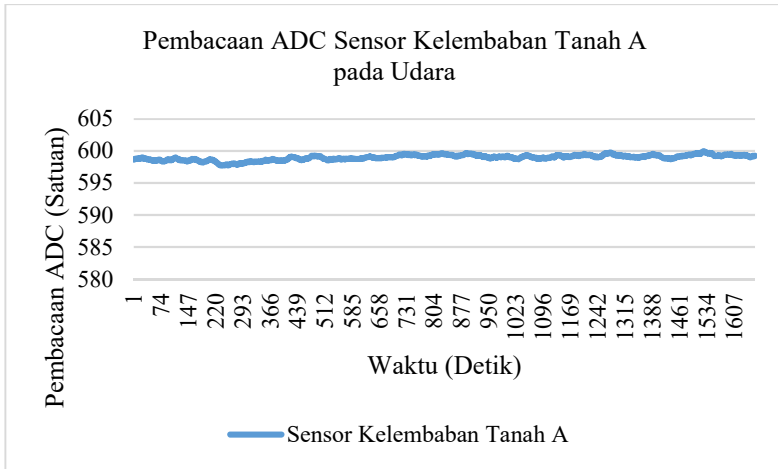
4.1. Pengujian Karakteristik Capacitive Soil Moisture Sensor

4.1.1. Penentuan LSB dan MSB ADC Sensor

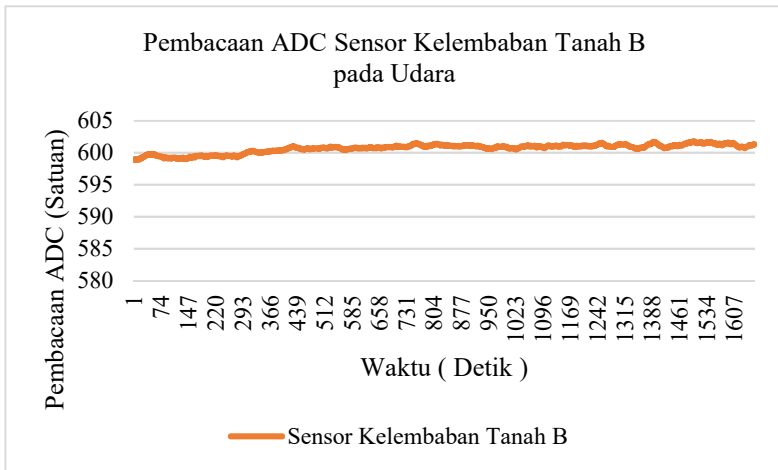
Pengujian sensor *Capacitive Moisture Soil Sensor* dilakukan dengan menggunakan mikrokontroler Arduino Nano. Tahap awal pengujian ini yaitu dengan meletakkan sensor di udara dan air. Pengujian di udara ditampilkan pada gambar 4.1 dan pengujian di air digambarkan pada gambar 4.6. Hal tersebut dilakukan untuk mengetahui batas bawah dan atas dalam pengukuran sensor tersebut. Rentang dari batas bawah dan atas tersebut akan dimapping menjadi rentang 0 sampai 100. Dari rentang tersebut akan dibagi menjadi 3 karakteristik yaitu kelembaban kering, sedang dan basah. Setelah rentang bawah dan atas dimapping, sensor digunakan untuk mengukur kelembaban pada tanah. Tanah yang digunakan adalah tanah yang ada di halaman rumah yang berlokasi di Desa Kewangunan, Kecamatan Petanahan, Kabupaten Kebumen, Provinsi Jawa Tengah. Pengujian sensor terhadap tanah ini dilakukan dengan mengukur kelembaban tanah dalam kondisi kering, sedang, dan basah. Tanah yang diukur diletakkan dalam gelas plastik ukuran kecil.



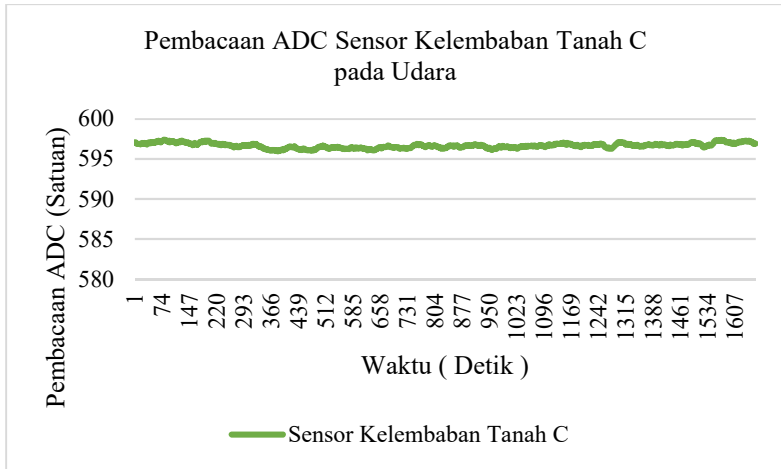
Gambar 4.1 Pengujian Kelembaban Tanah pada Udara



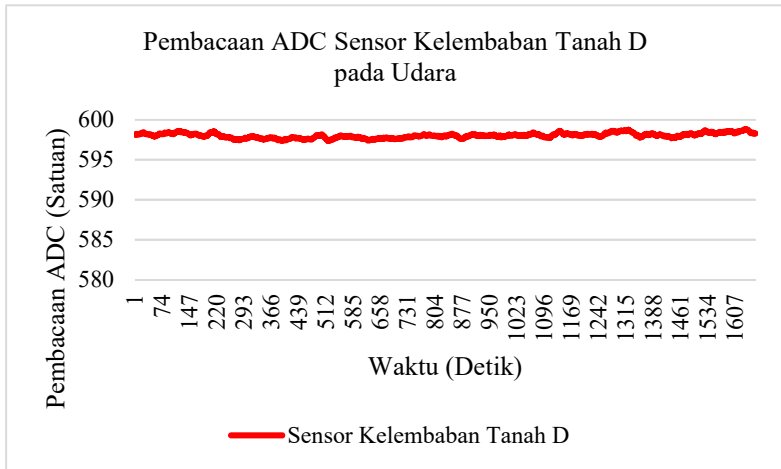
Gambar 4.2 Hasil Pengukuran Sensor Kelembaban Tanah A pada Udara



Gambar 4.3 Hasil Pengukuran Sensor Kelembaban Tanah B pada Udara



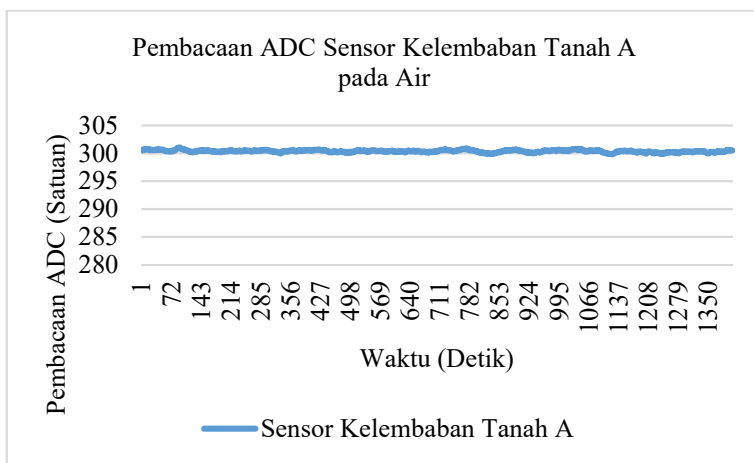
Gambar 4.4 Hasil Pengukuran Sensor Kelembaban Tanah C pada Udara



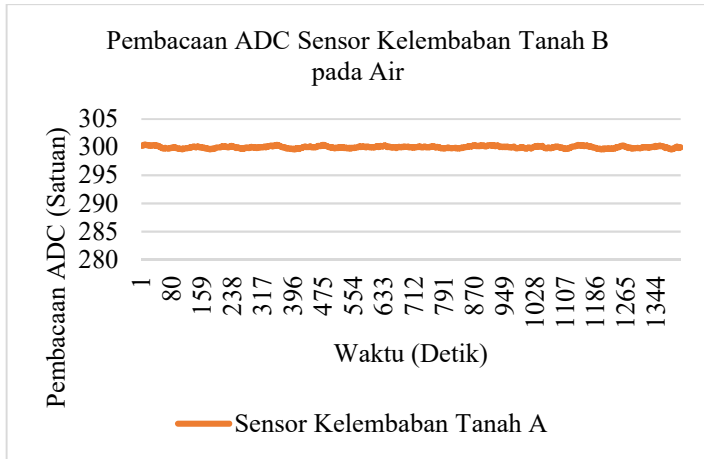
Gambar 4.5 Hasil Pengukuran Sensor Kelembaban Tanah D pada Udara



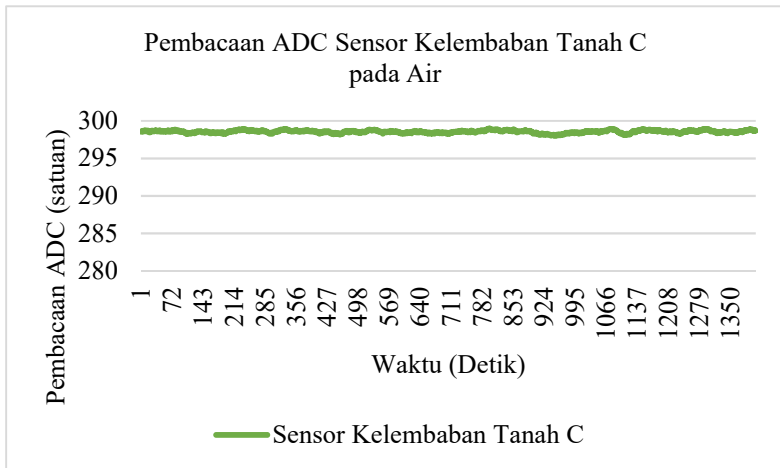
Gambar 4.6. Pengujian Kelembaban Tanah pada Air



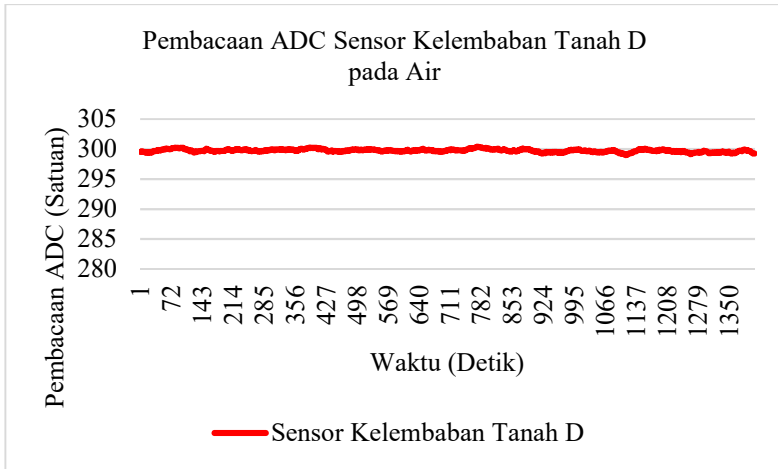
Gambar 4.7 Hasil Pengukuran Sensor Kelembaban Tanah A pada Air



Gambar 4.8 Hasil Pengukuran Sensor Kelembaban Tanah B pada Air



Gambar 4.9 Hasil Pengukuran Sensor Kelembaban Tanah C pada Air



Gambar 4.10 Hasil Pengukuran Sensor Kelembaban Tanah D pada Air

Dari hasil pembacaan tersebut didapatkan bahwa pengukuran untuk batas atas dan bawah memiliki nilai fluktuatif yang kecil. Pembacaan sensor untuk batas atas dengan sampel uji kelembaban tanah di udara adalah 595 sampai 601. Hal tersebut ditunjukkan pada gambar 4.1 sampai gambar 4.3. Sedangkan hasil dari pengukuran batas bawah dengan sampel uji kelembaban tanah di air adalah 297 sampai 315 yang dijelaskan pada gambar 4.7 sampai 4.10. Oleh karena itu, rentang mapping yang digunakan pada pemrograman adalah 285 sampai 615. Pemilihan nilai yang lebih tinggi dari batas atas dan lebih rendah dari batas bawah dikarenakan untuk mengantisipasi pembacaan nilai yang lebih tinggi atau bawah. Dari hasil tersebut akan dimapping menjadi range 0 sampai 100.

4.1.2. Pengujian Output Tegangan Sensor

Pada tahap ini dilakukan pengujian output tegangan sensor ketika mengukur kelembaban tanah. Penentuan batas bawah output tegangan sensor dilakukan dengan mengukur tegangan output saat mengukur kelembaban tanah di air. Sedangkan penentuan batas atas output tegangan sensor dilakukan dengan mengukur tegangan output saat mengukur kelembaban tanah di udara. Berikut hasil dari pengujian output tegangan

sensor dari 4 sensor yang ditampilkan pada tabel 4.1.

Tabel 4.1. Pengujian Output Tegangan Sensor

Sensor Kelembaban Tanah	Tegangan output minimum (Volt)	Tegangan output maksimum (Volt)
Sensor A	1,469726563	2,923014323
Sensor B	1,467936198	2,924641927
Sensor C	1,465983073	2,915527344
Sensor D	1,457845052	2,920572917

4.1.3. Pengujian Operational Voltage Sensor

Pada tahap ini dilakukan pengujian tegangan operasional minimum dan maksimum sensor kelembaban tanah ini. Pengujian ini dilakukan dengan cara mengubah tegangan input sensor dari 0 sampai tegangan maksimum alat tidak bisa membaca kelembaban tanah. Berikut hasil yang didapatkan.

Tabel 4.2 Pengujian *Operational Voltage* Sensor

Sensor Kelembaban Tanah	Tegangan operasional minimum (Volt)	Tegangan operasional maksimum (Volt)
Sensor Kelembaban Tanah Kapasitif	3,12	5,22

4.1.4. Kalibrasi Kelembaban Tanah

Setelah menentukan batas atas dan awal dengan penskalaan ulang, selanjutnya menentukan batas tingkat kelembaban tanah. Tingkat kelembaban yang digunakan ada 3 yaitu kelembaban kering, sedang, dan basah. Untuk menentukan batas tersebut, maka digunakan alat pengukur kelembaban tanah yang telah terkalibrasi.

4.1.4.1. Kelembaban Tanah Kering

Tahap ini dilakukan pengujian pada tanah dengan kelembaban kering sesuai dengan gambar 4.11. Kelembaban tanah ini diukur menggunakan sensor yang telah terkalibrasi dan sensor yang digunakan pada penelitian ini. Pengujian ini bertujuan menentukan batas kelembaban tanah kering

sampai pada persentase tertentu. Berikut hasil dari pengujian kelembaban pada tanah kering yang ditampilkan pada tabel 4.3.

Tabel 4.3 Tingkat Kelembaban Tanah Kering

Nilai Kelembaban (%)	Kriteria
30	Kering
21	Kering
8	Kering
17	Kering



Gambar 4.11 Kalibrasi Kelembaban Tanah Kering

4.1.4.2. Kelembaban Tanah Sedang

Tahap ini dilakukan pengujian pada tanah dengan kelembaban sedang sesuai dengan gambar 4.12. Kelembaban tanah ini diukur menggunakan sensor yang telah terkalibrasi dan sensor yang digunakan pada penelitian ini. Pengujian ini bertujuan menentukan batas kelembaban tanah kering sampai pada persentase tertentu. Hasil dari pengujian kelembaban pada tanah basah yang ditunjukkan pada tabel 4.4.

Tabel 4.4 Tingkat Kelembaban Tanah Sedang

Nilai Kelembaban (%)	Kriteria
70	Sedang
65	Sedang
62	Sedang
50	Sedang



Gambar 4.12 Kalibrasi Kelembaban Tanah Sedang

4.1.4.3. *Kelembaban Tanah Basah*

Tahap ini dilakukan pengujian pada tanah dengan kelembaban basah sesuai gambar 4.5. Kelembaban tanah ini diukur menggunakan sensor yang telah terkalibrasi dan sensor yang digunakan pada penelitian ini. Pengujian ini bertujuan menentukan batas kelembaban tanah kering sampai pada persentase tertentu. Berikut hasil dari pengujian kelembaban pada tanah basah yang ditampilkan dalam tabel 4.5.

Tabel 4.5 Tingkat Kelembaban Tanah Basah

Nilai Kelembaban (%)	Kriteria
78	Basah
80	Basah
86	Basah
93	Basah

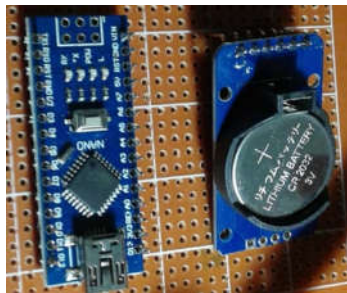


Gambar 4.13 Kalibrasi Kelembaban Tanah Basah

Berdasarkan data yang didapatkan, batas kelembaban tanah kering adalah antara 0 sampai 40. Batas kelembaban tanah sedang adalah 41 sampai 75. Sedangkan batas kelembaban basah adalah 76 sampai 100.

4.2. Pembacaan RTC

Pembacaan RTC dilakukan dengan menggunakan mikrokontroler Arduino Nano. RTC dikonfigurasi ke mikrokontroler sesuai yang ada di gambar 4.14. Pada pengujian ini, ada 5 data yang ditampilkan. Data pertama adalah data waktu pertama kali alat ini dioperasikan. Data kedua menunjukkan waktu beroperasi alat sekarang. Data ketiga adalah sebuah perhitungan yang merepresentasikan waktu awal operasi. Data keempat merupakan data perhitungan real time waktu alat beroperasi. Data kelima adalah data hari dimana alat bekerja. Dalam pengujian ini, nilai hari diskalakan menjadi 1 detik. Oleh karena itu, 1 detik direpresentasikan dengan 1 hari. Apabila nilai hari direpresentasikan sesuai nilai aslinya, maka tinggal diubah pada perhitungan nilai awal dan akhir. Berikut hasil pengujian dalam pembacaan RTC.



Gambar 4.14 Pengujian RTC

2019/5/11 10:18:25	2019/5/11 10:26:48	Awal = 4465	Akhir = 4968	hari ke - 503
2019/5/11 10:18:25	2019/5/11 10:26:49	Awal = 4465	Akhir = 4969	hari ke - 504
2019/5/11 10:18:25	2019/5/11 10:26:50	Awal = 4465	Akhir = 4970	hari ke - 505
2019/5/11 10:18:25	2019/5/11 10:26:51	Awal = 4465	Akhir = 4971	hari ke - 506
2019/5/11 10:18:25	2019/5/11 10:26:52	Awal = 4465	Akhir = 4972	hari ke - 507
2019/5/11 10:18:25	2019/5/11 10:26:53	Awal = 4465	Akhir = 4973	hari ke - 508
2019/5/11 10:18:25	2019/5/11 10:26:54	Awal = 4465	Akhir = 4974	hari ke - 509
2019/5/11 10:18:25	2019/5/11 10:26:55	Awal = 4465	Akhir = 4975	hari ke - 510
2019/5/11 10:18:25	2019/5/11 10:26:56	Awal = 4465	Akhir = 4976	hari ke - 511
2019/5/11 10:18:25	2019/5/11 10:26:57	Awal = 4465	Akhir = 4977	hari ke - 512
2019/5/11 10:18:25	2019/5/11 10:26:58	Awal = 4465	Akhir = 4978	hari ke - 513
2019/5/11 10:18:25	2019/5/11 10:26:59	Awal = 4465	Akhir = 4979	hari ke - 514
2019/5/11 10:18:25	2019/5/11 10:27:0	Awal = 4465	Akhir = 4980	hari ke - 515
2019/5/11 10:18:25	2019/5/11 10:27:1	Awal = 4465	Akhir = 4981	hari ke - 516
2019/5/11 10:18:25	2019/5/11 10:27:3	Awal = 4465	Akhir = 4983	hari ke - 518
2019/5/11 10:18:25	2019/5/11 10:27:4	Awal = 4465	Akhir = 4984	hari ke - 519

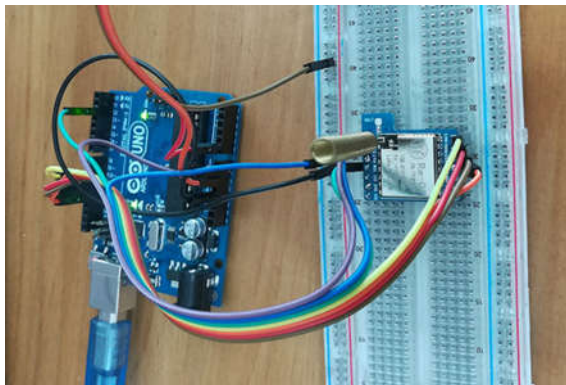
Gambar 4.15 Pembacaan RTC

Dalam pembacaan RTC di gambar 4.15 tersebut dihasilkan 5 data sesuai yang dijelaskan sebelumnya. Waktu awal yang ditampilkan berupa data tahun, bulan, tanggal, jam, menit, dan detik. Data awal yang ditetapkan di atas yaitu tanggal 11 bulan 5 tahun 2015 pada pukul 10 lebih 18 menit 25 detik. Data awal ini selalu tetap sampai berakhirnya operasi alat ini. Data kedua berupa waktu real time yang dilakukan oleh alat ini. Waktu yang ditunjukkan pada data 2 ini selalu berubah mengikuti waktu sekarang. Hal tersebut sesuai dengan yang ditampilkan pada gambar di atas. Pada data 3, didapatkan hasil perhitungan waktu awal/start alat beroperasi. Sesuai dengan data 1 yang tetap, perhitungan ini juga memiliki nilai yang tetap. Pada pengujian ini didapatkan hasil 4465 yang berasal dari $11 \times 86400 + 10 \times 3600 + 18 \times 60 + 25$. Sedangkan data 4 merupakan perhitungan dari waktu real time dari alat beroperasi. Hal tersebut bisa diambil contoh pada waktu 2019/5/11 10:26:48. Pada waktu tersebut didapatkan hasil 4968 dengan perhitungan $11 \times 86400 + 10 \times 3600 + 26 \times 60 + 48$. Sedangkan pada perhitungan hari didapatkan dari akhir – awal. Hal tersebut dapat dibuktikan pada waktu 2019/5/11 10:26:48 didapatkan hari ke 503 dengan perhitungan $4968 - 4465$.

4.3. Pengujian Pengiriman Data menggunakan SX1278

Pengujian SX1278 pada gambar 4.16 dilakukan dengan cara mengirimkan data dari pengirim dan penerima. Data yang dikirim berupa data kelembaban tanah A, B, C, dan D, serta hari. Pengujian SX1278 ini berguna untuk mendapatkan komunikasi data yang baik sebelum diuji pada alat sebenarnya. Komunikasi *boudrate* yang digunakan adalah 9600

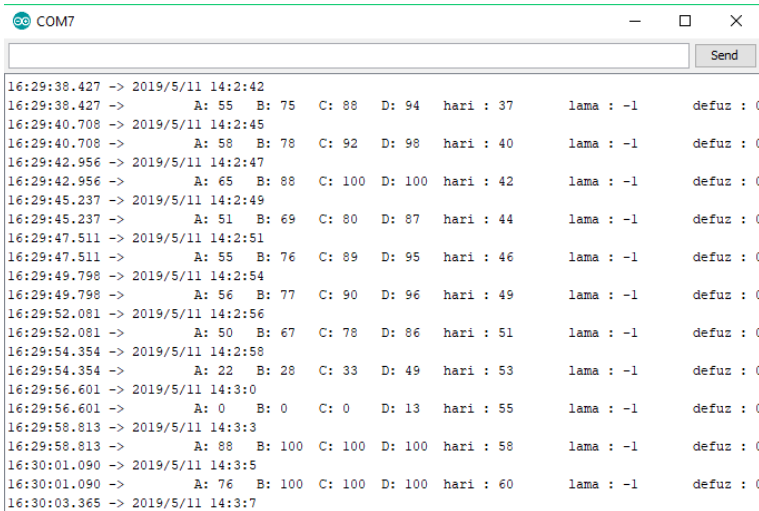
bps. Data yang dikirim bisa dilihat pada Gambar 4.17 dan data diterima pada Gambar 4.18 dibawah ini.



Gambar 4.16 Pengujian SX1278

```
COM6 (Arduino/Genuino Uno)
16:29:46.451 -> Initializing ... A: 55B: 76C: 89D: 95Hari: 46.
16:29:49.896 -> A: 56B: 77C: 90D: 96Hari: 49.
16:29:52.144 -> A: 50B: 67C: 78D: 86Hari: 51.
16:29:54.417 -> A: 22B: 29C: 33D: 49Hari: 53.
16:29:56.718 -> A: 0B: 0C: 0D: 13Hari: 55.
16:29:58.887 -> A: 88B: 100C: 100D: 100Hari: 58.
16:30:01.152 -> A: 76B: 100C: 100D: 100Hari: 60.
16:30:03.459 -> A: 37B: 49C: 57D: 69Hari: 62.
16:30:05.702 -> A: 33B: 45C: 53D: 65Hari: 65.
16:30:08.007 -> A: 16B: 21C: 25D: 42Hari: 67.
16:30:10.285 -> A: 4B: 7C: 10D: 31Hari: 69.
16:30:12.493 -> A: 0B: 0C: 0D: 18Hari: 71.
16:30:14.669 -> A: 89B: 100C: 100D: 100Hari: 73.
16:30:16.949 -> A: 79B: 100C: 100D: 100Hari: 76.
```

Gambar 4.17 Pengiriman Data SX1278

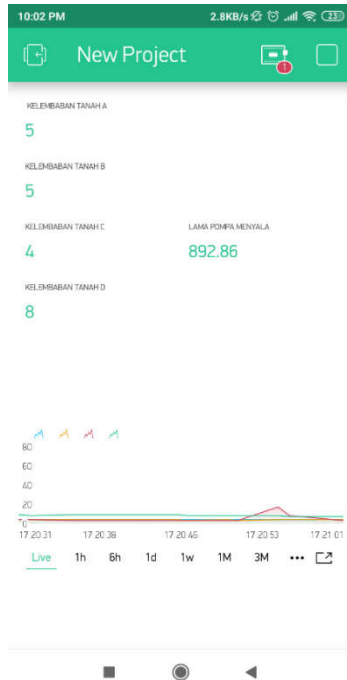


Gambar 4.18 Penerimaan Pengiriman Data SX1278

Pada gambar tersebut dapat menunjukkan bahwa data yang terima dan dikirim memiliki nilai yang sama. Tidak ada yang berubah dikarenakan data yang dikirm berupa string. Delay pengiriman data tersebut hanya sekitar 0.1 detik sehingga mendukung dalam pengiriman.

4.4. Pengujian Blynk

Data yang telah diterima oleh blok penerima SX1278 ditampilkan dalam aplikasi Blynk. Aplikasi berbasis IoT sehingga membutuhkan koneksi internet untuk mendapatkan datanya. Oleh karena itu, dalam penggunaannya menggunakan Nodemcu ESP8266 sebagai mikrokontrollernya. Data yang ditampilkan pada aplikasi ini berupa data Kelembaban Tanah A, Kelembaban Tanah B, Kelembaban Tanah C, Kelembaban Tanah D dan Lama waktu penyiraman. Berikut tampilan aplikasi Blynk yang digunakan pada gambar 4.20.



Gambar 4.20 Tampilan Blynk

4.5. Respon Kelembaban Tanah Terhadap Air

Pada tahap ini dilakukan pengukuran kelembaban tanah ketika diberikan air dengan takaran tertentu. Hal ini untuk mengetahui perubahan kelembaban tanah ketika diberikan air dengan volume tertentu. Hasil dari respon kelembaban tanah dan air akan digunakan untuk menentukan output dari fuzzy logic. Hal ini penting untuk mengetahui definisi output fuzzy logic. Seperti diketahui bahwa output fuzzy logic yaitu lama waktu pompa menyala. Anggota dari output ini adalah sedikit dan banyak. Berikut hasil dari pengukuran respon kelembaban tanah terhadap air ditampilkan pada tabel 4.6.

Tabel 4.6 Respon Kelembaban Tanah terhadap Air

Volume Air	Kelembaban Awal	Kelembaban Akhir	Selisih
10 ml	57	67	10
10 ml	74	84	10
10 ml	74	80	6
10 ml	44	54	10
20 ml	80	88	8
20 ml	38	74	36
20 ml	30	60	30
20 ml	44	74	30
30 ml	34	80	46
30 ml	22	70	48
30 ml	44	84	40

Dari hasil tersebut didapatkan bahwa ketika tanah diberikan air sebanyak 10 ml, maka penambahan kelembaban tanah yang terjadi antara 6 sampai 10 meskipun pada kelembaban tanah kering, sedang atau basah. Hal ini membuktikan bahwa pemanbahan air sebanyak 10 ml tidak mempengaruhi banyak tingkat kelembaban tanah. Kelembaban tanah cenderung tetap pada kondisinya masing – masing. Oleh karena itu, penambahan 10 ml tidak digunakan sebagai keluaran dari fuzzy logic.

Ketika tanah diberikan air sebanyak 20 ml, penambahan kelembaban tanah yang terjadi berkisar 8 sampai 36. Penambahan paling sedikit yaitu 8 ketika kelembaban tanah awal berada di tingkat kelembaban tanah basah. Penambahan yang sedikit dimungkinkan karena titik maksimal dari kelembaban tanah kurang lebih 88. Sedangkan pada kelembaban tanah kering dan sedang, penambahan kelembaban tanah tidak berbeda jauh antara 30 sampai 36. Penambahan air sebanyak 20 ml dapat mengubah tingkat kelembaban tanah dari kering menjadi sedang. Hal ini dibuktikan pada kelembaban awal 30 menjadi 60. Sedangkan ketika kelembaban awal berada pada kondisi sedang, maka kelembaban akhir berada pada kondisi basah. Hal tersebut dibuktikan dengan kelembaban awal senilai 44 menjadi 74.

Pada penambahan air sebanyak 30 ml, penambahan yang terjadi antara 40 sampai 48. Pemberian air sebanyak ini mengakibatkan kelembaban tanah akhir berada pada tingkat kelembaban tanah basah. Hal tersebut terjadi baik dari kondisi awal pada kelembaban tanah kering, sedang, maupun basah. Oleh karena itu, penambahan air 30 ml dapat

digunakan sebagai keluaran dari fuzzy logic.

Berdasarkan hasil tersebut, output fuzzy logic yang digunakan pada sistem ini adalah pemberian air 20 ml dan 30 ml. Pemberian air sebanyak 20 ml digunakan sebagai output fuzzy logic sedikit. Sedangkan pemberian air sebanyak 30 ml digunakan sebagai output fuzzy logic banyak.

4.6. Pengujian Fuzzy Logic

Setelah mengetahui tingkat kelembaban tanah dan responnya terhadap air, maka tahap selanjutnya adalah pengujian fuzzy logic. Dalam pengujian ini digunakan 3 set point. Masing masing tingkat kelembaban tanah memiliki set point yang berbeda. Untuk kelembaban tanah kering diberikan set point 30. Kelembaban tanah sedang memiliki set point sebesar 60. Sedangkan kelembaban tanah basah menggunakan set point 85. Pengujian fuzzy logic ini dilakukan dengan 3 set point yang berbeda.



Gambar 4.18 Pengujian Fuzzy Logic

4.6.1. Kelembaban Tanah Kering

Pada pengujian kelembaban tanah kering menggunakan set point 30. Berikut hasil dari pengujian yang ditampilkan pada tabel 4.7.

Tabel 4.7 Pengujian Fuzzy Logic pada Kelembaban Tanah Kering

Kelembaban Awal	Kelembaban Akhir	Selisih dengan Set Point
20	30	0
15	28	2
24	32	2
12	28	2
10	29	1
14	27	3
17	29	1

4.6.2. Kelembaban Tanah Sedang

Pada pengujian kelembaban tanah sedang, set point yang digunakan adalah 60. Berikut adalah hasil dari pengujian ini yang ditampilkan pada tabel 4.8.

Tabel 4.8 Pengujian Fuzzy Logic pada Kelembaban Tanah Sedang

Kelembaban Awal	Kelembaban Akhir	Selisih dengan Set Point
25	58	2
42	62	2
50	60	0
38	55	5
15	62	2
32	58	2
30	61	1

4.6.3. Kelembaban Tanah Basah

Pada pengujian kelembaban tanah basah, set point yang digunakan adalah 85. Berikut adalah hasil dari pengujiannya yang ditampilkan pada tabel 4.9.

Tabel 4.9 Pengujian Fuzzy Logic pada Kelembaban Tanah Basah

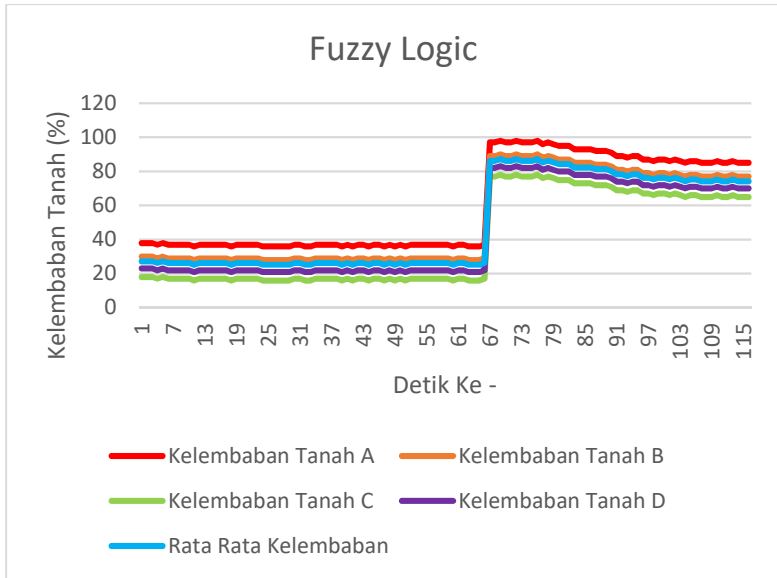
Kelembaban Awal	Kelembaban Akhir	Selisih dengan Set Point
15	84	1
24	85	0
30	81	4
55	85	0
68	87	2
73	83	2
75	85	0
80	88	3
81	88	3

4.7. Pengujian Alat di Tempat Sesungguhnya

Pengujian ini dilakukan di sawah daerah Keputih, Sukolilo, Surabaya dditunjukkan pada gambar 4.19. Aspek yang menjadi pengujian adalah respon kelembaban tanah dengan kontrol fuzzy. Tujuan pengujian ini untuk mendapatkan tingkat keberhasilan dalam pengontrolan kelembaban tanah di lahan sesungguhnya. Kasus yang dicoba pada pengujian adalah menjadikan kelembaban tanah bernilai 85% atau kondisi basah. Hasil yang didapatkan pada pengujian ini ditunjukkan dalam 4.20.



Gambar 4.19 Pengujian Alat di Tempat Sesungguhnya



Gambar 4.20 Hasil Fuzzy Logic di Lokasi Sesungguhnya

Dari pengujian tersebut didapatkan hasil bahwa kelembaban tanah akhir memiliki perbedaan dengan setpoint yang ditetapkan. Kelembaban awal dari pengujian ini adalah 30. Setelah dilakukan kontrol fuzzy, kelembaban rata rata akhir yang dihasilkan adalah 75. Selisih dengan setpoint adalah 10.

.....*Halaman ini sengaja dikosongkan*.....

BAB V PENUTUP

5.1. Kesimpulan

Pada tugas akhir ini, telah dibuat sebuah alat degan dimensi (15 x 11 x 5) cm. alat ini terdiri dari 2 bagian yaitu pengirim dan penerima. Pada bagian pengirim terdiri dari mikrokontroler Arduino Nano, 4 buah *capacitive soil moisture sensor*, RTC DS1307, LoRa SX1278, dan pompa 3 volt yang dihubungkan dengan relay. Sedangkan bagian penerima terdiri dari Arduino Uno, Nodemcu ESP8266, dan LoRa SX1278

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan yaitu Sistem Monitoring dan Controlling Kelembaban Tanah pada Irigasi Tetes dengan LoRa dapat bekerja dengan baik. Hal ini dibuktikan dengan hasil control kelembaban tanah menggunakan fuzzy logic dapat mencapai setpoint yang diinginkan dengan error 1 sampai 5 %. Data kelembaban tanah juga dapat dipantau dari jarak jauh menggunakan LoRa dengan jarak maksimum pada padat penduduk yaitu 100 meter dan tidak ada halangan sejauh 500 meter. Selain itu, data juga dapat dipantau menggunakan aplikasi Blynk dari smartphone yang membuktikan alat ini berbasis IoT.

5.2. Saran

Sebagai sarana pengembangan Sistem Monitoring dan Controlling Kelembaban Tanah pada Irigasi Tetes dengan LoRa ini, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan. Pencapaian set point dari kelembaban tanah terkadang membutuhkan beberapa kali penyiraman, maka lama pompa menyala diatur ulang agar sekali penyiraman bisa langsung mencapai set point. Penggunaan wifi dengan kecepatan stabil. Memperbanyak jumlah sensor kelembaban tanah sehingga lebih banyak cakupan dari monitoring dan controlling. Selain itu, lebih memperhatikan faktor estetika dari Sistem Monitoring dan Kontroling Kelembaban Tanah pada Irigasi Tetes dengan LoRa.

.....*Halaman ini sengaja dikosongkan*.....

DAFTAR PUSTAKA

- [1] N. Jin *et al.*, “Effects of water stress on water use efficiency of irrigated and rainfed wheat in the Loess Plateau, China,” *Sci. Total Environ.*, vol. 642, pp. 1–11, Nov. 2018.
- [2] R. K. Rai, V. P. Singh, and A. Upadhyay, “Irrigation Methods,” in *Planning and Evaluation of Irrigation Projects*, Elsevier, 2017, pp. 353–363.
- [3] A.-M. O. Mohamed and E. K. Paleologos, “Chapter 17 - Advances in the Determination of Soil Moisture Content,” in *Fundamentals of Geoenvironmental Engineering*, A.-M. O. Mohamed and E. K. Paleologos, Eds. Butterworth-Heinemann, 2018, pp. 639–680.
- [4] A.-M. O. Mohamed and E. K. Paleologos, “Soil-Water Interaction,” Elsevier, 2018, pp. 161–203.
- [5] G. Hochmuth, “5 - Drip Irrigation,” in *A Guide to the Manufacture, Performance, and Potential of Plastics in Agriculture*, M. D. Orzolek, Ed. Elsevier, 2017, pp. 79–105.
- [6] E. Sathyapriya, M. R. Naveenkumar, and V. Dhivya, “AN EMPIRICAL STUDY ON DRIP IRRIGATION,” presented at the National conference on Micro Irrigation, TNAU, Coimbatore, 2017.
- [7] S. Mechram, P. Satriyo, and E. Mutia, “Pengaruh Jumlah Emitter Terhadap Debit Emitter dan Koefisien Keseragaman Irigasi Tetes Sistem Gravitasi,” *Rona Tek. Pertnian*, vol. 3, no. 1, 2011.
- [8] R. Bansal *et al.*, “On-Farm Drip Irrigation in Rice for Higher Productivity and Profitability in Haryana, India,” *Int. J. Curr. Microbiol. Appl. Sci.*, vol. 7, no. 2, pp. 506–512, 2018.
- [9] C. Arif, B. I. Setiawan, and M. Mizoguchi, “Penentuan kelembaban tanah optimum untuk budidaya padi sawah sri (system of rice intensification) menggunakan algoritma genetika,” *Determining optimal soil moisture for system of rice intensification paddy field using genetic algorithms*, Oct. 2014.
- [10] G. Chavarria and H. Pessoa dos Santos, “Plant Water Relations: Absorption, Transport and Control Mechanisms,” 2012.
- [11] D. Tubagus and H. A. Nugroho, “RANCANG BANGUN SISTEM PENGUKUR SUHU DAN KELEMBABAN TANAH BERBASIS KOMUNIKASI RADIO,” *J. Sains Dan Teknol.*, vol. 7, no. 1, Apr. 2018.
- [12] M. Rivai, Rendyansyah, and D. Purwanto, *Implementation of fuzzy logic control in robot arm for searching location of gas leak*. 2015.
- [13] Y. E. M. Hamouda, “Smart Irrigation Decision Support Based on

- Fuzzy Logic Using Wireless Sensor Network,” in *2017 International Conference on Promising Electronic Technologies (ICPET)*, Deir El-Balah, 2017, pp. 109–113.
- [14] M. Rivai, K. Sambodho, and D. Purnomo, *Identification of Levee Strengh for Early Warning System Using Fuzzy Logic*. 2014.
- [15] A. Åkesson, M. Nordahl, G. Hedin, and B. Magnusson, *A DSL for composing IoT systems*. 2018.
- [16] S. B. Saraf and D. H. Gawali, “IoT based smart irrigation monitoring and controlling system,” in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, 2017, pp. 815–819.
- [17] S. Rizvi, A. Kurtz, J. Pfeffer, and M. Rizvi, *Securing the Internet of Things (IoT): A Security Taxonomy for IoT*. 2018.
- [18] M. Dzulkifli S., M. Rivai, and S. Suwito, *Rancang Bangun Sistem Irigasi Tanaman Otomatis Menggunakan Wireless Sensor Network*, vol. 5. 2016.
- [19] Radi, M. Murtiningrum, Ngadisih, F. Muzdrikah, M. Shohibun Nuha, and F. Alfinur Rizqi, *Calibration of Capacitive Soil Moisture Sensor (SKU:SEN0193)*. 2018.
- [20] G. Mander and M. Arora, “Design of capacitive sensor for monitoring moisture content of soil and analysis of analog voltage with variability in moisture.” 2014.
- [21] M. Bor, J. Vidler, and U. Roedig, *LoRa for the Internet of Things*. 2016.
- [22] A. Lavric and V. Popa, “Internet of Things and LoRa™ Low-Power Wide-Area Networks: A survey,” in *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*, Iasi, Romania, 2017, pp. 1–5.
- [23] S. Daud, T. Shi Yang, R. Muhamad Asmi, Z. Awang Ahmad, N. Mahrom, and R. Aliana A. Raof, *Performance Evaluation of Low Cost LoRa Modules in IoT Applications*, vol. 318. 2018.
- [24] “<https://core-electronics.com.au/mute-sounds-mini-submersible-pump-dc-3v-5v.html>.” .
- [25] M. H. Assiddiqi, *Rancang Bangun Robot Pembersih Lantai Kolam Renang*. surabaya, 2017.
- [26] “What is Arduino?,” *2017 Arduino*, 2017. [Online]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Accessed: 16-May-2018].

- [27] M. P. P. Jesani, T. J. Raval, and K. A. Chaudhary, "A Review On IoT Based Smart Home Using Blynk Framework," vol. 3, no. 5, p. 9, 2017.
- [28] N. A. Z. M. Noar and M. M. Kamal, "The development of smart flood monitoring system using ultrasonic sensor with blynk applications," in *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, Putrajaya, 2017, pp. 1–6.
- [29] M. Orzolek (Eds.), *A Guide to the Manufacture, Performance, and Potential of Plastics in Agriculture. A volume in Plastics Design Library*, 1st Edition. William Andrew, 2017.
- [30] <https://www.javamas.com/kelembaban-tanah-ideal/> diakses pada 22 April 2019

.....*Halaman ini sengaja dikosongkan*.....

LAMPIRAN A

Program Pengirim

```
#include <Wire.h>
#include <LoRaLib.h>
#include <LiquidCrystal_I2C.h>
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>
#include "RTClib.h"
RTC_DS1307 RTC;

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 20, 4);

///Setting pin input moisture sensor
int hygrometer1 = A0; //Hygrometer sensor analog pin output at
pin A0 of Arduino
int hygrometer2 = A1; //Hygrometer sensor analog pin output at
pin A1 of Arduino
int hygrometer3 = A2; //Hygrometer sensor analog pin output at
pin A2 of Arduino
int hygrometer4 = A3; //Hygrometer sensor analog pin output at
pin A3 of Arduino

//Setting output nilai sensor
float error,derror,last_error,pwm,output1;
Fuzzy* fuzzy = new Fuzzy();
int value1,value2,value3,value4,hari,kelembaban_total,awal,
akhir;
String value11,value21,value31,value41,hari_str;
//Range Pengukuran Sensor
int range_max = 615;
int range_min = 285;
FuzzySet* no = new FuzzySet(0, 0, 0, 0);
FuzzySet* sedikit = new FuzzySet(0, 10, 10, 20);
FuzzySet* sedang = new FuzzySet(10, 20, 20, 30);
FuzzySet* banyak = new FuzzySet(20, 30, 30, 35);
SX1278 lora = new LoRa;
void setup()
{
  Serial.begin(9600);
```

```

Wire.begin();
pinMode(4,OUTPUT);
digitalWrite(4,LOW);
lcd.init();
// initialize the lcd
lcd.init();
lcd.backlight();
// Print a message to the LCD.
RTC.begin();
if (! RTC.isrunning()) {
  Serial.println("RTC is NOT running!");
  // following line sets the RTC to the date & time this sketch
  was compiled
  RTC.adjust(DateTime(__DATE__, __TIME__));
}
DateTime now = RTC.now();
awal = now.day()*86400+now.hour()*3600 + now.minute()* 60 +
now.second();
Serial.print(F("Initializing ... "));
int state = lora.begin();
  if (state == ERR_NONE) {
    Serial.println(F("success!"));
  } else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
  }
}

void BacaKelembaban()
{
  //   DateTime now = RTC.now();
  //   awal = now.minute()* 60 + now.second();
  //   akhir = now.minute()* 60 + now.second();
  hari = 20;
  value1 = analogRead(hygrometer1); //Read analog value
  value1 = constrain(value1,range_min,range_max); //Keep the
ranges!
  value1 = map(value1,range_min,range_max,100,0); //Map
value : 0 will be 1023 and 1023 will be 0
  value2 = analogRead(hygrometer2); //Read analog value
  value2 = constrain(value2,range_min,range_max); //Keep the
ranges!
  value2 = map(value2,range_min,range_max,100,0); //Map
value : 0 will be 1023 and 1023 will be 0
  value3 = analogRead(hygrometer3); //Read analog value
  value3 = constrain(value3,range_min,range_max); //Keep the
ranges!
}

```

```

    value3 = map(value3,range_min,range_max,100,0); //Map
value : 0 will be 1023 and 1023 will be 0
    value4 = analogRead(hygrometer4); //Read analog value
    value4 = constrain(value4,range_min,range_max); //Keep the
ranges!
    value4 = map(value4,range_min,range_max,100,0); //Map
value : 0 will be 1023 and 1023 will be 0
    kelembaban_total = (value1+value2+value3+value4)/4;
}

```

```

String hasil;
void Display ()
{
    String A = "A : ";
    String B = "\tB : ";
    String C = "\tC : ";
    String D = "\tD : ";
    String hari_str = "\tHari : ";
    String lama_str = "\tLama : ";
    String kelembaban_total_str = "\tKelembaban Total : ";
    pwm = output1*100;
    hasil =
A+value1+B+value2+C+value3+D+value4+hari_str+hari+lama_str+pwm
+kelembaban_total_str+kelembaban_total;
    lcd.setCursor(12, 1);
    lcd.print(value4);
}
#include "fuzzy_kering.h"
#include "fuzzy_sedang.h"
#include "fuzzy_basah.h"
void loop()
{
    //lcd.setCursor(0, 0);
    // Time();
    // inisialisasi_sedang();

    if(hari<=30)
    {
        setup_kering();
        display_kering();
    }
    if(hari>30&&hari<=60)
    {
        setup_sedang();
        display_sedang();
    }
    if(hari>60&&hari<=90)
    {

```

```

        setup_basah();
        display_basah();
    }
    delay(1000); //Read every 1 sec.
}

```

```
float setpoint3=85;
```

```

FuzzySet* error6 = new FuzzySet(-15, -15, -10, 10);
FuzzySet* error7 = new FuzzySet(-10, 10, 10, 30);
FuzzySet* error8 = new FuzzySet(10, 30, 30, 50);
FuzzySet* error9 = new FuzzySet(40, 60, 85, 85);
//Input Derror
FuzzySet* derror6 = new FuzzySet(-20, -20, -1, 3);
FuzzySet* derror7 = new FuzzySet(-1, 3, 3, 7);
FuzzySet* derror8 = new FuzzySet(5, 7, 30, 30);
FuzzySet* derror9 = new FuzzySet(5, 7, 30, 30);

```

```

void setup_basah(){
    // Membuat fuzzyinput
    //Fuzzy Input error
    FuzzyInput* inputerror = new FuzzyInput(1);
    inputerror->addFuzzySet(error6);
    inputerror->addFuzzySet(error7);
    inputerror->addFuzzySet(error8);
    inputerror->addFuzzySet(error9);
    fuzzy->addFuzzyInput(inputerror);
    //Fuzzy Input Derror
    FuzzyInput* inputDerror = new FuzzyInput(2);
    inputDerror->addFuzzySet(derror6);
    inputDerror->addFuzzySet(derror7);
    inputDerror->addFuzzySet(derror8);
    inputDerror->addFuzzySet(derror9);
    fuzzy->addFuzzyInput(inputDerror);
    //Step 4. Membuat Fuzzy Rule
    //1. IF error=error6 and Derror=derror6 THEN velocity=stop
    FuzzyRuleAntecedent* iferrorerror6AndDerrorderror6 = new
FuzzyRuleAntecedent();
    iferrorerror6AndDerrorderror6->joinWithAND(error6, derror6);
}

```



```

        FuzzyRuleConsequent*   thenVelocityStop14   =   new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
        thenVelocityStop14->addOutput(no); // Adding corresponding
FuzzySet to Consequent object
        // Instantiating a FuzzyRule object
        FuzzyRule*   fuzzyRule14   =   new   FuzzyRule(14,
iferrorerror6AndDerrorerror6, thenVelocityStop14); // Passing the
Antecedent and the Consequent of expression
        fuzzy->addFuzzyRule(fuzzyRule14); // Adding FuzzyRule to
Fuzzy object

```

```

        //2. IF error=N and Derror=negkec THEN velocity=stop
        FuzzyRuleAntecedent*   iferrorerror7AndDerrorerror6   =   new
FuzzyRuleAntecedent();
        iferrorerror7AndDerrorerror6->joinWithAND(error7, derror6);
        FuzzyRuleConsequent*   thenVelocityStop15   =   new
FuzzyRuleConsequent();
        thenVelocityStop15->addOutput(no);
        // Instantiating a FuzzyRule object
        FuzzyRule*   fuzzyRule15   =   new   FuzzyRule(15,
iferrorerror7AndDerrorerror6, thenVelocityStop15);
        fuzzy->addFuzzyRule(fuzzyRule15);

```

```

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent*   iferrorerror8AndDerrorerror6   =   new
FuzzyRuleAntecedent();
        iferrorerror8AndDerrorerror6->joinWithAND(error8, derror6);
        FuzzyRuleConsequent*   thenVelocitySlow16   =   new
FuzzyRuleConsequent();
        thenVelocitySlow16->addOutput(sedikit);
        // Instantiating a FuzzyRule object
        FuzzyRule*   fuzzyRule16   =   new   FuzzyRule(16,
iferrorerror8AndDerrorerror6, thenVelocitySlow16);
        fuzzy->addFuzzyRule(fuzzyRule16);

```

```

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent*   iferrorerror9AndDerrorerror6   =   new
FuzzyRuleAntecedent();
        iferrorerror9AndDerrorerror6->joinWithAND(error9, derror6);

```

```

    FuzzyRuleConsequent*   thenVelocitySlow17   =   new
FuzzyRuleConsequent();
    thenVelocitySlow17->addOutput(sedikit);
    // Instantiating a FuzzyRule object
    FuzzyRule*   fuzzyRule17   =   new   FuzzyRule(17,
iferrorerror8AndDerrorderror6, thenVelocitySlow17);
    fuzzy->addFuzzyRule(fuzzyRule17);

    //1. IF error=NK and Derror=negkec THEN velocity=stop
    FuzzyRuleAntecedent*   iferrorerror6AndDerrorderror7 = new
FuzzyRuleAntecedent();
    iferrorerror6AndDerrorderror6->joinWithAND(error6, derror7);
    FuzzyRuleConsequent*   thenVelocityStop18   =   new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
    thenVelocityStop18->addOutput(no); // Adding corresponding
FuzzySet to Consequent object
    // Instantiating a FuzzyRule object
    FuzzyRule*   fuzzyRule18   =   new   FuzzyRule(18,
iferrorerror6AndDerrorderror7, thenVelocityStop18); // Passing the
Antecedent and the Consequent of expression
    fuzzy->addFuzzyRule(fuzzyRule18); // Adding FuzzyRule to
Fuzzy object

    //2. IF error=N and Derror=negkec THEN velocity=stop
    FuzzyRuleAntecedent*   iferrorerror7AndDerrorderror7 = new
FuzzyRuleAntecedent();
    iferrorerror7AndDerrorderror7->joinWithAND(error7, derror7);
    FuzzyRuleConsequent*   thenVelocityStop19   =   new
FuzzyRuleConsequent();
    thenVelocityStop19->addOutput(sedikit);
    // Instantiating a FuzzyRule object
    FuzzyRule*   fuzzyRule19   =   new   FuzzyRule(19,
iferrorerror7AndDerrorderror7, thenVelocityStop19);
    fuzzy->addFuzzyRule(fuzzyRule19);

    //3. IF error=Z and Derror=negkec THEN velocity=slow
    FuzzyRuleAntecedent*   iferrorerror8AndDerrorderror7 = new
FuzzyRuleAntecedent();
    iferrorerror8AndDerrorderror7->joinWithAND(error8, derror7);

```

```

        FuzzyRuleConsequent*    thenVelocitySlow20    =    new
FuzzyRuleConsequent();
        thenVelocitySlow20->addOutput(sedang);
        // Instantiating a FuzzyRule object
        FuzzyRule*    fuzzyRule20    =    new    FuzzyRule(20,
iferrorerror8AndDerrorerror7, thenVelocitySlow20);
        fuzzy->addFuzzyRule(fuzzyRule20);

```

```

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent*    iferrorerror9AndDerrorerror7    =    new
FuzzyRuleAntecedent();
        iferrorerror9AndDerrorerror7->joinWithAND(error9, derror7);
        FuzzyRuleConsequent*    thenVelocitySlow21    =    new
FuzzyRuleConsequent();
        thenVelocitySlow21->addOutput(sedang);
        // Instantiating a FuzzyRule object
        FuzzyRule*    fuzzyRule21    =    new    FuzzyRule(21,
iferrorerror8AndDerrorerror7, thenVelocitySlow21);
        fuzzy->addFuzzyRule(fuzzyRule21);

```

```

        //1. IF error=NK and Derror=negkec THEN velocity=stop
        FuzzyRuleAntecedent*    iferrorerror6AndDerrorerror8    =    new
FuzzyRuleAntecedent();
        iferrorerror6AndDerrorerror8->joinWithAND(error6, derror8);
        FuzzyRuleConsequent*    thenVelocityStop22    =    new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression
        thenVelocityStop22->addOutput(no);// Adding corresponding
FuzzySet to Consequent object
        // Instantiating a FuzzyRule object
        FuzzyRule*    fuzzyRule22    =    new    FuzzyRule(22,
iferrorerror6AndDerrorerror8, thenVelocityStop22); // Passing the
Antecedent and the Consequent of expression
        fuzzy->addFuzzyRule(fuzzyRule22); // Adding FuzzyRule to
Fuzzy object

```

```

        //2. IF error=N and Derror=negkec THEN velocity=stop
        FuzzyRuleAntecedent*    iferrorerror7AndDerrorerror8    =    new
FuzzyRuleAntecedent();
        iferrorerror7AndDerrorerror8->joinWithAND(error7, derror8);

```

```

        FuzzyRuleConsequent*   thenVelocityStop23   =   new
FuzzyRuleConsequent();
        thenVelocityStop23->addOutput(sedikit);
        // Instantiating a FuzzyRule object
        FuzzyRule*   fuzzyRule23   =   new   FuzzyRule(23,
iferrorerror7AndDerrorderror8, thenVelocityStop23);
        fuzzy->addFuzzyRule(fuzzyRule23);

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent*   iferrorerror8AndDerrorderror8   =   new
FuzzyRuleAntecedent();
        iferrorerror8AndDerrorderror8->joinWithAND(error8, derror8);
        FuzzyRuleConsequent*   thenVelocitySlow24   =   new
FuzzyRuleConsequent();
        thenVelocitySlow24->addOutput(sedang);
        // Instantiating a FuzzyRule object
        FuzzyRule*   fuzzyRule24   =   new   FuzzyRule(24,
iferrorerror8AndDerrorderror8, thenVelocitySlow24);
        fuzzy->addFuzzyRule(fuzzyRule24);

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent*   iferrorerror9AndDerrorderror8   =   new
FuzzyRuleAntecedent();
        iferrorerror9AndDerrorderror8->joinWithAND(error9, derror8);
        FuzzyRuleConsequent*   thenVelocitySlow25   =   new
FuzzyRuleConsequent();
        thenVelocitySlow25->addOutput(banyak);
        // Instantiating a FuzzyRule object
        FuzzyRule*   fuzzyRule25   =   new   FuzzyRule(25,
iferrorerror8AndDerrorderror8, thenVelocitySlow25);
        fuzzy->addFuzzyRule(fuzzyRule25);

        //1. IF error=NK and Derror=negkec THEN velocity=stop
        FuzzyRuleAntecedent*   iferrorerror6AndDerrorderror9   =   new
FuzzyRuleAntecedent();
        iferrorerror6AndDerrorderror9->joinWithAND(error6, derror9);
        FuzzyRuleConsequent*   thenVelocityStop26   =   new
FuzzyRuleConsequent(); // Instantiating a Consequent to expression

```

```

        thenVelocityStop26->addOutput(sedang);//          Adding
corresponding FuzzySet to Consequent object
        // Instantiating a FuzzyRule object
        FuzzyRule*    fuzzyRule26    =    new    FuzzyRule(26,
iferrorerror6AndDerrorerror9, thenVelocityStop26); // Passing the
Antecedent and the Consequent of expression
        fuzzy->addFuzzyRule(fuzzyRule26); // Adding FuzzyRule to
Fuzzy object

        //2. IF error=N and Derror=negkec THEN velocity=stop
        FuzzyRuleAntecedent* iferrorerror7AndDerrorerror9 = new
FuzzyRuleAntecedent();
        iferrorerror7AndDerrorerror9->joinWithAND(error7, derror9);
        FuzzyRuleConsequent*    thenVelocityStop27    =    new
FuzzyRuleConsequent();
        thenVelocityStop27->addOutput(banyak);
        // Instantiating a FuzzyRule object
        FuzzyRule*    fuzzyRule27    =    new    FuzzyRule(27,
iferrorerror7AndDerrorerror9, thenVelocityStop27);
        fuzzy->addFuzzyRule(fuzzyRule27);

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent* iferrorerror8AndDerrorerror9 = new
FuzzyRuleAntecedent();
        iferrorerror8AndDerrorerror9->joinWithAND(error8, derror9);
        FuzzyRuleConsequent*    thenVelocitySlow28    =    new
FuzzyRuleConsequent();
        thenVelocitySlow28->addOutput(banyak);
        // Instantiating a FuzzyRule object
        FuzzyRule*    fuzzyRule28    =    new    FuzzyRule(28,
iferrorerror8AndDerrorerror9, thenVelocitySlow28);
        fuzzy->addFuzzyRule(fuzzyRule28);

        //3. IF error=Z and Derror=negkec THEN velocity=slow
        FuzzyRuleAntecedent* iferrorerror9AndDerrorerror9 = new
FuzzyRuleAntecedent();
        iferrorerror9AndDerrorerror9->joinWithAND(error9, derror9);
        FuzzyRuleConsequent*    thenVelocitySlow29    =    new
FuzzyRuleConsequent();

```

```

    thenVelocitySlow29->addOutput(banyak);
    // Instantiating a FuzzyRule object
    FuzzyRule*    fuzzyRule29    =    new    FuzzyRule(29,
iferrorerror8AndDerrorderror9, thenVelocitySlow29);
    fuzzy->addFuzzyRule(fuzzyRule29);
}
void display_basah(){

    /*Controlled with fuzzy logic controller and gas sensors*/
    last_error = error;
    error = setpoint3 - value1 ;
    derror = error - last_error;

    Serial.println(error);
    Serial.println(derror);

    fuzzy->setInput(5, error);
    fuzzy->setInput(6, derror);
    fuzzy->fuzzify();
    //Tampilkan data Error
    Serial.print("Error: ");
    Serial.print(error6->getPertinence());    Serial.print(", ");
    Serial.print(error7->getPertinence());    Serial.print(", ");
    Serial.println(error8->getPertinence());    Serial.print(", ");
    //Tampilkan data Derror
    Serial.print("Derror: ");
    Serial.print(derror6->getPertinence());Serial.print(", ");
    Serial.print(derror7->getPertinence());    Serial.print(", ");
    Serial.println(derror8->getPertinence());    Serial.print(", ");
    output1 = fuzzy->defuzzify(3);
    Serial.print("KELUARAN          AVERAGE:          ");
Serial.println(output1);
    Serial.print(hasil);
    lora.transmit(hasil);
    Serial.print('\n');
    digitalWrite(4,LOW);
    delay(output1*100);
    digitalWrite(4,HIGH);
}

```

LAMPIRAN B

Program Penerima

```
#include <LoRaLib.h>

// create instance of LoRa class using SX1278 module
// this pinout corresponds to RadioShield
// https://github.com/jgromes/RadioShield
// NSS pin: 10 (4 on ESP32/ESP8266 boards)
// DI00 pin: 2
// DI01 pin: 3
SX1278 lora = new LoRa;

void setup() {
  Serial.begin(9600);
  Serial.print(char(50));
  // initialize SX1278 with default settings
  Serial.print(F("Initializing ... "));
  // carrier frequency:      434.0 MHz
  // bandwidth:              125.0 kHz
  // spreading factor:       9
  // coding rate:            7
  // sync word:              0x12
  // output power:           17 dBm
  // current limit:          100 mA
  // preamble length:        8 symbols
  // amplifier gain:         0 (automatic gain control)
  int state = lora.begin();
  if (state == ERR_NONE) {
  } else {
    Serial.print(F("failed, code "));
    Serial.println(state);
    while (true);
  }
}

void loop() {

  // you can receive data as an Arduino String
  String str, str1;
  int state = lora.receive(str);

  // you can also receive data as byte array
  /*
  byte byteArr[8];
  int state = lora.receive(byteArr, 8);
  */
}
```

```

*/
// lora.standby();
if (state == ERR_NONE) {
    // packet was successfully received

    // print data of the packet

    //Serial.println(str);
    Serial.print(str);
    Serial.print("\n");

    // print RSSI (Received Signal Strength Indicator)
    // of the last received packet
    //Serial.print(lora.getRSSI());

    // print SNR (Signal-to-Noise Ratio)
    // of the last received packet
    //Serial.print(lora.getSNR());

    // print frequency error
    // of the last received packet
    //Serial.print(lora.getFrequencyError());

} else if (state == ERR_RX_TIMEOUT) {
    // timeout occurred while waiting for a packet

} else if (state == ERR_CRC_MISMATCH) {
    // CCpacket was received, but is malformed
}
}
}

```


LAMPIRAN C

PROGRAM BLYNK

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "371f4b1a4ac844b2845956990dd2af23";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Sholeh";
char pass[] = "12345678";
BlynkTimer timer;
int value1[127];
int value2[127];
int value3[127];
int value4[127];
#include "data.h"

int kelembaban_total;
int pwm;
String kriteria,masa,kelembaban;
void setup() {
  Serial.begin(9600);
  Wire.begin();
  data();
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 8442);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100),
8442);

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}
void sendSensor()
{
  // float af = Float(a);
  // float bf = Float(b); // or dht.readTemperature(true) for
Fahrenheit
```

```

// float cf = Float(c);
// float df = Float(d); // or dht.readTemperature(true) for
Fahrenheit
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V6, nilai_A);
Blynk.virtualWrite(V7, nilai_B);
Blynk.virtualWrite(V8, nilai_C);
Blynk.virtualWrite(V9, nilai_D);
Blynk.virtualWrite(V10, nilai_Lama);
Blynk.virtualWrite(V11, kelembaban);
Blynk.virtualWrite(V12, masa);
}
void loop()
{
for ( i = 65;i<=128;i++)
{
Blynk.run();
timer.run();

kelembaban_total=(value1[i]+value2[i]+value3[i]+value4[i])/4;
if (kelembaban_total <=40)
{
kriteria=" kering";
pwm=2825;
}
if (kelembaban_total > 40 && kelembaban_total<=75)
{
kriteria=" sedang";
pwm=000;
}
if (kelembaban_total > 75 && kelembaban_total <=85)
{
kriteria=" basah";
pwm=000;
}
if (kelembaban_total > 85 && kelembaban_total <=100)
{
kriteria=" basah";
pwm=0000;
}
kelembaban=kelembaban_total+" "+kriteria;
masa="krng";
}
}

```

LAMPIRAN D

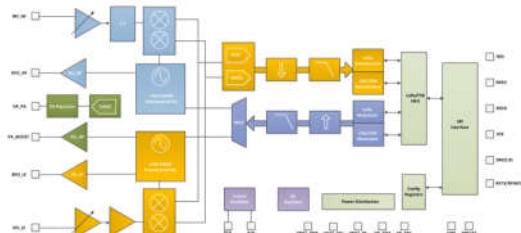
DATASHEET SX1278



SX1276/77/78/79

WIRELESS, SENSING & TIMING DATASHEET

SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver



GENERAL DESCRIPTION

The SX1276/77/78/79 transceivers feature the LoRa™ long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption.

Using Semtech's patented LoRa™ modulation technique SX1276/77/78/79 can achieve a sensitivity of over -148dBm using a low cost crystal and bill of materials. The high sensitivity combined with the integrated +20 dBm power amplifier yields industry leading link budget making it optimal for any application requiring range or robustness. LoRa™ also provides significant advantages in both blocking and selectivity over conventional modulation techniques, solving the traditional design compromise between range, interference immunity and energy consumption.

These devices also support high performance (G)FSK modes for systems including WMBus, IEEE802.15.4g. The SX1276/77/78/79 deliver exceptional phase noise, selectivity, receiver linearity and IIP3 for significantly lower current consumption than competing devices.

ORDERING INFORMATION

Part Number	Delivery	MOQ / Multiple
SX1276IMLTRT	T&R	3000 pieces
SX1277IMLTRT	T&R	3000 pieces
SX1278IMLTRT	T&R	3000 pieces
SX1279IMLTRT	T&R	3000 pieces

- QFN 28 Package - Operating Range [-40,+85°C]
- Pb-free, Halogen free, RoHS/WEEE compliant product

KEY PRODUCT FEATURES

- LoRa™ Modem
- 168 dB maximum link budget
- +20 dBm - 100 mW constant RF output vs. V supply
- +14 dBm high efficiency PA
- Programmable bit rate up to 300 kbps
- High sensitivity: down to -148 dBm
- Bullet-proof front end: IIP3 = -11 dBm
- Excellent blocking immunity
- Low RX current of 9.9 mA, 200 nA register retention
- Fully integrated synthesizer with a resolution of 61 Hz
- FSK, GFSK, MSK, GMSK, LoRa™ and OOK modulation
- Built-in bit synchronizer for clock recovery
- Preamble detection
- 127 dB Dynamic Range RSSI
- Automatic RF Sense and CAD with ultra-fast AFC
- Packet engine up to 256 bytes with CRC
- Built-in temperature sensor and low battery indicator

APPLICATIONS

- Automated Meter Reading
- Home and Building Automation
- Wireless Alarm and Security Systems
- Industrial Monitoring and Control
- Long range Irrigation Systems

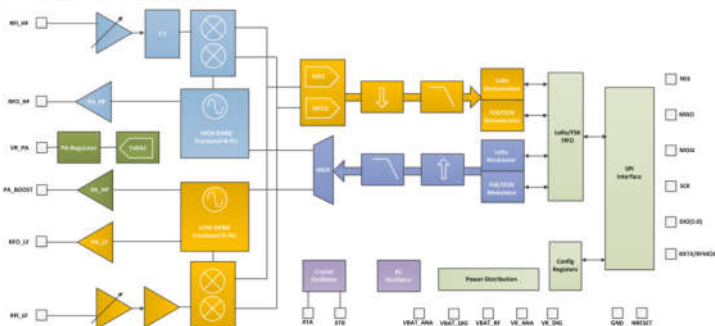
1. General Description

The SX1276/77/78/79 incorporates the LoRa™ spread spectrum modem which is capable of achieving significantly longer range than existing systems based on FSK or OOK modulation. At maximum data rates of LoRa™ the sensitivity is 8dB better than FSK, but using a low cost bill of materials with a 20ppm XTAL LoRa™ can improve receiver sensitivity by more than 20dB compared to FSK. LoRa™ also provides significant advances in selectivity and blocking performance, further improving communication reliability. For maximum flexibility the user may decide on the spread spectrum modulation bandwidth (BW), spreading factor (SF) and error correction rate (CR). Another benefit of the spread modulation is that each spreading factor is orthogonal - thus multiple transmitted signals can occupy the same channel without interfering. This also permits simple coexistence with existing FSK based systems. Standard GFSK, FSK, OOK, and GMSK modulation is also provided to allow compatibility with existing systems or standards such as wireless MBUS and IEEE 802.15.4g.

The SX1276 and SX1279 offer bandwidth options ranging from 7.8 kHz to 500 kHz with spreading factors ranging from 6 to 12, and covering all available frequency bands. The SX1277 offers the same bandwidth and frequency band options with spreading factors from 6 to 9. The SX1278 offers bandwidths and spreading factor options, but only covers the lower UHF bands.

The SX1276 and SX1279 offer bandwidth options ranging from 7.8 kHz to 500 kHz with spreading factors ranging from 6 to 12, and covering all available frequency bands. The SX1277 offers the same bandwidth and frequency band options with spreading factors from 6 to 9. The SX1278 offers bandwidths and spreading factor options, but only covers the lower UHF bands.

1.1. Simplified Block Diagram



1.2. Product Versions

The features of the four product variants are detailed in the following table.

Table 1 SX1276/77/78/79 Device Variants and Key Parameters

Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
SX1276	137 - 1020 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1277	137 - 1020 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
SX1278	137 - 525 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
SX1279	137 - 960MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

1.3. Pin Diagram

The following diagram shows the pin arrangement of the QFN package, top view.

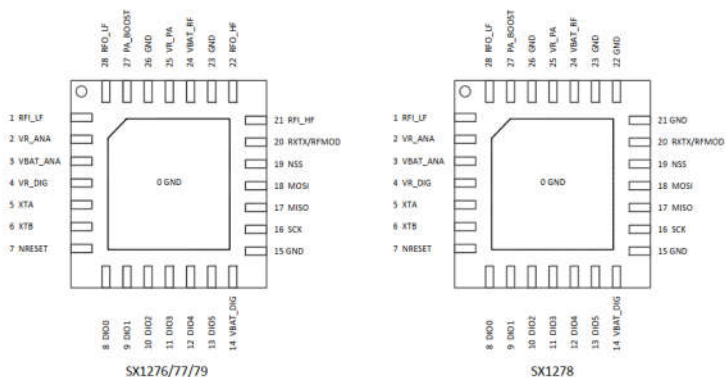


Figure 2. Pin Diagrams

4. SX1276/77/78/79 Digital Electronics

4.1. The LoRa™ Modem

The LoRa™ modem uses spread spectrum modulation and forward error correction techniques to increase the range and robustness of radio communication links compared to traditional FSK or OOK based modulation. Examples of the performance improvement possible, for several possible settings, are summarised in the table below. Here the spreading factor and error correction rate are design variables that allow the designer to optimise the trade-off between occupied bandwidth, data rate, link budget improvement and immunity to interference.

Table 12 Example LoRa™ Modem Performances, 868MHz Band

Bandwidth (kHz)	Spreading Factor	Coding rate	Nominal Rb (bps)	Sensitivity Indication (dBm)	Frequency Reference
10.4	6	4/5	782	-131	TCXO
	12	4/5	24	-147	
20.8	6	4/5	1562	-128	
	12	4/5	49	-144	
62.5	6	4/5	4688	-121	XTAL
	12	4/5	146	-139	
125	6	4/5	9380	-118	
	12	4/5	293	-136	

Notes - for all bandwidths lower than 62.5 kHz, it is advised to use a TCXO as a frequency reference. This is required to meet the frequency error tolerance specifications given in the Electrical Specification

- Higher spreading factors and longer transmission times impose more stringent constraints on the short term frequency stability of the reference. Please get in touch with a Semtech representative to implement extremely low sensitivity products.

For European operation the range of crystal tolerances acceptable for each sub-band (of the ERC 70-03) is given in the specifications table. For US based operation a frequency hopping mode is available that automates both the LoRa™ spread spectrum and frequency hopping spread spectrum processes.

Another important facet of the LoRa™ modem is its increased immunity to interference. The LoRa™ modem is capable of co-channel GMSK rejection of up to 20 dB. This immunity to interference permits the simple coexistence of LoRa™ modulated systems either in bands of heavy spectral usage or in hybrid communication networks that use LoRa™ to extend range when legacy modulation schemes fail.

LAMPIRAN E

DATASHEET CAPACITIVE SOIL MOISTURE SENSOR

Specification

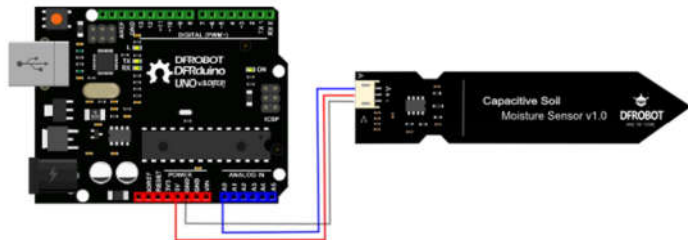
- Operating Voltage: 3.3 – 5.5 VDC
- Output Voltage: 0 ~ 3.0VDC
- Operating Current: 5mA
- Interface: PH2.0-3P
- Dimensions: 3.86 x 0.905 inches (L x W)
- Weight: 15g

Tutorial

Requirements

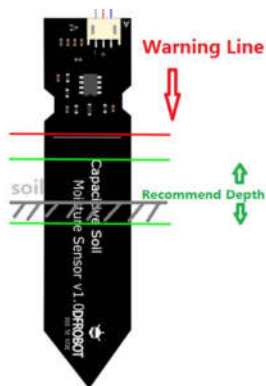
- **Hardware**
 - UNO x1
 - Capacitive Soil Moisture Sensor x1
 - Jumper Cable x3
- **Software**
 - Arduino IDE V1.6.5 [Click to Download Arduino IDE](#)

Connection Diagram



Calibration

Calibration Range



1. Open the serial port monitor and set the baud rate to 9600
2. Record the sensor value when the probe is exposed to the air as "Value 1". This is the boundary value of dry soil "Humidity: 0%RH"
3. Take a cup of water and insert the probe into it no further than the red line in the diagram
4. Record the sensor value when the probe is exposed to the water as "Value 2". This is the boundary value of moist soil "Humidity: 100%RH"

LAMPIRAN F

DATASHEET RTC DS1307



DS1307

64 x 8 Serial Real-Time Clock

www.maxim-ic.com

FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

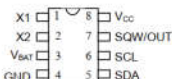
ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

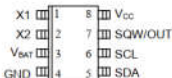
DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

PIN ASSIGNMENT



DS1307 8-Pin DIP (300-mil)

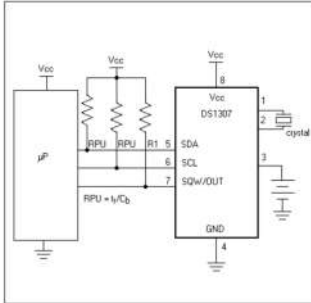


DS1307 8-Pin SOIC (150-mil)

PIN DESCRIPTION

V _{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V _{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

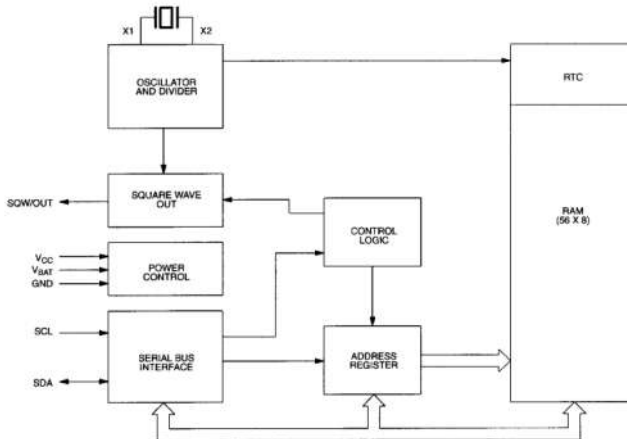
TYPICAL OPERATING CIRCUIT



OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below $1.25 \times V_{BAT}$ the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low-current battery backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than $V_{BAT} + 0.2V$ and recognizes inputs when V_{CC} is greater than $1.25 \times V_{BAT}$. The block diagram in Figure 1 shows the main elements of the serial RTC.

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5V input. When 5V is applied within normal limits, the device is fully accessible and data can be written and read. When a 3V battery is connected to the device and V_{CC} is below 1.25 x V_{BAT}, reads and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage. As V_{CC} falls below V_{BAT} the RAM and timekeeper are switched over to the external power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} – Battery input for any standard 3V lithium cell or other energy source. Battery voltage must be held between 2.0V and 3.5V for proper operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See “Conditions of Acceptability” at <http://www.maxim-ic.com/TechSupport/QA/ntrl.htm>.

SCL (Serial Clock Input) – SCL is used to synchronize data movement on the serial interface.

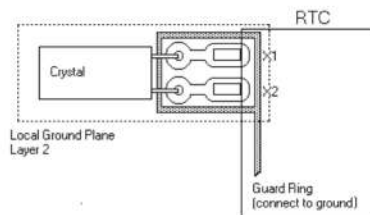
SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

SQW/OUT (Square Wave/Output Driver) – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pull-up resistor. SQW/OUT will operate with either Vcc or Vbat applied.

X1, X2 – Connections for a standard 32.768kHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5pF.

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, “Crystal Considerations with Dallas Real-Time Clocks.” The DS1307 can also be driven by an external 32.768kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

RECOMMENDED LAYOUT FOR CRYSTAL



CLOCK ACCURACY

The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. See Application Note 58, "Crystal Considerations with Dallas Real-Time Clocks" for detailed information.

Please review Application Note 95, "Interfacing the DS1307 with a 8051-Compatible Microcontroller" for additional information.

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

DS1307 ADDRESS MAP Figure 2

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The RTC registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. Bit 7 of register 0 is the clock halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

Please note that the initial power-on state of all registers is not defined. Therefore, it is important to enable the oscillator (CH bit = 0) during initial configuration.

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.

BIODATA PENULIS



Mabarrus Sholeh lahir di Kebumen pada tanggal 05 Juli 1997 merupakan anak pertama dari empat bersaudara. Penulis menyelesaikan pendidikan dasar di SD Negeri 2 Kewangunan, dilanjutkan pendidikan pendidikan tingkat menengah di SMP Negeri 1 Petanahan dan sekolah tingkat atas di SMA Negeri 2 Kebumen. Penulis memulai kehidupan perkuliahan pada tahun 2015 di Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya.

Selama masa perkuliahan, penulis aktif dalam berbagai kegiatan kepanitiaan dan organisasi mulai dari tingkat jurusan hingga nasional. Selain itu, penulis juga menjadi asisten praktikum di bidang studi elektronika.

Email : Mabarrussholeh@gmail.com
Hp/WA : 085735269672
Facebook : Mabarrus Sholeh
Line : 123shol

