

**IMPLEMENTASI KEBUN CERDAS PADA PERKEBUNAN  
HIDROPONIK SISTEM *DEEP FLOW TECHNIQUE* (DFT)  
TERINTEGRASI IoT**

**PROYEK AKHIR**

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung



Disusun Oleh :

Edi Kuswara

NIM: 1051907

Muhammad Miftah Syahfiqri NIM: 1051918

**POLITEKNIK MANUFAKTUR NEGERI  
BANGKA BELITUNG  
TAHUN 2023**

**LEMBAR PENGESAHAN**

**IMPLEMENTASI KEBUN CERDAS PADA PERKEBUNAN  
HIDROPONIK SISTEM *DEEP FLOW TECHNIQUE* (DFT)  
TERINTEGRASI IoT**

Oleh:

Edi Kuswara

NIM: 1051907

Muhammad Miftah Syahfiqri NIM: 1051918

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat  
kelulusan Program Sarjana Terapan Politeknik Manufaktur Negeri  
Bangka Belitung

Menyetujui,

Pembimbing 1



Muhammad Iqbal Nugraha, M.Eng.

Pembimbing 2



Zanu Saputra, M. Tr.T.

Penguji 1



Aan Febriansyah, M.T

Penguji 2



Indra Dwisaputra, M.T

## PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Edi Kuswara NIM : 1051907

Nama Mahasiswa 2 : M. Miftah Syahfiqri NIM : 1051918

Dengan Judul : Implementasi Kebun Cerdas Pada  
Perkebunan Hidroponik Ssitem *Deep Flow Technique* (DFT)  
Terintegrasi IoT

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 16 Februari 2023

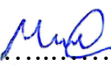
Nama Mahasiswa

TandaTangan

1. Edi Kuswara

(..........)

2. Muhammad Miftah Syahfiqri

(..........)

## ABSTRAK

*Hidroponik merupakan teknik budidaya tanaman sayuran yang menggunakan media air sebagai pengganti media tanah. Hal terpenting dalam teknik budaya ini adalah menjaga aspek kebutuhan tanaman, diantaranya kebutuhan air, kadar nutrisi dalam air (PPM), pH, suhu air, dan kebutuhan sinar matahari, yang semuanya masih dilakukan secara manual pada hidroponik konvensional. Tujuan dari penelitian ini adalah untuk mengembangkan sistem yang dapat memantau aspek-aspek tersebut terintegrasi aplikasi IoT, sekaligus dapat menjaga kebutuhan air dan nutrisi pada perkebunan hidroponik secara otomatis. Pada penelitian ini, metode regresi linier digunakan pada proses kalibrasi sensor, sehingga didapatkan hasil pembacaan sensor dengan akurasi yang baik, yaitu 95,61% pada sensor TDS, sensor pH 99,06%, sensor suhu air 99,27%, dan sensor ultrasonik 99,31%. Sistem kendali pengisian air dan nutrisi otomatis mampu menjaga larutan nutrisi pada kadar PPM kisaran 600-800 PPM. Selain itu, intensitas cahaya pada lingkungan tanaman berbasis sensor lux dan lampu dapat dikendalikan melalui aplikasi smartphone dengan tingkat akurasi 100%. Tanaman selada hasil pertanian yang diproduksi dengan menerapkan sistem memiliki karakteristik dengan tinggi rata-rata 37 cm, berat 186 gr, dan jumlah daun 18 helai.*

*Kata kunci: Hidroponik, Regresi Linier, Aplikasi Smartphone*

## **ABSTRACT**

*Hydroponics are a technique for harvesting vegetables that use water-based media to replace soil media. The most important thing in these cultural techniques is to address the needs of plants, including water needs, nutrition levels in water (PPM), pH, water temperature, and sunlight needs, all of which are still done by hand in conventional hydroponics. The purpose of the study is to develop systems that monitor those aspects integrated the application of IoT, while also maintaining water and nutrients needs on automatically hydroponics plantations. In this study, linear regression methods are used in the process of sensor calibration, and they are obtained with good accuracy from sensor reading, which is 95.61% in TDS sensors, 99.06% pH sensors, water temperature 99.27%, and ultrasonic 99.31%. The control system that replaces water and automated nutrients keeps the nutrient solution at a high of 600-800 PPM range. In addition, light intensity in the custom-based plant environment and lights can be controlled through a smartphone application with a 100% accuracy. Agricultural lettuce plants produced by applying a system have characteristics of an average height of 37 cm (23 in.), 186 gr (186 in.), and the number of leaves of 18 leaves.*

*Keywords: Hydroponics, Linear Regression, Smartphone Applications*

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa berkat rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan makalah proyek akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan.

Makalah ini dibuat untuk memenuhi salah satu syarat wajib kelulusan Diploma IV di Politeknik Manufaktur Negeri Bangka Belitung. Dengan adanya makalah ini diharapkan pembaca dapat mengetahui gambaran proyek akhir yang dibuat oleh penulis. Makalah proyek akhir ini dibuat dengan melakukan pengembangan dari jurnal-jurnal penelitian yang sudah pernah dilakukan sebelumnya. Penulis mencoba menerapkan ilmu pengetahuan yang didapat selama 4 tahun menempuh pendidikan di Politeknik Manufaktur Negeri Bangka Belitung dalam pembuatan proyek akhir ini. Selain itu, penulis juga mendapatkan informasi berupa data-data pendukung dari makalah-makalah proyek akhir mahasiswa Politeknik Manufaktur Negeri Bangka Belitung tahun-tahun sebelumnya.

Selama menyusun makalah proyek akhir ini penulis mendapatkan banyak bantuan dan bimbingan dari berbagai pihak sehingga penulisan makalah ini dapat diselesaikan dengan baik. Oleh karena itu, penulis ingin mengucapkan terima kasih kepada:

1. Orang tua dan keluarga tercinta yang selalu memberikan dukungan baik secara moral maupun materil sehingga penulis dapat memberikan hasil yang terbaik.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
3. Bapak Muhammad Iqbal Nugraha, M.Eng selaku dosen pembimbing 1 dalam proyek akhir ini.
4. Bapak Zanu Saputra, M.Tr.T. selaku dosen pembimbing 2 dalam proyek akhir ini.

5. Bapak Muhammad Iqbal Nugraha, M.Eng. selaku Ketua Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Bapak Indra Dwisaputra, M.T. selaku Ketua Program Studi DIV Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
7. Seluruh tenaga pendidik dan kependidikan Politeknik Manufaktur Negeri Bangka Belitung.
8. Rekan-rekan mahasiswa Politeknik Manufaktur Negeri Bangka Belitung yang telah membantu dalam penyelesaian proyek akhir.
9. Seluruh pihak yang ikut terlibat baik secara langsung maupun tidak langsung dalam penyelesaian proyek akhir ini yang tidak bisa disebutkan satu persatu.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan karena terbatasnya pengetahuan dan kemampuan penulis. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari semua pihak demi perbaikan dan pengembangan penulisan makalah ini di masa yang akan datang. Penulis berharap makalah ini dapat bermanfaat bagi pihak yang berkepentingan pada khususnya dan bagi perkembangan ilmu pengetahuan dan teknologi pada umumnya.

Sungailiat, 16 Februari 2023

Penulis

## DAFTAR ISI

PROYEK AKHIR .....	i
PERNYATAAN BUKAN PLAGIAT .....	iii
ABSTRAK .....	iv
<i>ABSTRACT</i> .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR .....	xiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Proyek Akhir.....	2
BAB II LANDASAN TEORI.....	3
2.1 Hidroponik .....	3
2.1.1 Sistem DFT ( <i>Deep Flow Technique</i> ) .....	3
2.2 Sistem Automasi Hidroponik .....	4
2.3 Regresi Linier .....	5
2.4 Sensor Ultrasonik (SRF HRC04) .....	6
2.5 Sensor pH (pH DF Robot V1.1).....	7
2.6 Sensor TDS (TDS DF Robot Gravity) .....	8
2.7 Sensor Suhu (DS18B20) .....	10
2.8 <i>Real Time Clock</i> (RTC).....	11
2.9 Arduino Mega 2560 .....	11
2.10 Pompa Air .....	12



2.11	Nutrisi A-B <i>Mix</i> .....	13
2.12	LED UV .....	13
2.13	<i>Internet of things</i> (IoT).....	14
2.14	ESP32 .....	15
2.15	<i>Firebase</i> .....	16
2.16	<i>Thunkable</i> .....	16
BAB III METODE PELAKSANAAN .....		17
3.1	Studi Literatur .....	18
3.2	Perancangan dan Perakitan <i>Hardware</i> .....	18
3.2.1	Desain Modul Hidroponik.....	18
3.2.2	Perakitan Modul Hidroponik.....	19
3.2.3	Desain Rangkaian Elektrik.....	19
3.2.4	Perakitan Rangkaian Elektrik.....	20
3.3	Pengujian dan Kalibrasi Sensor.....	21
3.4	Perancangan dan Pembuatan Sistem Otomatis dan apliasi IoT .....	21
3.4.1	Desain <i>Software</i> .....	21
3.4.2	Pembuatan <i>Software</i> .....	22
3.4.3	<i>Login Screen</i> 22	
3.4.4	<i>Content Screen</i> .....	25
3.4.5	Notifikasi Kebutuhan Cahaya .....	29
3.4.6	Kontrol LED UV .....	30
3.4.7	Pembuatan Sistem Pengisian Air dan Nutrisi Otomatis.....	31
3.5	Pengujian Sistem Otomatis dan Aplikasi IoT .....	36
3.6	Pengujian Alat Secara Keseluruhan .....	36
3.7	Analisa Data .....	36

3.8	Pembuatan Laporan Akhir dan Publikasi.....	36
BAB IV PEMBAHASAN.....		38
4.1	Deskripsi Alat.....	38
4.2	Proses Kalibrasi dan Pengujian Sensor .....	39
4.2.1	Kalibrasi Sensor TDS.....	39
4.2.2	Pengujian Sensor TDS .....	42
4.2.3	Kalibrasi Sensor pH .....	44
4.2.4	Pengujian Sensor pH.....	48
4.2.5	Kalibrasi Sensor Suhu DS18B20 .....	50
4.2.6	Pengujian Sensor Suhu DS18B20 .....	53
4.2.7	Pengujian Sensor TDS, Sensor Suhu, dan Sensor pH Secara Bersamaan	55
4.2.8	Kalibrasi Sensor Ultrasonik .....	56
4.2.9	Pengujian Sensor Ultrasonik.....	59
4.2.10	Kalibrasi Sensor Ultrasonik Terhadap Pembacaan Volume Air.....	60
4.2.11	Kalibrasi Sensor Lux.....	63
4.2.12	Pengujian Sensor Lux.....	67
4.3	Pengujian Pompa Air .....	68
4.3.1	Pengujian Pompa Air Nutrisi .....	69
4.3.2	Pengujian Debit Pompa Air Sumur.....	70
4.4	Pengujian Hubungan Penambahan Nutrisi A-B <i>Mix</i> Terhadap Kenaikan Nilai PPM.....	72
4.5	Pengujian Hubungan Penambahan Air Terhadap Penurunan Nilai PPM .	74
4.6	Pengujian Data Waktu Pengosongan Air Pada Pipa Tanaman .....	75
4.7	Pengujian Sistem Pengisian Air dan Nutrisi Otomatis.....	76
4.8	Pengujian Aplikasi .....	78

4.8.1	Pengujian Pembacaan Sensor Pada Aplikasi.....	78
4.8.2	Pengujian Kontrol Lampu <i>Growlight</i> Pada aplikasi .....	79
4.9	Pengujian Keseluruhan Alat pada Sistem Hidroponik.....	80
4.9.1	Pengujian Pembacaan Sensor Pada LCD.....	80
4.9.2	Pengujian <i>Monitoring</i> Pada Aplikasi .....	84
4.9.3	Pengujian Kontrol Lampu Jarak Jauh Dengan Aplikasi .....	86
4.9.4	Hasil Pertanian .....	88
BAB V PENUTUP.....		91
5.1	Kesimpulan.....	91
5.2	Saran.....	92
DAFTAR PUSTAKA .....		93
LAMPIRAN 1 DAFTAR RIWAYAT HIDUP.....		98
LAMPIRAN 2 PROGRAM ARDUINO DAN NodeMCU ESP32 .....		101

## DAFTAR TABEL

Tabel 4. 1 Perbandingan nilai ADC sensor TDS dan nilai PPM .....	40
Tabel 4. 2 Data pengujian sensor TDS.....	43
Tabel 4. 3 Perbandingan nilai ADC sensor pH dan nilai pH .....	46
Tabel 4. 4 Data pengujian sensor pH .....	48
Tabel 4. 5 Data kalibrasi sensor Suhu .....	51
Tabel 4. 6 Data pengujian sensor Suhu.....	53
Tabel 4. 7 Data pengujian sensor TDS, sensor Suhu, dan sensor pH secara bersamaan.....	55
Tabel 4. 8 Data kalibrasi sensor Ultrasonik .....	57
Tabel 4. 9 Data Pengujian sensor Ultrasonik.....	59
Tabel 4. 10 Tabel Kalibrasi Sensor Ultrasonik untuk Pembacaan Volume Air....	61
Tabel 4. 11 Data kalibrasi sensor Lux.....	64
Tabel 4. 12 Tabel Pengujian Sensor Lux .....	68
Tabel 4. 13 Data pengujian Pompa Air Nutrisi.....	69
Tabel 4. 14 Data pengujian debit Pompa Air Sumur .....	71
Tabel 4. 15 Hubungan Pemanbahan Nutrisi A-B Mix terhadap Kenaikan PPM..	72
Tabel 4. 16 Hubungan Penambahan Air Terhadap Penurunan Nilai PPM .....	74
Tabel 4. 17 Data Pengujian Waktu Pengosongan Air pada Pipa Tanaman .....	75
Tabel 4. 18 Data pengujian sistem pengisian Air dan Nutrisi otomatis.....	77
Tabel 4. 19 Data pengujian sensor aplikasi HYDRO-JET.....	78
Tabel 4. 20 Data pengujian kontrol manual LED UV .....	79
Tabel 4. 21 Data pengujian sensor keseluruhan sistem.....	81
Tabel 4. 22 Tabel Tingkat Akurasi Pembacaan Sensor Keseluruhan Sistem .....	81
Tabel 4. 23 Data Pengujian Waktu Respon Pembacaan Sensor TDS.....	83
Tabel 4. 24 Perbandingan Pembacaan Sensor pada LCD dan Aplikasi Smartphone .....	84
Tabel 4. 25 Data Hasil Pengujian Kontrol Lampu dengan Aplikasi Smartphone	86
Tabel 4. 26 Data Sayuran Hasil Pertanian Hidroponik .....	88

## DAFTAR GAMBAR

Gambar 2. 1 Kebun Hidroponik.....	3
Gambar 2. 2 Sistem Deep Flow Technique (DFT) .....	3
Gambar 2. 3 Modul Sensor BH1750FV1 Light Sensor .....	5
Gambar 2. 4 Sensor Ultrasonik .....	6
Gambar 2. 5 Sensor pH .....	8
Gambar 2. 6 Sensor TDS (Part per Million) .....	9
Gambar 2. 7 Sensor Suhu (DS18B20) .....	10
Gambar 2. 8 RTC DS3231 .....	11
Gambar 2. 9 Arduino Mega 2560 .....	11
Gambar 2. 10 Pompa Air Kecil.....	12
Gambar 2. 11 Pompa Air Besar .....	12
Gambar 2. 12 Pupukl Nutrisi A-B Mix.....	13
Gambar 2. 13 LED UV (Growing light).....	14
Gambar 2. 14 IoT .....	15
Gambar 2. 15 ESP32.....	15
Gambar 2. 16 Firebase .....	16
Gambar 2. 17 Thunkable.....	16
Gambar 3. 1 Flowchart Pelaksanaan Pembuatan Proyek Akhir .....	17
Gambar 3. 2 Desain Awal Alat .....	19
Gambar 3. 3 Desain Rangkaian Elektrik.....	20
Gambar 3. 4 Perakitan Komponen ke dalam Kotak Panel.....	20
Gambar 3. 5 Login screen Aplikasi HYDRO-JET.....	22
Gambar 3. 6 Block Login Screen Aplikasi HYDRO-JET .....	23
Gambar 3. 7 Tampilan Berhasil Login.....	24
Gambar 3. 8 Tampilan Gagal Login .....	24
Gambar 3. 9 Content Screen Aplikasi HYDRO-JET.....	25
Gambar 3. 10 Block Inisialisasi Semua Data Dari Firebase .....	26
Gambar 3. 11 Block Memanggil Data Sensor Cahaya Ke Thunkable.....	26

Gambar 3. 12 Block Memanggil Data Sensor pH Ke Thunkable.....	26
Gambar 3. 13 Block Memanggil Data Sensor Suhu Ke Thunkable .....	27
Gambar 3. 14 Block Memanggil Data Sensor TDS Ke Thunkable .....	27
Gambar 3. 15 Block Memanggil Data Volume Ke Thunkable.....	27
Gambar 3. 16 Block Menampilkan Semua Data Sensor Ke Path Thunkable .....	28
Gambar 3. 17 Notifikasi Saran Menghidupkan LED UV .....	29
Gambar 3. 18 Block Notifikasi Saran Menghidupkan LED UV .....	29
Gambar 3. 19 Kondisi Switch Ketika LED UV Hidup .....	30
Gambar 3. 20 Kondisi Switch Ketika LED UV Mati .....	30
Gambar 3. 21 Block Program Kontrol Lampu Growlight .....	30
Gambar 3. 22 Program Arduino Pengisian Air dan Nutrisi Otomatis .....	31
Gambar 3. 23 Program Pada Saat Sistem Mengisi Ulang Air .....	32
Gambar 3. 24 Program Untuk Memberikan Delay Waktu Pengosongan Pipa Tanaman.....	32
Gambar 3. 25 Program Cek Point Apabila Kadar Nutrisi (PPM) Dalam Air kurang Dari Yang Ditetapkan .....	33
Gambar 3. 26 Program Cek Point Untuk Pengencaran Kadar PPM .....	34
Gambar 3. 27 Program Waktu Penyesuaian Bagi Pembacaan Response time Sensor TDS.....	35
Gambar 3. 28 Program Untuk Melakukan Distribusi Air Nutrisi Setelah Pengisian Ulang.....	35
Gambar 4. 1 Blok Diagram Alat .....	38
Gambar 4. 2 Rangkaian Skematik Kalibrasi Sensor TDS.....	39
Gambar 4. 3 Program Arduino pada Proses Kalibrasi Sensor TDS.....	40
Gambar 4. 4 Grafik Hubungan nilai PPM dan ADC .....	41
Gambar 4. 5 Program Arduino Untuk Konversi Nilai ADC Menjadi PPM Dengan Persamaan Regresi Linier .....	42
Gambar 4. 6 Pengujian Kadar PPM Menggunakan Sensor TDS dan TDS Meter	44
Gambar 4. 7 Pembacaan Sensor TDS pada Serial Monitor .....	44
Gambar 4. 8 Rangkaian Skematik Pengujian Sensor pH.....	45
Gambar 4. 9 Program Arduino Kalibrasi Sensor pH .....	45

Gambar 4. 10 Sampel Air Kalibrasi Sensor pH .....	46
Gambar 4. 11 Grafik Hubungan nilai PPM dan ADC .....	47
Gambar 4. 12 Program Arduino Pembacaan Nilai pH dengan Regresi Linier .....	47
Gambar 4. 13 Sampel Air Pengujian Sensor pH.....	49
Gambar 4. 14 Tampilan Serial Monitor Pengujian Sensor pH .....	49
Gambar 4. 15 Pengujian Sensor pH dan pH Meter .....	50
Gambar 4. 16 Rangkaian Skematik Pengujian Sensor Suhu DS18B20.....	50
Gambar 4. 17 Program Arduino Pembacaan Sensor Suhu DS18B20.....	51
Gambar 4. 18 Grafik Hubungan nilai Suhu antara sensor Suhu dan Termometer	52
Gambar 4. 19 Program Arduino Regresi Linier Sensor Suhu.....	53
Gambar 4. 20 Proses Pengujian Suhu dengan Sensor Suhu DS18B20 dan Termometer.....	54
Gambar 4. 21 Hasil Pembacaan Sensor Suhu DS18B20 pada Serial Monitor .....	54
Gambar 4. 22 Pengujian Sensor TDS, pH, dan Suhu bersamaan .....	56
Gambar 4. 23 Rangkaian Skematik Pengujian Sensor Ultrasonik.....	56
Gambar 4. 24 Program Arduino Pembacaan Sensor Ultrasonik.....	57
Gambar 4. 25 Grafik Hubungan nilai jarak antara sensor Ultrasonik dan Meteran .....	58
Gambar 4. 26 Program Arduino Sensor Ultrasonik Regresi Linier.....	59
Gambar 4. 27 Pengujian Pembacaan Jarak dengan Sensor Ultrasonik dan Meteran .....	60
Gambar 4. 28 Grafik Hubungan Jarak dan Volume air .....	61
Gambar 4. 29 Program Arduino Konversi Nilai Jarak Menjadi Volume Air .....	62
Gambar 4. 30 Kalibrasi Sensor Ultrasonik terhadap Pembacaan Volume Air .....	63
Gambar 4. 31 Rangkaian Skematik Pengujian Sensor Lux .....	63
Gambar 4. 32 Program Arduino Pembacaan Sensor Lux .....	64
Gambar 4. 33 Grafik Hubungan Nilai Lux Pembacaan Sensor dan Lux Meter (1) .....	65
Gambar 4. 34 Grafik Hubungan Nilai Lux Pembacaan Sensor dan Lux Meter (2) .....	66

Gambar 4. 35 Grafik Hubungan Nilai Lux Pembacaan Sensor dan Lux Meter (3)	66
.....	66
Gambar 4. 36 Program Kalibrasi Sensor Lux dengan Rumus Regresi.....	67
Gambar 4. 37 Pengujian Pompa Air Nutrisi .....	69
Gambar 4. 38 Pengujian Nilai Debit Pompa Air Sumur.....	72
Gambar 4. 39 Pengujian Hubungan Penambahan Nutrisi AB Mix Terhadap Nilai PPM.....	73
Gambar 4. 40 Pengujian Hubungan Penambahan Air Terhadap Nilai PPM .....	75
Gambar 4. 41 Proses Pengujian Data Waktu Pengosongan Air dengan Stopwatch .....	76
Gambar 4. 42 Pengujian Sistem Pengisian Air dan Nutrisi Otomatis.....	77
Gambar 4. 43 Hasil pembacaan semua sensor pada saat sistem berjalan pada LCD .....	78
Gambar 4. 44 Hasil pembacaan semua sensor pada saat sistem berjalan pada aplikasi HYDRO-JET .....	79
Gambar 4. 45 Pengujian Kontrol Lampu dari Aplikasi Smartphone.....	80
Gambar 4. 46 Tampilan LCD Pengujian Sensor pada Keseluruhan Sistem.....	82
Gambar 4. 47 Pengujian nilai PPM dan Suhu Air .....	82
Gambar 4. 48 Pengujian nilai pH Air dan Intensitas Cahaya .....	83
Gambar 4. 49 Tampilan pembacaan sensor pada LCD.....	85
Gambar 4. 50 Tampilan pembacaan sensor pada aplikasi .....	85
Gambar 4. 51 Pengujian Kontrol Lampu Growlight.....	87
Gambar 4. 52 Pengujian kontrol lampu berbasis aplikasi dari jarak jauh.....	88
Gambar 4. 53 Proses Pengukuran Tinggi Sayuran .....	89
Gambar 4. 54 Proses Menimbang Berat Sayuran Hasil Pertanian.....	90



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Metode bertani hidroponik merupakan metode bertani yang mengharuskan para pelakunya memantau keadaan volume air serta nilai jumlah zat terlarut dalam air yang memiliki satuan *Parts Per Million* (PPM) pada bak nutrisi tanaman. Ketika nilai PPM serta volume air pada bak nutrisi sudah berkurang, petani hidroponik tidak berada dekat dengan perkebunan, sehingga tidak dapat melakukan pengisian air dan nutrisi pada bak sesegera mungkin. Serta salah satu masalah lain yang berpotensi sering terjadi ketika melakukan budidaya tanaman hidroponik secara manual yaitu kondisi takaran air yang belum tepat serta nutrisi yang selalu berubah-ubah. Dari permasalahan tersebut, maka perlu dibangun sebuah sistem otomatisasi pada pengontrolan larutan nutrisi tanaman hidroponik [1]–[3]. Kondisi nutrisi yang selalu berubah-ubah atau tidak konstan membutuhkan pengontrolan menggunakan arduino sebagai mikrokontroler pada sistem otomatisasi ini, hal tersebut dibutuhkan agar mencapai tujuan nilai larutan nutrisi yang nantinya akan diberikan pada tanaman selalu tepat dan tidak berubah-ubah [4], [5]. Selain itu masalah serius yang banyak dihadapi oleh para pelaku hidroponik ini adalah masalah kebutuhan cahaya bagi tanaman yang dipengaruhi oleh cuaca hujan atau berawan yang dapat mempengaruhi proses fotosintesis tanaman. Solusi yang dapat diberikan bagi pelaku hidroponik dalam kasus ini salah satunya seperti harus memperkerjakan seorang pegawai untuk memantau hidroponiknya demi mendapatkan hasil pertanian yang maksimal. Dengan solusi seperti itu secara tidak langsung akan mengurangi efisiensi serta efektifitas dari metode bertani ini sendiri dikarenakan harus menyertai biaya tambahan untuk membayar pegawai tersebut.

Berdasarkan masalah tersebut diatas, maka sistem proses pengairan dan penakaran nutrisi pada hidroponik secara otomatis terintegrasi teknologi IoT sangat diperlukan dengan mengombinasikan beberapa sensor dan actuator yang relevan sehingga dapat mengurangi dampak akibat *human error*. Proyek akhir ini focus

kepada sistem implementasi kebun cerdas pada perkebunan hidroponik sistem *Deep Flow Technique* (DFT) terintegrasi IoT.

## 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang masalah, maka beberapa rumusan masalah yang dikaji bersama penulis dalam laporan proyek akhir ini adalah sebagai berikut:

1. Bagaimana membaca, mengolah, dan mengkalibrasi sensor pH, PPM, suhu, ketinggian air, dan intensitas cahaya, serta ditampilkan melalui *smartphone* ?
2. Bagaimana cara mengontrol pemenuhan kebutuhan cahaya, nutrisi dan ketinggian air pada wadah nutrisi sistem perkebunan hidroponik secara otomatis ?

## 1.3 Batasan Masalah

Batasan - batasan masalah pada proyek akhir ini adalah sebagai berikut:

1. Jenis tanaman yang digunakan adalah tanaman sayur selada.
2. Sistem dapat mengontrol pembuatan air nutrisi dengan nilai PPM sesuai dengan kebutuhan jenis tanaman secara otomatis dan menghidupkan lampu LED UV secara manual.
3. Sistem hanya dapat melakukan *monitoring* nilai pH, suhu, PPM, dan ketinggian air nutrisi pada wadah nutrisi, serta melakukan *monitoring* nilai intensitas cahaya pada instalasi hidroponik.

## 1.4 Tujuan Proyek Akhir

Adapun tujuan pembuatan proyek akhir ini adalah sebagai berikut:

1. Merancang dan membuat sistem implementasi kebun cerdas pada perkebunan hidroponik sistem *Deep Flow Technique* (DFT) terintegrasi IoT
2. Meningkatkan produksi pertanian hidroponik dengan memaksimalkan kebutuhan nutrisi dan cahaya bagi tanaman.

## BAB II

### LANDASAN TEORI

#### 2.1 Hidroponik

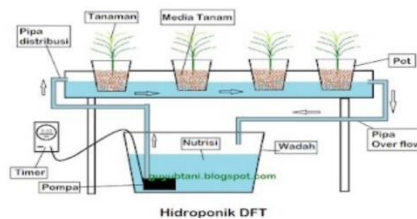
Hidroponik adalah teknik budidaya tanaman sayuran yang menggunakan media air sebagai pengganti media tanah. Teknik ini biasanya disertai dengan adanya greenhouse yang berguna untuk menjaga pertumbuhan tanaman dari pengaruh unsur luar seperti hama, hujan, iklim dan lainnya[6], Dua dari beberapa system yang ada pada hidroponik adalah sistem NFT (*Nutrient Film Technique*) dan DFT (*Deep Flow Technique*).



Gambar 2. 1 Kebun Hidroponik  
Sumber : (Suara.com)

##### 2.1.1 Sistem DFT (*Deep Flow Technique*)

Sistem DFT (*Deep Flow Technique*) pada hidroponik adalah jenis sistem yang menyisahkan air tergenang pada pipa tanaman ketika proses sirkulasi air dimatikan dengan frekuensi yang tergolong pelan [7]. Sehingga saat proses penanaman sayuran hidroponik ketika sirkulasi air dalam kondisi tidak hidup, sayur tetap bias menerima asupan air serta nutrisi dari air yang tergenang pada pipa.



Gambar 2. 2 Sistem *Deep Flow Technique* (DFT)  
Sumber : (jurnalsoreang.pikiran-rakyat.com)

## 2.2 Sistem Automasi Hidroponik

Sistem automasi hidroponik merupakan sebuah ekosistem lanjutan dari pertanian hidroponik. Pada sistem ini controlling dan monitoring tanaman yang biasanya dipantau manual oleh manusia langsung akan terlaksa secara otomatis dengan memanfaatkan teknologi berbasis IoT [8]. Penelitian terdahulu tentang sistem automasi hidroponik sudah pernah dilakukan oleh Susilo Firmansyah, Cahyo Prihantoro, M. Zaky Farras Baihaqi , dan Hilmy Ahamd Haidar dengan judul penelitian “Sistem Automasi Hidroponic Berbasis IOT” pada tahun 2022 serta Aldion Amirrul Endryanto dan Nuril Esti Khomariah dengan judul penelitian “Kontrol dan Monitoring Tanaman Hiroponik Sistem Nutrient Film Technique Berbasis IoT” [6].

Dari kedua judul tersebut, para peneliti sudah berhasil membuat sistem automasi hidroponik yang dimana pengisian ulang larutan nutrisi terjadi secara otomatis ketika sudah mencapai titik minimal tertentu seperti nilai PPM atau volume larutan serta monitoring semua aspek yang ada pada sistem dengan memanfaatkan teknologi IoT. Namun pembacaan nilai PPM pada larutan nutrisi melalui tds sensor pada alat yang sudah dibuat kurang stabil, serta kurangnya beberapa unsur yang belum disediakan pada penelitian sebelumnya seperti pH, suhu, dan kebutuhan cahaya pada tanaman. Pada penelitian ini, penulis bermaksud memodifikasi alat yang sudah ada dengan menambahkan metode yang berbeda pada saat mengambil nilai PPM pada larutan, serta menambahkan kondisi Mixing pada sistem pengisian ulang larutan nutrisi yang semuanya bertujuan untuk mendapatkan nilai PPM yang stabil pada waktu yang bersamaan. Lalu penulis juga ingin menambahkan monitoring terhadap kondisi pH dan suhu larutan nutrisi serta control secara manual keadaan cahaya dalam satuan lux di lingkungan sistem automasi hidroponik, yang berfungsi untuk meningkatkan kualitas pertumbuhan tanaman.

### 2.3 Regresi Linier

Regresi linier merupakan suatu metode yang digunakan dalam membentuk suatu model hubungan antara variable dependen (terikat) dengan satu atau lebih variabel independent (bebas)[9]. Pada penelitian ini, penulis menggunakan metode regresi linier untuk mendapatkan nilai persamaan dari hubungan 2 buah variable. Penggunaan metode ini terletak pada proses kalibrasi sensor. Penulis menggunakan Microsoft Excel sebagai aplikasi pembantu dalam mendapatkan persamaan regresi linier.

#### Sensor Cahaya (Modul Sensor BH1750FV1 Light Sensor)

Sensor cahaya merupakan sebuah komponen elektronika yang berguna untuk mengukur serta melihat nilai intensitas cahaya dalam satuan Lux (lx). Sensor jenis BH750FVI ini memiliki tingkat pembacaan yang cukup tinggi di bandingkan dengan sensor LDR atau photodiode [10]. Dipilihnya sensor ini karena memiliki range pengukuran yang mumpuni dibandingkan dengan sensor TSL2561 pada skala perhitungan cuaca, serta mengalahkan pesaingnya dari segi spesifikasi [11]. Pada penelitian Rifki Nur Ilham tentang Sistem Monitoring Pendingin Pada Panel Surya, telah dibuktikan bahwa performa sensor BH 1750 dapat mengambil data intensitas dengan baik pada kondisi diluar ruangan [12].



Gambar 2. 3 Modul Sensor BH1750FV1 Light Sensor  
Sumber : (gravitechthai.com)

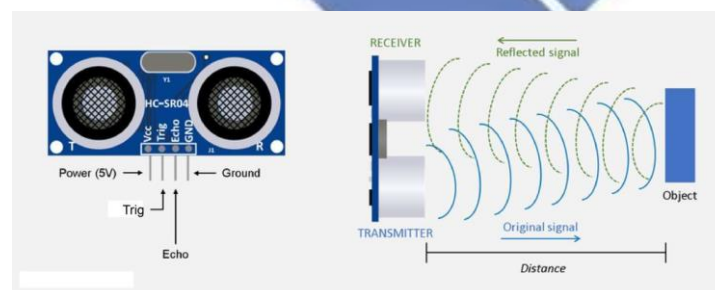
Pada pembuatan proyek ini, sensor cahaya digunakan untuk mengukur besar intensitas cahaya. Sensor akan mendeteksi gelap terang cahaya matahari yang

masuk ke lingkungan kebun hidroponik. Nilai lux yang didapatkan dari sensor akan menjadi indikator penghidupan UV LED pada sistem hidroponik.

#### 2.4 Sensor Ultrasonik (SRF HRC04)

Sensor ultrasonik adalah sensor yang mengubah besaran bunyi menjadi besaran listrik. Sensor ultrasonik memiliki fungsi secara umum untuk suatu mengukur jarak suatu objek dengan titik ukur lainnya. Sensor ini bekerja dengan memancarkan gelombang ultrasonik menuju ke suatu target yang ditentukan lalu dipantulkan kembali gelombang tersebut balik ke arah sensor. Kemudian sistem akan mengukur waktu yang dihabiskan pada saat proses pemancaran gelombang dimulai sampai kembali ke sensor serta menggunakan kecepatan suara untuk menghitung jarak target. Rangkaian penyusun sensor ini adalah *transmitter*, *receiver*, dan *comparator* [13].

Pemilihan sensor ultrasonic jenis ini dikarenakan sensor HRC04 ini memiliki tingkat akurasi pembacaan yang tinggi, namun dengan harga yang terjangkau. Pada penelitian Imam Muklisin yang berjudul Pendeteksi Volume Tandon Air Secara Otomatis Menggunakan Sensor Ultrasonik Berbasis Arduino Uno R3 menunjukkan bahwa sensor ultrasonic HRC04 ini mampu mendeteksi volume air dengan tingkat akurasi 99,10% [14].



Gambar 2. 4 Sensor Ultrasonik  
Sumber : (mahirelektro.com)

Pada proyek ini, sensor ultrasonik digunakan untuk mengukur nilai dari ketinggian air nutrisi pada wadah nutrisi hidroponik dalam satuan sentimeter (cm).

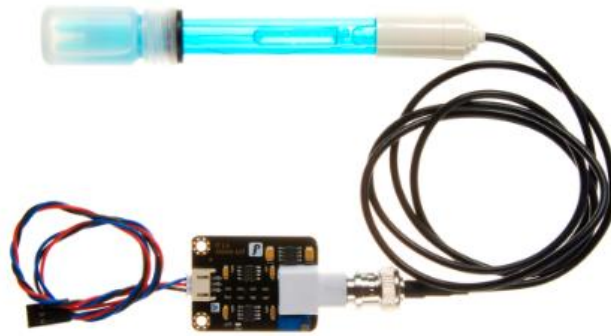
Nilai yang didapat dari sensor ini nantinya akan digunakan pada perhitungan volume air dan kebutuhan serta penyesuaian nilai PPM pada wadah nutrisi.

## **2.5 Sensor pH (pH DF Robot V1.1)**

PH adalah satuan derajat keasaman untuk mengukur basa atau asam pada suatu larutan. Larutan tersebut dapat dibilang asam atau basa berdasarkan tingkat derajat keasaman yang dimilikinya. Sejauh ini tingkat derajat keasaman memiliki rentang nilai dari 0 sampai 14. Larutan netral memiliki derajat keasaman pada nilai 6,5 hingga 7,5. Derajat keasaman lebih dari 7,5 bernilai basa dan derajat keasaman kurang dari 6,5 bernilai asam [15].

Nilai pH yang tidak sesuai akan mempengaruhi penyerapan unsur hara oleh tanaman. Bila kondisi pH pada media tumbuh tanaman bersifat asam, maka penyerapan unsur hara oleh tanaman akan terhambat yang menyebabkan pertumbuhan tanaman terlambat atau menjadi kerdil [16]. Hal ini pula yang menjadikan nilai pH dari larutan nutrisi tanaman menjadi salah satu aspek penting dalam perkebunan Hidroponik.

Sensor pH Meter merupakan alat ukur untuk melihat tingkat keasaman cairan (pH) menggunakan penginderaan pengukur berstandar industri yang menjadi komponen utamanya. Bagian elektroda sensor terbuat dari membran kaca sensitif yang memiliki tingkat impedansi kecil sehingga menghasilkan pengukuran dengan respon yang cepat dan stabilitas. Hasil pembacaan dari sensor ini dapat diakses oleh mikrokontroler [17] . Penggunaan sensor pH pada proyek ini adalah untuk mengetahui nilai pH dari air nutrisi yang ada pada wadah nutrisi secara *real time*, sehingga nilai pH yang merupakan salah satu aspek penting dalam pertanian hidroponik dapat terpantau .



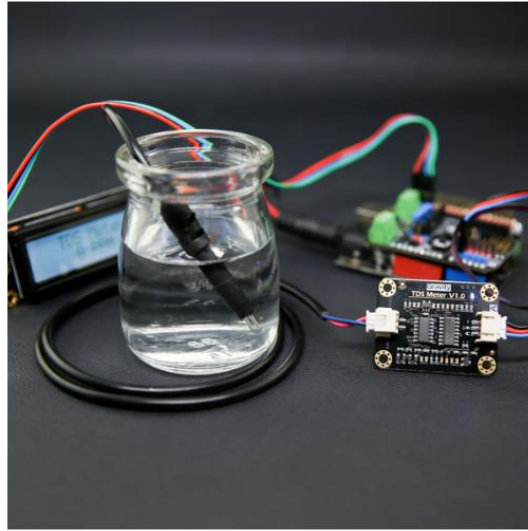
Gambar 2. 5 Sensor pH  
Sumber : (dfrobot.com)

Pemilihan sensor pH jenis ini berlandaskan penelitian yang dilakukan oleh Dewi Ratna Wati yang berjudul Pengontrol pH dan Nutrisi Tanaman Selada pada Hidroponik Sistem NFT Berbasis Arduino. Pada penelitian ini, dibuktikan bahwa tingkat akurasi pembacaan dari sensor pH DF Robot V1.1 berada pada nilai 99,96% [18].

## 2.6 Sensor TDS (TDS DF Robot Gravity)

Sensor TDS adalah jenis sensor yang memiliki cara kerja mendeteksi konduktivitas pada suatu larutan. Apabila pada larutan terkanandung banyak mineral maka konduktivitasnya akan semakin tinggi dan *outputnya* juga akan semakin besar[19]. Tampilan sensor TDS diperlihatkan pada gambar 2.6.





Gambar 2. 6 Sensor TDS (*Part per Million*)  
Sumber : (dfrobot.com)

Dalam proyek ini, sensor TDS digunakan untuk melakukan pengukuran terhadap nilai kepekatan dari larutan nutrisi tanaman. Satuan yang digunakan dari hasil pembacaan sensor TDS adalah PPM (*Part per Million*).

PPM sendiri merupakan singkatan dari “*Part Per Million*” atau “*Sepersejuta Bagian*” merupakan satuan untuk mengukur kepekatan suatu larutan. Dalam sistem pertanian hidroponik, pengukuran kepekatan pada larutan nutrisi diperlukan untuk menyesuaikan kebutuhan nutrisi pada fase pertumbuhan tanaman [20]. Nilai PPM dapat diukur menggunakan alat TDS meter.

Pemilihan Sensor TDS jenis ini berlandaskan penelitian yang telah dilakukan oleh Dhodit Rengga Tisna tentang Metode Peningkatan Akurasi pada Sensor TDS berbasis Arudino untuk Nutrisi Air Menggunakan Regresi Linier. Dari penelitian tersebut, didapatkan tingkat akurasi pembacaan sensor TDS DF Robot Gravity berada pada angka 98,3% [21]. Hal ini menunjukkan bahwa sensor TDS jenis ini mampu mengukur kadar TDS dalam air dengan baik.

## 2.7 Sensor Suhu (DS18B20)

Sensor suhu adalah suatu komponen elektronika yang digunakan untuk mengubah besaran panas menjadi besaran listrik untuk mendeteksi ada atau tidaknya perubahan suhu pada suatu objek [22]. Suhu air menjadi salah satu faktor yang penting dalam sistem perkebunan hidroponik karena air merupakan sumber hara bagi tanaman tersebut. Suhu yang tergolong baik untuk tanaman berkisar pada  $25^{\circ}\text{C}$  sampai  $28^{\circ}\text{C}$  [23] . Berikut ini merupakan gambar dari Sensor Suhu DS18B20.

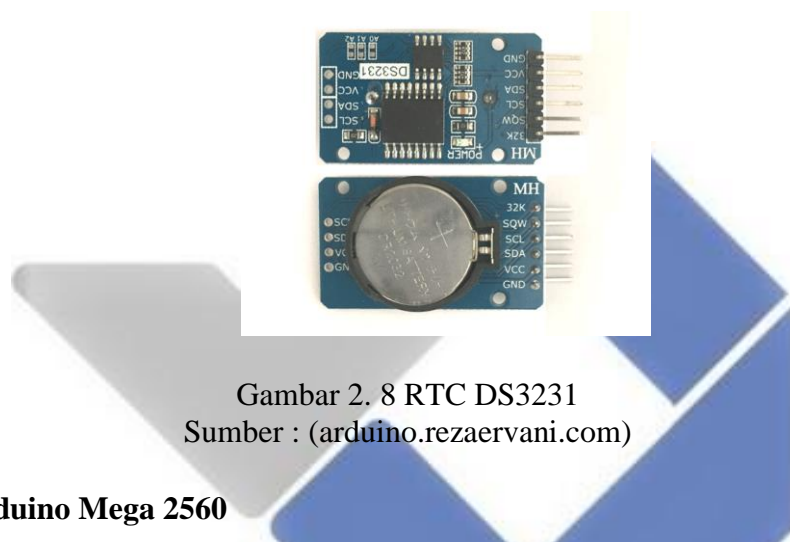


Gambar 2. 7 Sensor Suhu (DS18B20)  
Sumber : (dfrobot.com)

Sensor suhu DS18B20 digunakan untuk mengukur suhu dalam wadah nutrisi hidponik, dimana data hasil pembacaan sensor akan ditampilkan pada LCD dan aplikasi sistem *monitoring*. Alasan dari penggunaan sensor suhu DS18B20 dikarenakan sensor ini memiliki fitur *waterproof*. Serta pada penelitian yang dilakukan oleh Anggita Nur Fathoni yang berjudul Blackbox Testing terhadap Prototipe Sistem Monitoring Kualitas Air Berbasis IoT, didapatkan hasil pembacaan dari sensor suhu jenis ini memiliki tingkat akurasi rata-rata pada angka 99,02 % [24]. Hal ini menunjukkan sensor Suhu DS18B20 *waterproof* ini dapat mengukur suhu air dengan baik.

## 2.8 Real Time Clock (RTC)

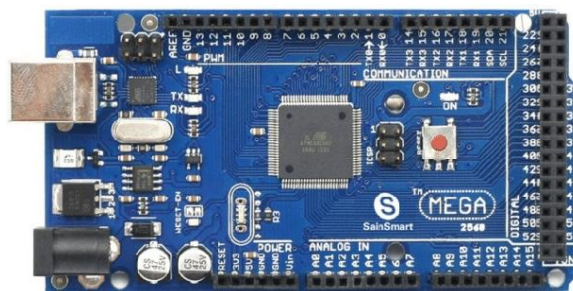
RTC merupakan komponen elektronika yang mampu mendeteksi dan dapat mengakses data waktu serta data kalender berupa detik, menit, jam, tanggal, bulan dan tahun secara real time [25]. Komponen RTC memiliki fungsi pada penelitian ini sebagai notif waktu jeda pada beberapa kondisi ketika sistem pengisian ulang larutan nutrisi berjalan, seperti kondisi pengosongan, pembacaan sensor pada saat proses penyesuaian larutan nutrisi dan sebagainya.



Gambar 2. 8 RTC DS3231  
Sumber : (arduino.rezaervani.com)

## 2.9 Arduino Mega 2560

Arduino adalah sebuah komponen elektronika bersifat *open-source project* yang dapat menciptakan perangkat berbasis mikrokontroler bertujuan untuk merakit perangkat digital bahkan peralatan interaktif yang dapat mendeteksi serta mengontrol suatu perangkat fisik. Arduino Mega merupakan *board* yang dilengkapi dengan mikrokontroler Atmega [26]. Arduino Mega berfungsi sebagai mikrokontroler pada sistem automasi hidroponik ini secara keseluruhan.



Gambar 2. 9 Arduino Mega 2560  
Sumber : (researchgate.net)

## 2.10 Pompa Air

Pompa air merupakan suatu alat yang memiliki fungsi untuk memindahkan cairan dari suatu tempat ke tempat lainnya dengan cara menaikkan tekanan pada cairan tersebut melalui sumber tenaga [27].



Gambar 2. 10 Pompa Air Kecil  
Sumber : (biggo.id)



Gambar 2. 11 Pompa Air Besar  
Sumber : (berita.99.co)

Pompa Air kecil digunakan untuk memompa larutan nutrisi A & B *Mix* dari penyimpanan ke wadah air nutrisi. Sedangkan pompa air besar digunakan untuk menyalurkan air nutrisi yang telah tercampur pada wadah air nutrisi ke pipa-pipa tanaman hidroponik, *Mixing* cairan nutrisi serta pengisian ulang air nutrisi.

## 2.11 Nutrisi A-B Mix

Nutrisi A-B *Mix* adalah jenis pupuk racikan yang mengandung unsur makro dan mikro terkombinasi, yang berfungsi sebagai nutrisi bagi tanaman. Jenis nutrisi A-B *Mix* ini banyak digunakan pada pertanian hidroponik [28].



Gambar 2. 12 Pupukl Nutrisi A-B *Mix*  
Sumber : (dadimakmur.com)

## 2.12 LED UV

LED *Ultraviolet* (UV) atau *growing light* merupakan suatu komponen semikonduktor yang memancarkan cahaya monokromatik tidak koheren ketika diberi tegangan. Lampu jenis ini dapat meningkatkan proses pertumbuhan pada tanaman sehingga memberikan produksi yang tergolong optimal pada pertanian hidroponik [29]. Keuntungan penggunaan lampu LED jenis ini diantaranya adalah memiliki spektrum cahaya yang kecil, lalu konsumsi daya listrik serta panas yang dihasilkan oleh lampu jenis ini juga lebih rendah dibandingkan dengan lampu jenis lainnya seperti lampu neon dan lampu pijar [30]



Gambar 2. 13 LED UV (*Growing light*)  
Sumber : (dinomarket.com)

Penggunaan LED UV pada proyek ini adalah sebagai sumber cahaya tambahan bagi tanaman hidroponik, dimana LED UV akan memenuhi kekurangan cahaya matahari yang didapat tanaman selama proses fotosintesis.

### **2.13 *Internet of things* (IoT)**

*Internet of things* (IoT) adalah salah satu jenis sistem teknologi yang dimana dapat mengubah semua aktifitas yang dilakukan oleh manusia, dapat dikendalikan atau dipantau melalui jaringan internet [6]. *Internet of things* juga merupakan konsep objek sehari-hari yang terhubung dengan internet serta mampu mengidentifikasi ke perangkat lainnya. *Internet of things* ini terdiri dari dua bagian yaitu Internet sebagai sistem jaringan komputer yang terhubung menggunakan paket protocol internet standar (TCP/IP) untuk dapat melayani pengguna serta Internet yang menjadi kontrol konektivitas dan Things adalah objek atau perangkat [31]. Secara mudahnya prinsip dari teknologi IoT ini adalah media yang dapat mempermudah kegiatan sehari-hari yang biasanya dilakukan oleh manusia dalam banyak bidang kehidupan seperti lingkungan, kesehatan, pertanian, individu, pemerintah, otomasi rumah dan sejenisnya [32]. Teknologi IoT ini dibidang mampu mempermudah kegiatan manusia sehari-hari dikarenakan alat atau mesin yang

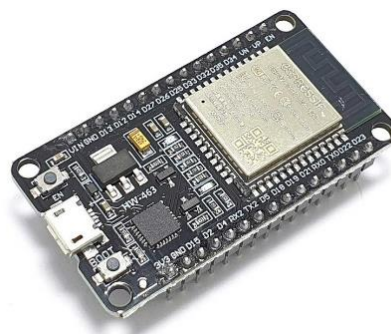
terhubung dengan teknologi IoT akan bekerja secara otomatis dan tentunya dapat dipantau [33].



Gambar 2. 14 IoT  
Sumber : (e2consulting.co.id)

#### 2.14 ESP32

ESP32 merupakan komponen mikrokontroler yang memiliki keunggulan, mulai dari pin out dan pin analog yang dimiliki lebih banyak, memori yang tersedia lebih besar, terdapat opsi *bluetooth 4.0 low energy* serta modul WiFi yang sudah tersedia di *board* mikrokontroler ESP32 ini yang akan berguna untuk mengaplikasikan konsep teknologi *Internet of things* (IoT) [34]. Pada proyek ini, komponen ESP32 ini berfungsi untuk penghubung antara data sensor yang terbaca pada mikrokontroler Arduino ke *firebase* serta aplikasi melalui komunikasi serial, dan juga menjadi mikrokontroler yang mengontrol penghidupan LED UV pada sistem.



Gambar 2. 15 ESP32  
Sumber : (ecadio.com)

## 2.15 *Firebase*

*Firebase* database adalah suatu wadah atau media penyimpanan basis data berupa non SQL yang dapat memungkinkan menyimpan beberapa tipe data seperti *String, long, Boolean*, dan sebagainya. Ada dua solusi database berbasis cloud yang dapat diakses client diberika oleh *Firebase*, yaitu *Realtime database* dan *Cloud Firestore* [33]. Berdasarkan fungsinya kontribusi *firebase* pada proyek akhir ini adalah menyimpan data sensor yang terbaca pada mikrokontroler untuk dapat diakses pada aplikasi atau *website monitoring* ataupun kontrol melalui internet.



Gambar 2. 16 *Firebase*  
Sumber : (*firebase.google.com*)

## 2.16 *Thunkable*

*Thunkable* adalah salah satu aplikasi dengan *tools* IDE yang bersifat open source, yang menyediakan pembuatan aplikasi yang dapat digunakan pada Android dan iOS. Pada aplikasi ini *tools* yang digunakan berupa sistem *block programming*, sehingga para pengguna tidak perlu lagi untuk membuat kode program secara manual untuk membuat tampilan atau fungsi pada aplikasi [35]. Pada proyek ini, *Thunkable* berperan dalam pembuatan *interface* serta aplikasi *HYDRO-JET*. Tampilan pada aplikasi berupa bar untuk menampilkan nilai volume tendon, tds larutan, intensitas cahaya, pH dan juga suhu pada larutan nutrisi.

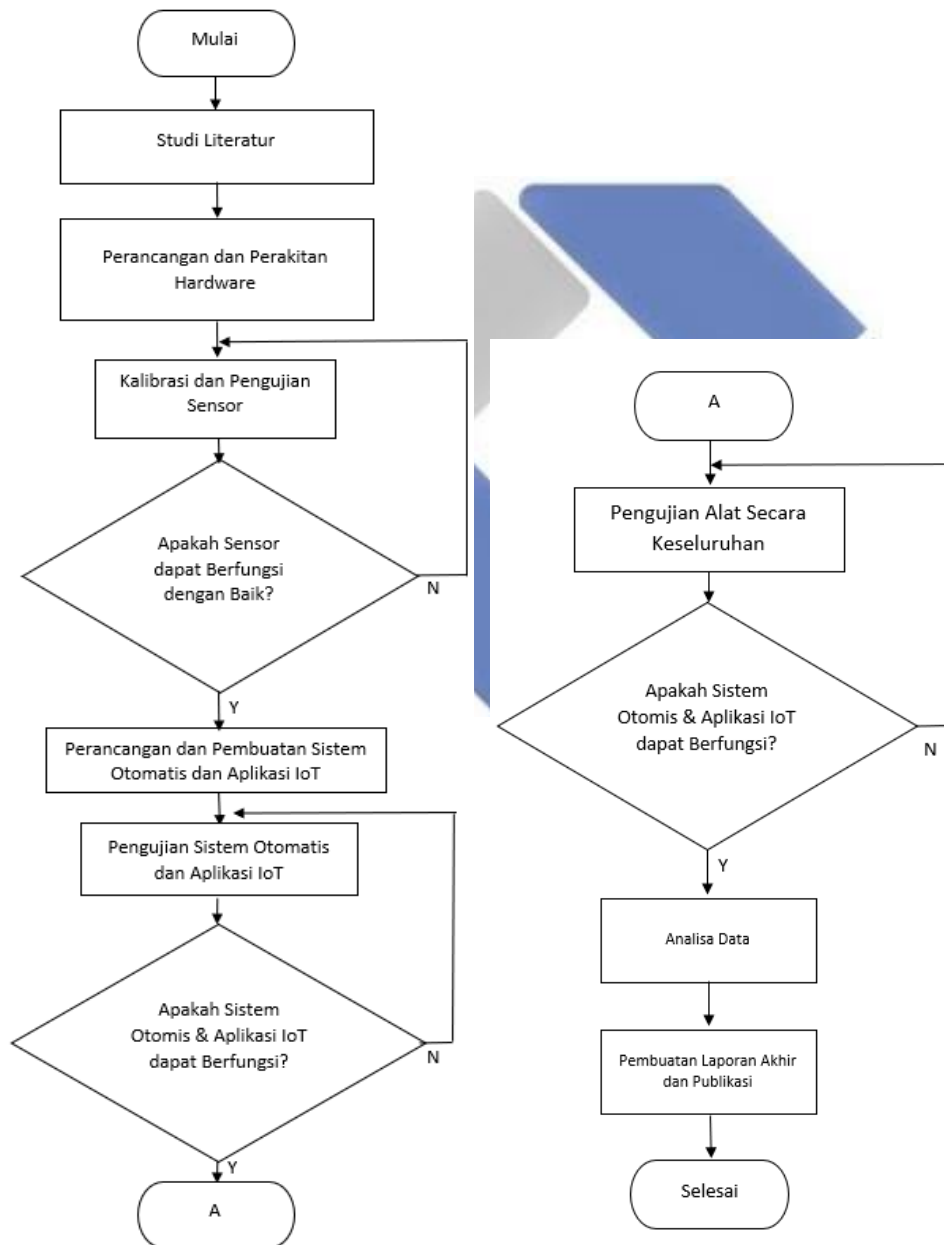


Gambar 2. 17 *Thunkable*  
Sumber : (*Thunkable.co*)



### BAB III METODE PELAKSANAAN

Pada bab ini membahas mengenai metode pelaksanaan yang akan dilakukan selama proses pengerjaan proyek akhir yang digambarkan seperti diagram *flowchart* di bawah ini:



Gambar 3. 1 Flowchart Pelaksanaan Pembuatan Proyek Akhir

### **3.1 Studi Literatur**

Tahap ini merupakan tahap mengidentifikasi permasalahan tentang implementasi konsep kebun cerdas pada perkebunan hidroponik yang terintegrasi IoT. Pengumpulan informasi dengan membaca dan mempelajari buku-buku atau jurnal penelitian terkait untuk mengetahui bagaimana konsep dari kebun cerdas, serta bagaimana implementasinya pada perkebunan hidroponik.

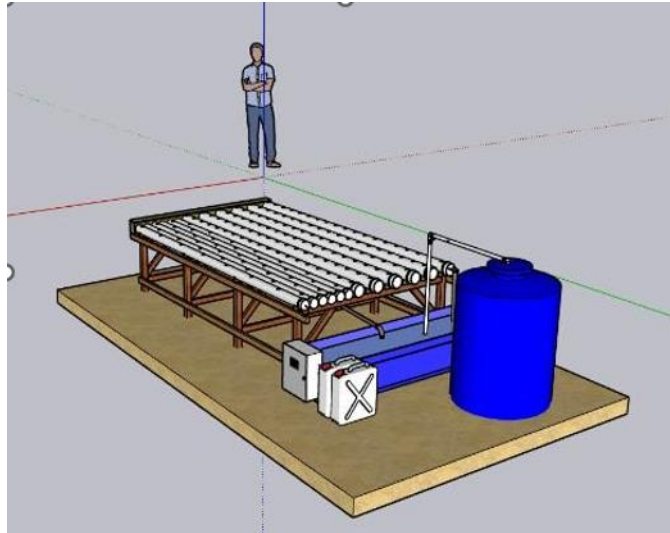
Pencarian informasi mengenai komponen juga penting agar proses pembuatan proyek akhir dapat berjalan lancar dan terstruktur. Informasi-informasi tersebut seperti data jenis sensor, pompa air, mikrokontroler, serta *software* yang digunakan untuk membuat aplikasi *smartphone* nantinya.

### **3.2 Perancangan dan Perakitan *Hardware***

Pada tahap ini dilakukan perancangan untuk konstruksi modul hidroponik dan rangkaian elektrik. Setelah proses perancangan, akan dilanjutkan dengan proses perakitan *hardware*.

#### **3.2.1 Desain Modul Hidroponik**

Proses perancangan desain konstruksi modul hidroponik dilakukan menggunakan aplikasi *sketch up*. Pada tahap ini ditentukan jenis bahan yang akan digunakan serta ukuran dari modul hidroponik. Hasil dari proses desain modul hidroponik ini, didapati ukuran dari meja hidroponik yang akan digunakan dengan panjang 200 cm dan lebar 80 cm. Media pengaliran air tanaman yang akan digunakan adalah pipa PVC. Bahan yang akan digunakan untuk membangun meja hidroponik adalah baja ringan kanal C. Berikut merupakan desain alat pada aplikasi *sketch up*.



Gambar 3. 2 Desain Awal Alat

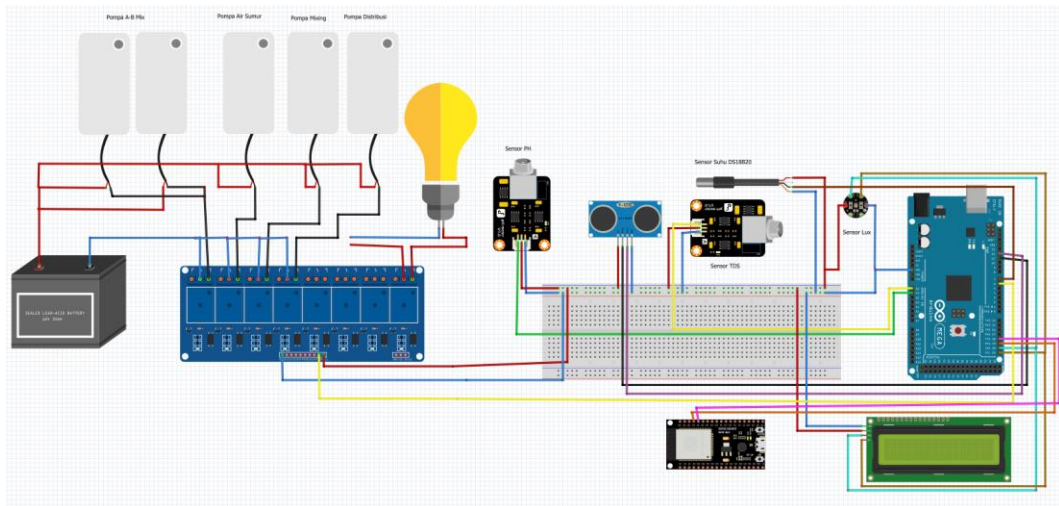
### 3.2.2 Perakitan Modul Hidroponik

Tahap ini merupakan tahap perakitan konstruksi modul hidroponik yang telah di desain sebelumnya. Proses perakitan diawali dengan pemotongan baja ringan, yang kemudian akan disambungkan hingga membentuk meja dengan ukuran 80 cm x 200 cm. Proses dilanjutkan dengan pemotongan pipa PVC yang kemudian akan dilubangi dengan ukuran  $\frac{3}{4}$  inc.

Lubang ini akan digunakan sebagai tempat menaruh tanaman. Berikut merupakan bukti dokumentasi saat perakitan modul hidroponik, serta hasil akhir perakitan.

### 3.2.3 Desain Rangkaian Elektrik

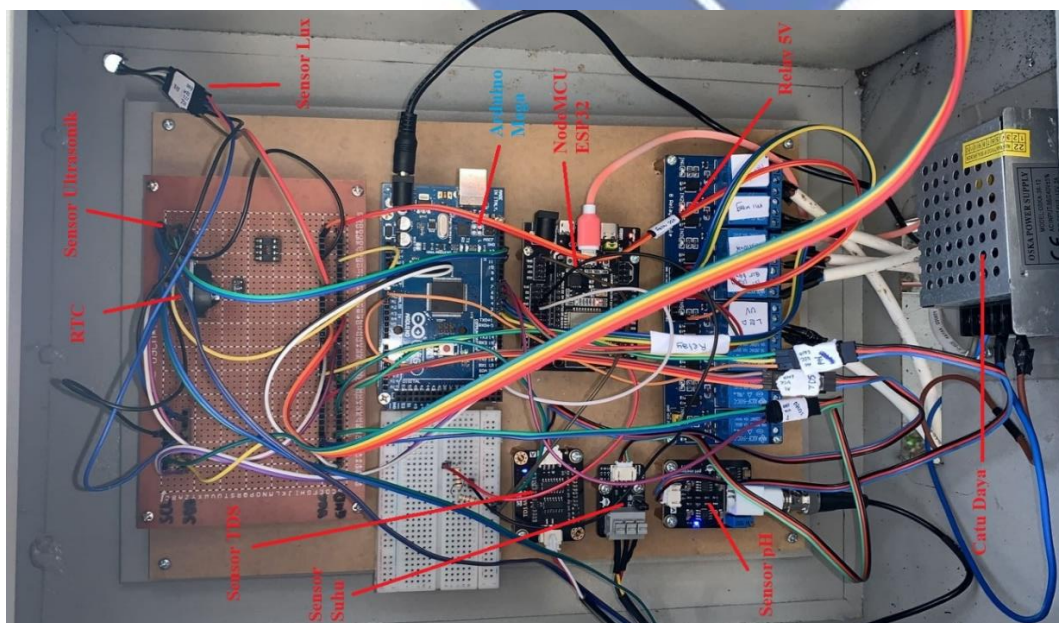
Pada tahap ini dilakukan perancangan rangkaian elektrik yang bertujuan mengatur tata letak dari *wiring* diagram masing-masing komponen elektronika. Perancangan rangkaian elektrik menggunakan aplikasi *fritzing*. Berikut ini desain hasil rancangan rangkaian elektrik.



Gambar 3. 3 Desain Rangkaian Elektrik

### 3.2.4 Perakitan Rangkaian Elektrik

Tahap ini diawali dengan perakitan rangkaian sensor pada Arduino, kemudian dilanjutkan dengan perakitan rangkaian pompa air pada *relay* 8 channel, dilanjutkan dengan pemasangan kabel dari LCD dan ESP32 pada Arduino Mega, dan diakhiri dengan penempatan rangkaian elektrik pada kotak panel. Berikut merupakan hasil dari perakitan rangkaian elektrik.



Gambar 3. 4 Perakitan Komponen ke dalam Kotak Panel

### 3.3 Pengujian dan Kalibrasi Sensor

Pada tahap ini merupakan tahap untuk pengkalibrasian sensor untuk mendapatkan hasil pembacaan yang sama atau mendekati pembacaan dengan alat ukur. Kemudian dilanjutkan pada tahap pengujian sensor yang bertujuan untuk mendapatkan nilai akurasi dari semua sensor. Metode yang digunakan dalam proses kalibrasi sensor adalah metode regresi linier. Hasil dari metode regresi linier adalah berupa persamaan dari hubungan dua buah *variable* hasil pengujian. Persamaan ini akan digunakan untuk konversi nilai pembacaan sensor. Berikut merupakan rumus menghitung nilai akurasi pembacaan sensor.

$$Error = \frac{|Pembacaan Sensor - Pembacaan Alat Ukur|}{Pembacaan Sensor} \times 100\%$$

$$Akurasi = 100\% - Error$$

### 3.4 Perancangan dan Pembuatan Sistem Otomatis dan aplikasi IoT

Tahap ini merupakan tahap membuat program untuk sistem otomatis dan pembuatan aplikasi IoT. Sistem otomatis akan dibuat pada pemrograman Arduino, sedangkan pembuatan aplikasi IoT akan menggunakan *Thunkable*.

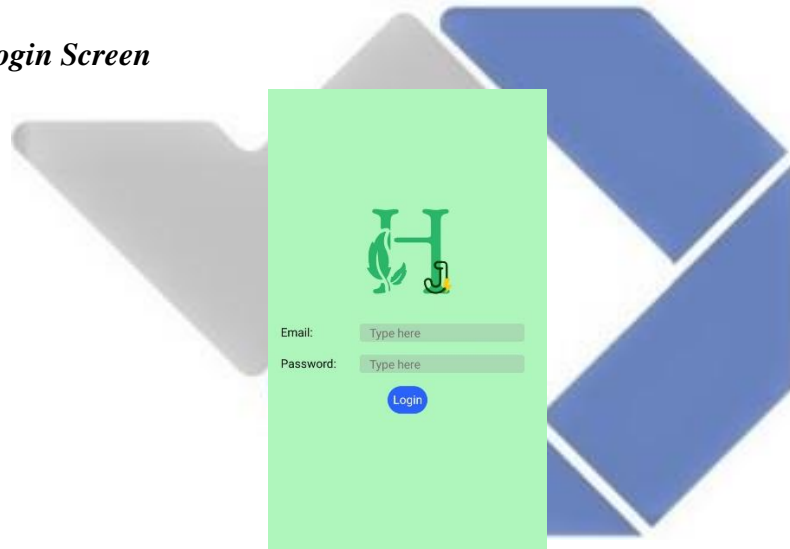
#### 3.4.1 Desain Software

Desain *software* dari sistem *monitoring* larutan nutrisi dan kebutuhan cahaya tanaman, serta kontrol LED UV terdiri atas pemrograman Arduino mega 2560 untuk membaca semua data sensor serta terhubung ke *relay* LED UV yang akan dikontrol, dan juga pengiriman data ke ESP32 melalui proses komunikasi serial. Lalu pada ESP32 pemrograman yang dilakukan berguna untuk mengirim hasil data dari proses komunikasi serial Arduino mega 2560 ke *Firestore Realtime database*. Selanjutnya, membuat desain aplikasi tampilan *interface* pada device *smartphone* menggunakan *Thunkable* untuk menampilkan nilai PPM, pH, suhu, dan volume pada larutan nutrisi beserta intensitas cahaya pada sistem perkebunan hidroponik berikut dengan kontrol hidup dan mati LED UV.

### 3.4.2 Pembuatan Software

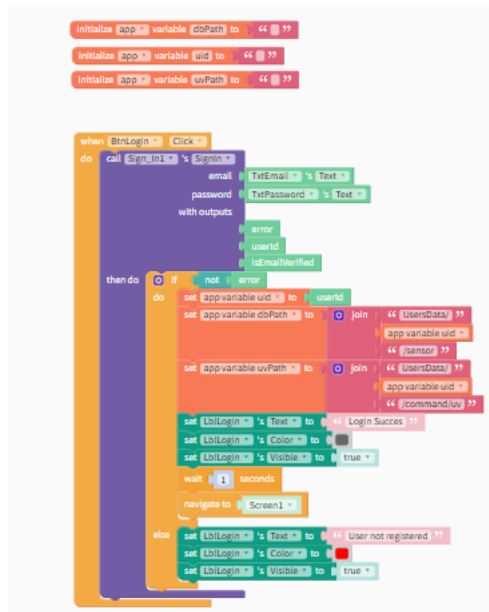
Pada proses pembuatan *software*, tahapan pertama yang dilakukan adalah membuat program pada mikrokontroler. Pemograman ini bertujuan untuk membaca data sensor ultrasonik, lux, pH, TDS, dan suhu yang nantinya akan dikirimkan ke ESP32 melalui komunikasi serial. Setelah itu pembuatan program sistem pompa air untuk pengisian ulang larutan nutrisi dan LCD 20 x 4. Lalu membuat program pada ESP32 untuk melakukan transaksi data antara Arduino mega 2560 dan *Firestore Realtime database*. Dan pada tahap akhir dilakukan pemrograman pada *Thunkable* untuk pembuatan aplikasi *monitoring* larutan nutrisi dan intensitas cahaya serta kontrol LED UV pada perkebunan hidroponik.

### 3.4.3 Login Screen



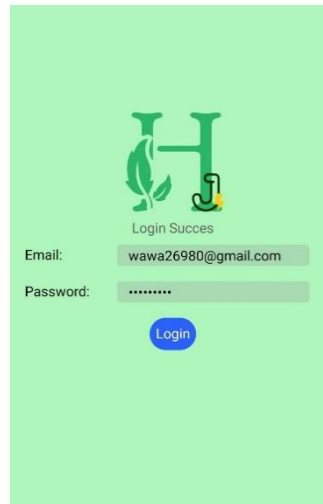
Gambar 3. 5 Login screen Aplikasi *HYDRO-JET*

Gambar diatas merupakan tampilan awal dari aplikasi *HYDRO-JET*. Elemen yang digunakan pada *screen login* ini terdiri dari satu blok gambar (logo aplikasi *HYDRO-JET*), empat label (label *email*, label *text email*, label *password*, dan label *text password*), dan satu tombol (*login button*).



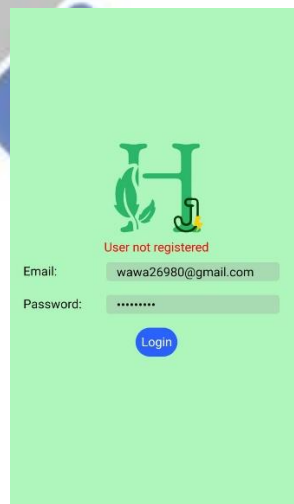
Gambar 3. 6 Block Login Screen Aplikasi HYDRO-JET

Pada *Logic Block* tampilan *login*, hal pertama yang dilakukan adalah inialisasi *variable dbPath* (semua data yang akan ditampilkan) serta inialisasi *variable uid* (*user id* yang telah terdaftar pada *firebase*). Lalu pada kondisi *button Login* di tekan maka akan terjadi pemanggilan sesi *sign in* pada *firebase*, lalu akan sekaligus mengkonfirmasi alamat *email* dan *password* yang ditulis didalam label *text email* dan *label text password*. Setelah itu *output user id* “*error*” / “*apakah email terverifikasi*” akan disiapkan untuk dipanggil. Kemudian proses selanjutnya, *variable uid (userID)* pada *firebase* akan dipanggil untuk disinkronkan dengan *userID* yang tertulis pada *text email* dan *text password* tampilan *login*. Dalam waktu yang bersamaan juga memanggil *variable dbPath* yang merupakan *address variable* tersebut di *firebase*. Ketika sinkronisasi *uid* tidak *error*, maka *set text info* menjadi “*Login Succes*” dan berwarna hitam. Namun jika sinkronisasi *uid error*, *set text info* menjadi “*User not registered*” dan berwarna merah. Setelah proses itu, sistem akan diatur untuk beralih ke *content screen* dengan menunggu selama 1 detik. Perpindahan dari tampilan *login* ke *content screen* hanya bisa terjadi ketika sinkronisasi *user id* tidak *error*.



Gambar 3. 7 Tampilan Berhasil *Login*

Pada gambar 3.7 merupakan tampilan aplikasi ketika *email* dan *password* yang diberikan sesuai pada saat proses sinkronisasinya, sehingga muncul tulisan '*Login Succes*'.



Gambar 3. 8 Tampilan Gagal *Login*

Sedangkan gambar 3.8 adalah tampilan aplikasi ketika *email* dan *password* yang diberikan tidak sesuai ketika melakukan sinkronikasi, maka muncul tulisan '*User not registered*' berwarna merah.

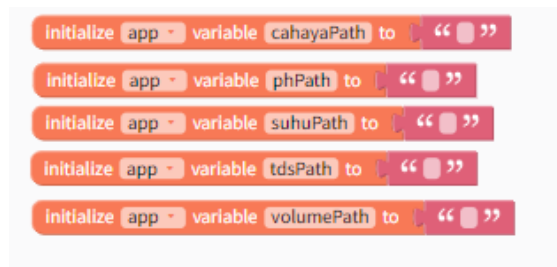


### 3.4.4 Content Screen



Gambar 3. 9 Content Screen Aplikasi *HYDRO-JET*

*Content screen* merupakan tampilan aplikasi yang menyediakan hasil baca data sensor yang digunakan pada sistem hidroponik sekaligus kontrol yang bisa pengguna pakai saat menggunakan aplikasi. Data yang ditampilkan pada *screen* ini adalah, volume tendon nutrisi, nilai PPM, pH dan suhu pada larutan nutrisi, serta nilai intensitas cahaya yang ada di lingkungan sistem hidroponik. Untuk sistem kontrol yang dapat dikendalikan melalui aplikasi adalah, *on*, dan *off* LED UV. Kontrol LED UV dilakukan secara manual, berdasarkan data yang diperoleh dari nilai intensitas cahaya. Elemen yang diperlukan dalam menyusun tampilan ini, yakni, 16 label (label judul, label VOLUME, label data volume, label satuan “liter”, label TDS, label data sensor tds, label satuan “PPM”, label INTENSITAS CAHAYA, label data sensor lux, label satuan “lux”, label PH, label data sensor PH, label SUHU, label data sensor suhu, label satuan “°C”, dan label LED UV) dan 1 *switch* (untuk kontrol LED UV).

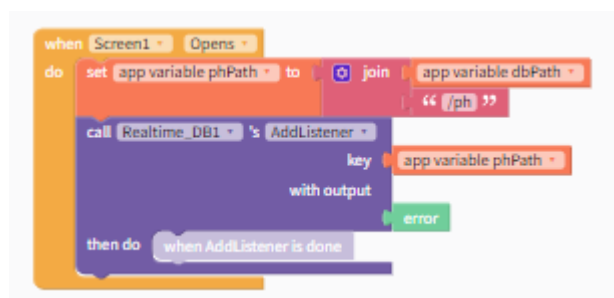


Gambar 3. 10 *Block* Inisialisasi Semua Data Dari *Firebase*

Data-data yang ditampilkan di *content screen* ini diambil melalui proses pertama berupa inisialisasi *variable* dari *firebase* (*cahayatPath*, *phPath*, *suhuPath*, *tdsPath*, *volumePath*) untuk diambil datanya ke *Thunkable* sesuai dengan *block* diatas. Setelah itu proses selanjutnya *Thunkable* memanggil data dari *firebase* ke label *path*.

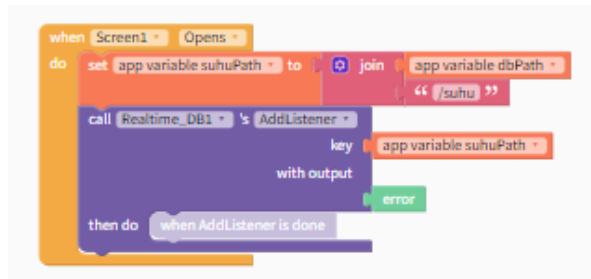


Gambar 3. 11 *Block* Memanggil Data Sensor Cahaya Ke *Thunkable*  
*Block* diatas merupakan kondisi dimana ketika *Thunkable* memanggil *realtime database* dengan *key variable cahayaPath* untuk dimasukkan ke dalam *variable cahayaPath* dengan *address/cahaya*, ketika *Screen1* (*screenContent*) dibuka.



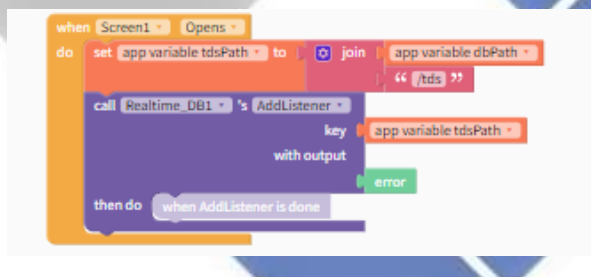
Gambar 3. 12 *Block* Memanggil Data Sensor pH Ke *Thunkable*

Block diatas merupakan kondisi dimana ketika *Thunkable* memanggil *realtime database* dengan *key variable phPath* untuk dimasukkan ke dalam *variable phPath* dengan *address /ph*, ketika *Screen1 (screenContent)* dibuka.



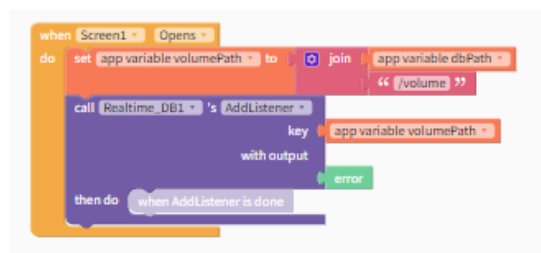
Gambar 3. 13 Block Memanggil Data Sensor Suhu Ke *Thunkable*

Block diatas merupakan kondisi dimana ketika *Thunkable* memanggil *realtime database* dengan *key variable suhuPath* untuk dimasukkan ke dalam *variable suhuPath* dengan *address /suhu* ketika *Screen1 (screenContent)* dibuka.



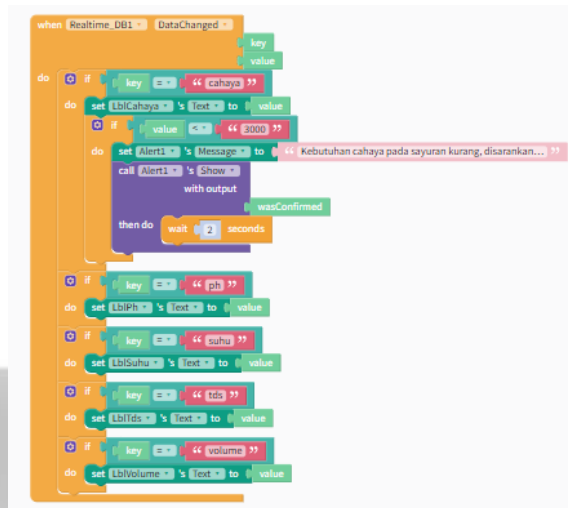
Gambar 3. 14 Block Memanggil Data Sensor TDS Ke *Thunkable*

Block diatas merupakan kondisi dimana ketika *Thunkable* memanggil *realtime database* dengan *key variable tdsPath* untuk dimasukkan ke dalam *variable tdsPath* dengan *address /tds*, ketika *Screen1 (screenContent)* dibuka.



Gambar 3. 15 Block Memanggil Data Volume Ke *Thunkable*

Block diatas merupakan kondisi dimana ketika *Thunkable* memanggil *realtime database* dengan *key variable volumePath* untuk dimasukkan ke dalam *variable volumePath* dengan *address /volume*, ketika *Screen1 (screenContent)* dibuka.

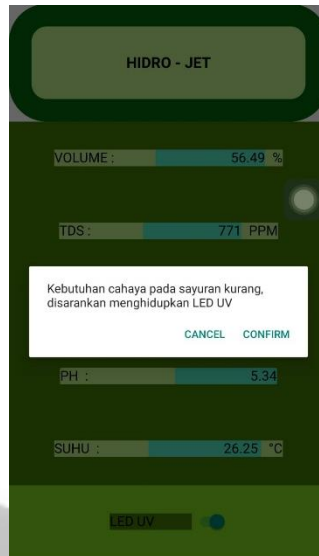


Gambar 3. 16 Block Menampilkan Semua Data Sensor Ke Path *Thunkable*

Block ini memiliki fungsi untuk *update* data yang masuk dari *firebase* agar dapat ditampilkan secara *realtime* pada *screen content*. Setiap nilai sensor yang ditampilkan pada label di *Thunkable* diambil dari *variable* yang tersedia pada *firebase*. Pada saat pengaktifan aplikasi, *Thunkable* akan mengambil data dari *firebase* yang sudah dikaitkan dalam bentuk *key* dan *value*. *Key* adalah nama *variable* yang ada pada *firebase*, sedangkan *value* adalah nilai dari *key* itu sendiri sesuai dengan *variable key* yang diambil. Maka ketika ingin memanggil nilai salah satu *key* pada suatu label *path*, maka hal yang perlu dilakukan hanya menginisialisasi *key* apa yang kita perlu panggil beserta memberikan *value* nya, dapat dilihat pada *block* gambar diatas. Ketika *LblPh* ingin mengambil nilai *ph* dari *firebase*, maka yang ditambah dipanggil adalah *key* nya dengan nama *variable* "ph" (bersesuaian dengan yang ada di *firebase*, lalu setelah itu *value* dari *key* yang awalnya berupa *text* tersebut dipanggil ke dalam *LblPh* menjadi bentuk *value*

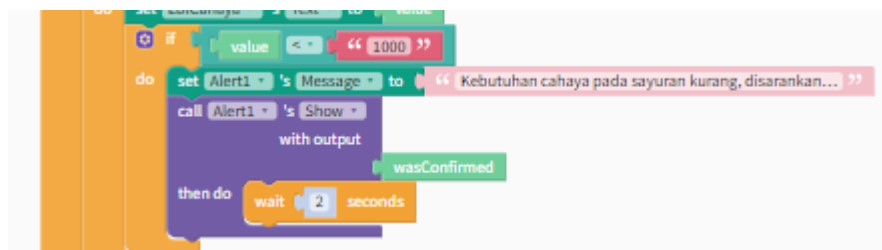
kembali. Hal yang sama juga terjadi pada *label path* lainnya dalam proses mengambil data sensor dari *firebase*.

### 3.4.5 Notifikasi Kebutuhan Cahaya



Gambar 3. 17 Notifikasi Saran Menghidupkan LED UV

Gambar diatas merupakan kondisi aplikasi pada saat memberikan notifikasi ketika mendeteksi bahwa cahaya pada lingkungan kebun hidroponik kurang, dengan ditandanya nilai lux di angka kurang dari 1000 lux. Pada sistem ini, munculnya notifikasi berdasarkan 2 tolak ukur yaitu gelap dan terang. Dan untuk mengetahui kondisi gelap terang tersebut melalui nilai intensitas cahaya yang diberikan oleh sensor lux.

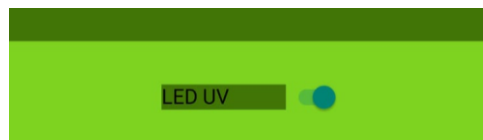


Gambar 3. 18 *Block* Notifikasi Saran Menghidupkan LED UV

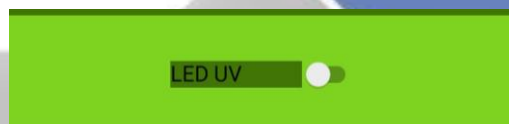
Pada blok ini, notifikasi bekerja pada saat membaca *value* dari variabel cahaya yang terukur di *LblCahaya* kurang dari 1000 lux, maka *Thinkable* akan *set*

sekaligus memanggil *Alert1* untuk memunculkan pesan bertuliskan “Kebutuhan cahaya pada sayuran kurang, disarankan menghidupkan LED UV”. Setelah proses memunculkan pesan telah di *confirm* oleh pengguna aplikasi, *Alert1* akan menunggu selama 2 detik sebelum bisa muncul kembali jika terdeteksi nilai cahaya masih kurang dari 1000 lux.

### 3.4.6 Kontrol LED UV

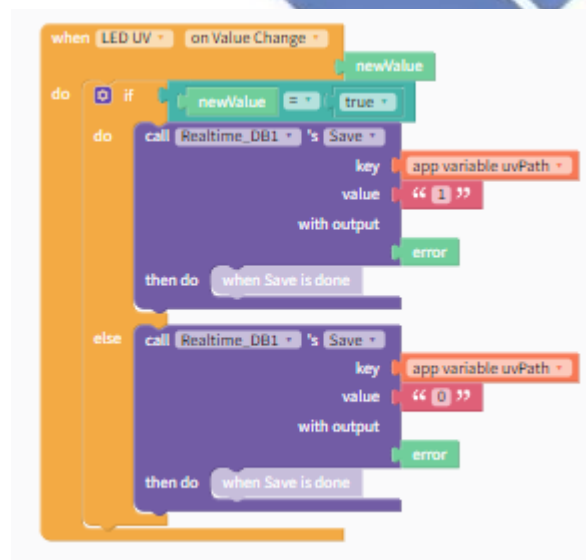


Gambar 3. 19 Kondisi *Switch* Ketika LED UV Hidup



Gambar 3. 20 Kondisi *Switch* Ketika LED UV Mati

Kontrol on-off LED UV pada *Thunkable* menggunakan elemen slide dengan nama “LED UV”. *Switch* ke posisi kanan menandakan LED UV sedang hidup, sedangkan slide berada di sebelah kiri menandakan LED UV dalam keadaan mati.



Gambar 3. 21 *Block* Program Kontrol Lampu *Growlight*

Jika *switch* berada disisi kanan akan memanggil *realtime database* dan menyimpan variabel *uvPath* dengan *value* 1 (*value* 1 pada program arduino akan mengaktifkan *relay* yang terhubung ke LED UV). Sedangkan ketika *switch* berada di posisi kiri akan memanggil *realtime database* dan menyimpan variabel *uvPath* dengan *value* 0 (*value* 0 pada program arduino akan menonaktifkan *relay* yang terhubung ke LED UV).

### 3.4.7 Pembuatan Sistem Pengisian Air dan Nutrisi Otomatis

Sistem ini bekerja hanya pada saat sensor ultrasonik mendeteksi jarak sensor dengan air ketika volume air 30 liter, atau mudahnya sistem otomatis ini bekerja ketika volume air sudah mencapai 30 liter. Ketika persyaratan tersebut sudah terpenuhi maka program akan menjalankan fungsi “isiUlang()”, yang dimana fungsi ini merupakan tahap awal pengisian air serta pembuatan larutan nutrisi secara otomatis.

```
//KONTROL ISI ULANG AIR
// -Air Penuh (CEK POINT)
if (distance >= 25.88 && distance <= 28.97) { //70 liter = 28.97
| cekUlang();
}

// -Air Habis
else if (distance >= 42.57) { //30 liter = 42.57 ; 50 liter = 35.63
| wPengosongan = true;
| isiUlang();
}

//Harus distribusi
else if (distance < 25.88) { //80 liter = 25.88
| distribusi();
}
```

Gambar 3. 22 Program Arduino Pengisian Air dan Nutrisi Otomatis

Pada saat sistem otomatis ini bekerja, program akan mematikan pompa distribusi dan memasuki proses pengosongan dengan waktu yang ditentukan. Pada proses pengosongan, program tidak menjalankan apapun kecuali menunggu proses ini selesai, yang bertujuan untuk membuang sisa air pada pipa tanaman ke dalam tandon larutan nutrisi. Setelah itu, program akan menghidupkan pompa yang memindahkan air dari sumber air bersih ke tandon nutrisi hidroponik sampai ke titik

yang sudah ditentukan. Pada proyek ini, pompa air bersih akan mati ketika volume air sudah mencapai 70 liter, sekaligus juga pada titik ini menjadi cek poin untuk menentukan harus melakukan penambahan nutrisi AB *Mix* atau pengenceran larutan.

```
void isiUlang(){
    blmDisimpan = true;

    digitalWrite(pompa_distribusi, HIGH);
    delay(100);

    //step 1 (waktu Pengosongan)
    if (wPengosongan == true){
        pengosongan();
        wPengosongan = false;
    }

    //step 2 (penyesuaian nutrisi)
    //PPM kurang
    if (wPengosongan == false) {
        digitalWrite(pompa_air, LOW);
    }
}
```

Gambar 3. 23 Program Pada Saat Sistem Mengisi Ulang Air

```
void pengosongan() {
    simpan_data();

    while(yTerpenuhi == true){

        ambildata_komponen();
        logikaRTC();

        if (detikBerlalu > WAKTU_PENGOSONGAN){ //2 menit
            yTerpenuhi = false;
            blmDisimpan = true;
            detikBerlalu = 0;
        }

        //Pembacaan sensor
        display();
        delay(500);
    }
}
```

Gambar 3. 24 Program Untuk Memberikan Delay Waktu Pengosongan Pipa Tanaman

Penambahan atau pengenceran larutan nutrisi di tendon ditentukan dengan nilai PPM terakhir yang terukur pada sensor, sistem ini berjalan pada fungsi program “cekUlang()”. Pada program ketika nilai PPM lebih kecil dari 700 maka akan dilakukan penambahan nutrisi AB *Mix*. Penambahan nutrisi AB *Mix* akan dihitung melalui formula yang sudah ada pada program. Program akan menentukan



dulu berapa kurang PPM pada air dengan cara mengurangi 700 dengan nilai PPM yang terukur sekarang. Setelah itu program akan menghitung banyaknya nutrisi yang harus diberi berdasarkan kurangnya nilai PPM yang sudah didapatkan tadi, melalui cara membandingkannya dengan data pengaruh penambahan nutrisi AB *Mix* pada air. Nilai kebutuhan nutrisi AB *Mix* yang sudah didapatkan hanya berlaku ketika volume air 1 liter, maka proses selanjutnya, nilai tersebut akan dikalikan dengan total volume tandon nutrisi sekarang. Setelah mendapatkan total kebutuhan nutrisi, maka program akan membandingkannya dengan data debit pompa nutrisi AB *Mix*. Lalu setelah melalui semua proses itu, program akan menghidupkan pompa nutrisi sesuai dengan waktu yang sudah didapat dari perbandingan total kebutuhan nutrisi AB *Mix* dengan data debit pompa nutrisi AB *Mix*.

```

//PPM kurang
if (tdsValue < 700){
    waktu_penyesuaian();
    while(tdsValue < 700){
        ambildata_komponen();
        simpan_data();
        while(yTerpenuhi == true){

            ambildata_komponen();
            PPM_Sekarang = simpanPPM;

            //Perhitungan lama hidup pompa nutrisi
            kurangPPM = ketentuan - PPM_Sekarang;
            butuhPPM = (kurangPPM/hasil_nutrisi)*nutrisiAB;
            total_butuhPPM = butuhPPM*volume_sensor;
            Pompa_nutrisi_harus_hidup = (waktu_pompa_nutrisi*total_butuhPPM)/real_nutrisi_keluar;

            digitalWrite(pompa_nutrisi, LOW);
            logikaRTC();
            if (detikBerlalu > Pompa_nutrisi_harus_hidup){
                digitalWrite(pompa_nutrisi, HIGH);
                yTerpenuhi = false;
                bInDisimpan = true;
                detikBerlalu = 0;
            }
            display();
            delay(500);
        }
        //menunggu sampai teraduk rata
        waktu_penyesuaian();
        display();
        delay(500);
    }
}

```

Gambar 3. 25 Program Cek Point Apabila Kadar Nutrisi (PPM) Dalam Air kurang Dari Yang Ditetapkan

Ketika nilai PPM lebih besar dari 800 maka akan dilakukan pengenceran. Penambahan air bersih akan dihitung melalui formula yang sudah ada pada program. Program akan menentukan dulu berapa lebih nilai PPM pada air dengan cara mengurangi nilai PPM yang terukur sekarang dengan 700. Setelah itu program akan menghitung banyaknya air bersih yang harus diberi berdasarkan lebihnya nilai

PPM yang sudah didapatkan tadi, melalui cara membandingkannya dengan data pengaruh penambahan air bersih terhadap larutan nutrisi *AB Mix*. Niali kebutuhan air bersih yang sudah didapatkan hanya berlaku ketika volume air 1 liter, maka proses selanjutnya, nilai tersebut akan dikalikan dengan total volume tandon nutrisi sekarang.

Setelah mendapatkan total kebutuhan air bersih, maka program akan membandingkannya dengan data debit pompa isi ulang air bersih. Lalu setelah melalui semua proses itu, program akan menghidupkan pompa isi ulang air bersih sesuai dengan waktu yang sudah didapat dari perbandingan total kebutuhan air bersih dengan data debit pompa isi ulang air bersih. Namun pada proses ini, ketika penambahan air bersih sudah mencapai di titik volume 80 liter, maka sistem akan langsung mendistribusikan larutan nutrisi tersebut dengan nilai PPM berapa pun. Hal ini bertujuan agar air yang tersalurkan pada tandon tidak berlebihan ataupun tumpah dari tandon nutrisi.

```

//PPM lebih
//cuma dilakukan sekalipenyesuaian, dikarenakan batas tandon
else if (tdsValue > 800){
  ambildata_komponen();
  simpan_data();
  while(yTerpenuhi == true){

    ambildata_komponen();
    PPM_Sekarang = simpanPPM;

    //Perhitungan lama hidup pompa air
    lebihPPM = PPM_Sekarang - ketentuan;
    butuhAir = (lebihPPM/berkurang_PPM)*air;
    total_butuhAir = butuhAir*volume_sensor;
    Pompa_air_harus_hidup = (waktu_pompa_air*total_butuhAir)/real_air_keluar;

    digitalWrite(pompa_air, LOW);
    logikaRTC();
    if (detikBerlalu > Pompa_air_harus_hidup | distance < 25.88){ //ketika 80 liter, pompa air mati
      digitalWrite(pompa_air, HIGH);
      yTerpenuhi = false;
      blmDisimpan = true;
      detikBerlalu = 0;
    }
    display();
    delay(500);
  }
}
//menunggu sampai teraduk rata
waktu_penyesuaian();

display();
delay(500); }

```

Gambar 3. 26 Program Cek Point Untuk Pengencaran Kadar PPM

```

void waktu_penyesuaian() {
    simpan_data();
    while(yTerpenuhi == true){
        ambildata_komponen();
        logikaRTC();
        if (detikBerlalu > WAKTU_PENYESUAIAN){ //60 detik
            yTerpenuhi = false;
            blmDisimpan = true;
            detikBerlalu = 0;
        }
        //Pembacaan sensor
        display();
        delay(500);
    }
}

```

Gambar 3. 27 Program Waktu Penyesuaian Bagi Pembacaan *Response time* Sensor TDS

Pada setiap proses penambahan dan pengenceran nutrisi akan ada proses menunggu selama kurang lebih 60 detik setelah itu, yang dimana proses ini bertujuan untuk sensor TDS membaca nilai PPM pada tandon nutrisi hingga ke titik yang stabil, sehingga program dapat mengambil keputusan kembali setelah memperoleh nilai PPM yang sudah stabil tersebut. Ketika program telah mendapatkan nilai PPM diantara 700 dengan 800 dari proses otomatis yang telah dilakukan, maka program akan menghidupkan kembali pompa distribusi untuk menyalurkan larutan nutrisi hasil pengisian otomatis tadi ke pipa tanaman.

```

//step 2 (ketika ppm sudah terpenuhi)
else if (tdsValue >= 700 && tdsValue <= 800){
    distribusi();
}

```

```

void distribusi () {
    digitalWrite(pompa_distribusi, LOW);
    digitalWrite(pompa_air, HIGH);
    digitalWrite(pompa_nutrisi, HIGH);
    detikBerlalu = 0;
    blmDisimpan = true;
}

```

Gambar 3. 28 Program Untuk Melakukan Distribusi Air Nutrisi Setelah Pengisian Ulang

### **3.5 Pengujian Sistem Otomatis dan Aplikasi IoT**

Pada tahap ini dilakukan untuk melihat apakah sistem otomatis dan aplikasi IoT yang telah dibuat, apakah dapat bekerja sesuai dengan prosedur yang diinginkan. Pengujian sistem otomatis dilakukan dengan melihat nilai PPM dari hasil pengisian ulang air dan nutrisi pada bak hidroponik, dengan kadar PPM yang diinginkan adalah 600-800 PPM. Kemudian untuk pengujian Aplikasi IoT, yang akan diuji adalah kemampuan aplikasi untuk dapat menampilkan data pembacaan sensor, serta dapat mengontrol hidup mati lampu *Growlight* melalui *smartphone*.

### **3.6 Pengujian Alat Secara Keseluruhan**

Pada tahap ini, dilakukan pengujian terhadap keseluruhan sistem kontrol dan *monitoring* pada perkebunan hidroponik. Pengujian diawali dengan pengujian sistem *monitoring* pembacaan sensor melalui aplikasi pada *smartphone*. Hasil *monitoring* akan dibandingkan kembali dengan alat ukur untuk melihat tingkat keakuratan dari hasil pembacaan. Kemudian dilanjutkan dengan pengujian pada sistem pengisian air dan nutrisi otomatis. Pada pengujian ini, dilakukan pengamatan terhadap nilai PPM dari hasil pengisian air dan nutrisi secara otomatis, apakah nilai PPM termasuk pada *range* yang diinginkan, yaitu 600-800PPM. Pengujian terakhir adalah pengujian terhadap kontrol lampu *Growlight*, apakah lampu dapat dihidup matikan melalui aplikasi yang ada di *smartphone*.

### **3.7 Analisa Data**

Pada Tahap ini dilakukan analisa terhadap data hasil pengujian, apakah sistem yang telah dibuat dapat berjalan sesuai yang diinginkan. Apabila masih ditemukan kekurangan ataupun masalah pada sistem, *software*, maupun *hardware*, maka akan dilakukan perancangan ulang ulang untuk mendapatkan hasil yang diinginkan.

### **3.8 Pembuatan Laporan Akhir dan Publikasi**

Tahap pembuatan makalah ini merupakan tahap terakhir dalam pembuatan proyek akhir. Pada tahapan ini bertujuan untuk merangkum keseluruhan data yang berhubungan dengan proyek akhir dan memberikan informasi yang didapat pada proyek akhir yang telah dilakukan. Sedangkan publikasi akan menjadi bukti atau rekam jejak dari buah hasil penelitian yang telah dilakukan.

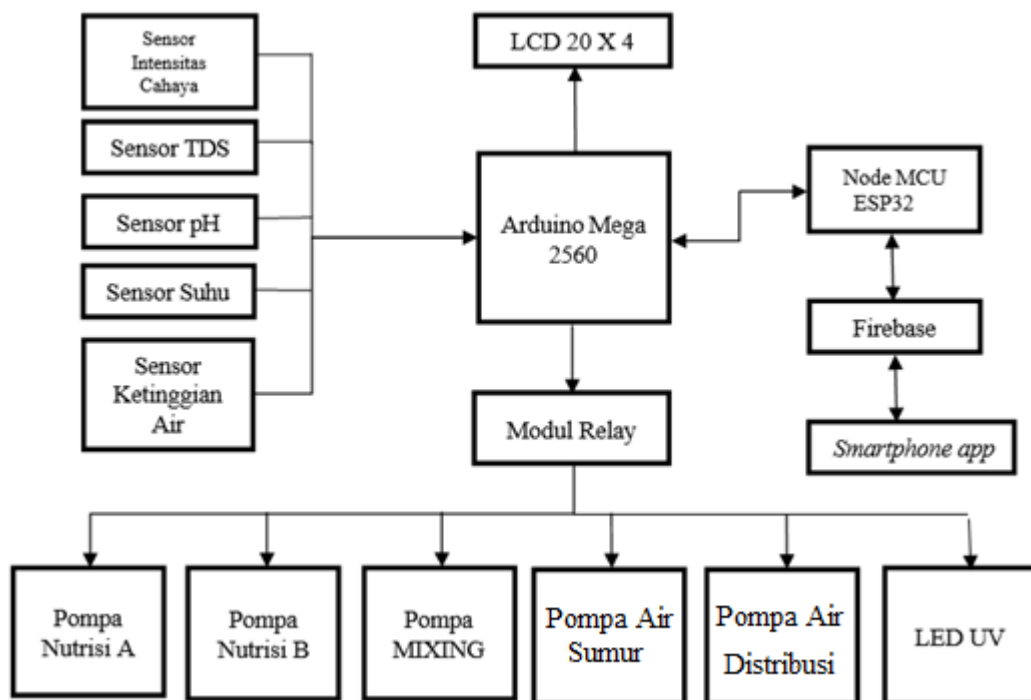
Proses ini merupakan proses membangun aplikasi *smartphone* yang akan digunakan untuk menampilkan data hasil pembacaan sensor dan pemberi kontrol perintah bagi lampu LED UV. Pada aplikasi ini akan disematkan pula fitur notifikasi apabila keadaan intensitas cahaya pada tanaman dibawah nilai standar.



## BAB IV PEMBAHASAN

### 4.1 Deskripsi Alat

Implementasi kebun cerdas pada perkebunan hidroponik merupakan sebuah konsep sistem yang diharapkan dapat melakukan *monitoring* setiap aspek yang dapat mempengaruhi pertumbuhan tanaman pada sistem hidroponik dari jarak jauh menggunakan *Internet of Things* (IoT). Beberapa dari aspek tersebut adalah kadar nutrisi dalam air (PPM), derajat keasaman (pH) air, volume air yang ada pada sistem hidroponik, suhu air, serta keadaan pencahayaan yang ada pada kebun hidroponik. Selain mampu untuk melakukan *monitoring*, alat ini mampu untuk mengisi ulang air dan nutrisi pada bak hidroponik secara otomatis, dengan hasil kadar PPM air yang diinginkan. Berikut ini adalah blok diagram untuk sistem kontrol dan *monitoring* pada perkebunan hidroponik yang terintegrasi IoT



Gambar 4. 1 Blok Diagram Alat

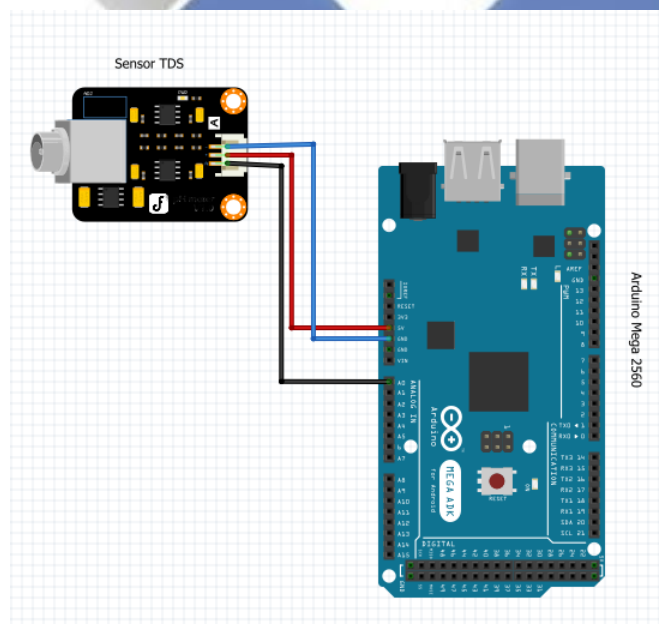
Pada pembuatan proyek ini, digunakan 5 jenis sensor yang berfungsi untuk melakukan *monitoring*, penambahan kontrol *growlight* yang dapat dikontrol dari jarak jauh. Penggunaan lampu *growlight* bertujuan untuk memenuhi kebutuhan cahaya matahari bagi tanaman, sehingga tanaman dapat bertumbuh secara maksimal.

## 4.2 Proses Kalibrasi dan Pengujian Sensor

Pada tahap ini dilakukan proses kalibrasi dan pengujian terhadap sensor-sensor yang akan digunakan pada sistem. Proses kalibrasi bertujuan untuk mendapatkan hasil pembacaan sensor yang akurat dengan melakukan perbandingan pembacaan sensor dengan pembacaan pada alat ukur. Sensor tersebut adalah sensor TDS, sensor pH, sensor suhu DS18B20, sensor lux, dan sensor ultrasonik.

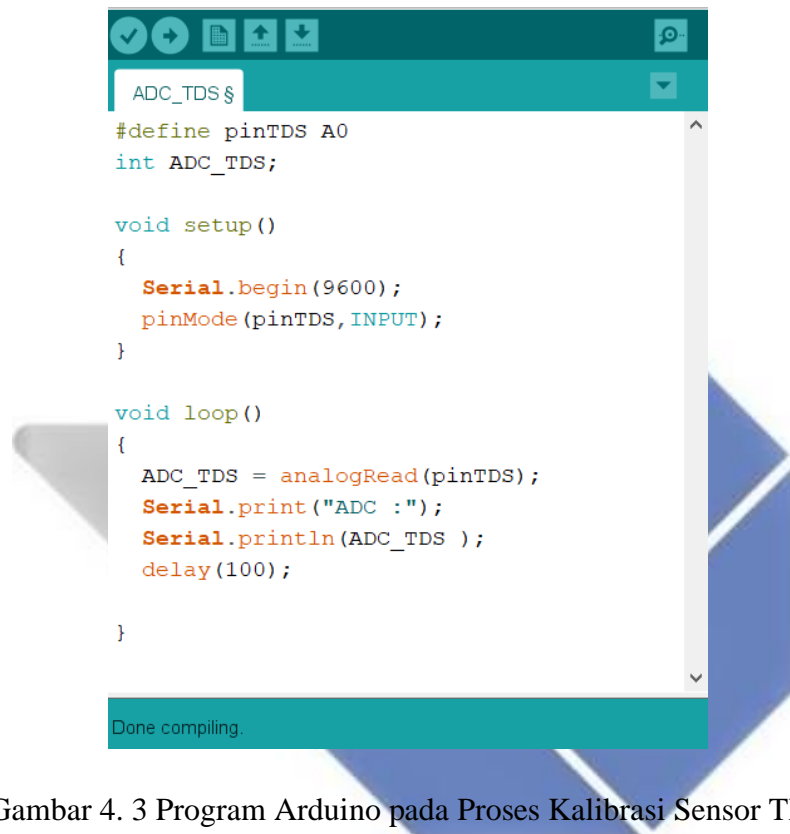
### 4.2.1 Kalibrasi Sensor TDS

Pada tahap ini dilakukan proses kalibrasi pada sensor TDS. Kalibrasi sensor dilakukan dengan membandingkan hasil pembacaan nilai PPM pada TDS meter dengan hasil pembacaan ADC pada Sensor TDS.



Gambar 4. 2 Rangkaian Skematik Kalibrasi Sensor TDS

Pin *output* analog dari sensor TDS dihubungkan dengan pin Analog *input* A0 pada Arduino. Pin VCC dan *ground* dari sensor dihubungkan juga ke pin VCC dan *ground* dari Arduino. Berikut ini merupakan program Arduino yang akan digunakan pada proses kalibrasi sensor TDS.



Gambar 4. 3 Program Arduino pada Proses Kalibrasi Sensor TDS

Pembacaan nilai ADC dari sensor TDS menggunakan perintah “analogRead” pada pin A0 arduino yang telah didefinisikan sebagai *input* analog dari sensor. Tipe data yang digunakan pada pembacaan analog adalah bilangan bulat (*integer*). Hasil pembacaan sensor akan ditampilkan dalam serial monitor. Hasil pengukuran nilai PPM pada TDS meter dan pembacaan nilai ADC dari sensor TDS akan ditampilkan pada tabel dibawah.

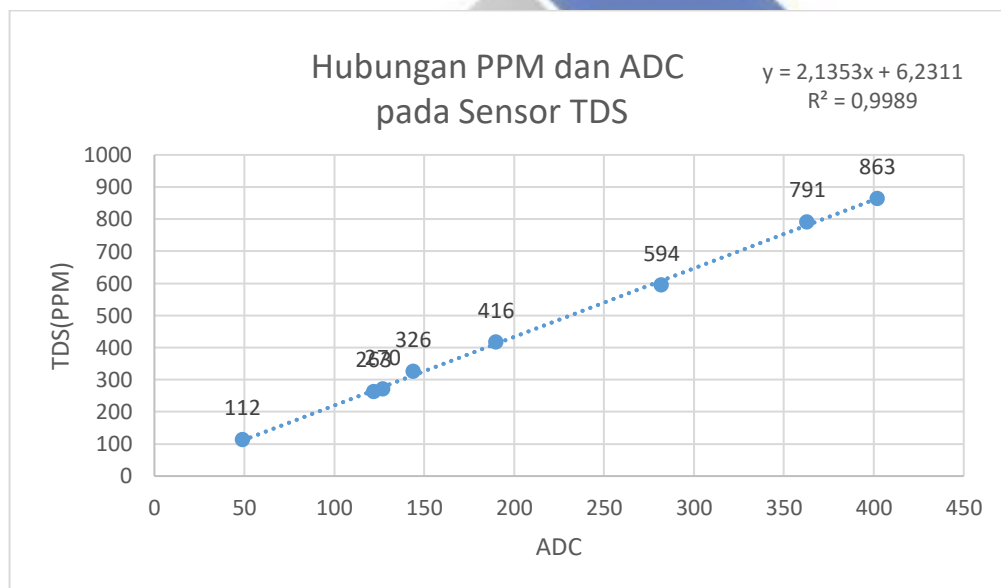
Tabel 4. 1 Perbandingan nilai ADC sensor TDS dan nilai PPM

No	Sensor TDS (ADC)	TDS Meter (PPM)
1	49	112



2	122	263
3	127	270
4	144	326
5	190	416
6	282	594
7	363	791
8	402	863

Berdasarkan data pada tabel 4.1, didapatkan grafik hubungan nilai PPM dari pembacaan TDS Meter dan nilai ADC dari pembacaan sensor TDS.



Gambar 4. 4 Grafik Hubungan nilai PPM dan ADC

Dari grafik persamaan pada gambar 4.4, dapat dilihat bahwa hasil pembacaan nilai ADC dari sensor TDS memiliki hubungan linieritas tinggi terhadap nilai PPM pada TDS meter. Hal ini ditunjukkan oleh nilai koefisien determinasi ( $R^2$ ) dengan nilai 99,89 %. Persamaan diatas akan digunakan sebagai persamaan untuk konversi nilai pembacaan sensor, yang semulanya berupa nilai tegangan ADC menjadi nilai PPM.

```
ADC_TDS
#define pinTDS A0
int ADC_TDS;
int PPM;
void setup()
{
  Serial.begin(9600);
  pinMode(pinTDS, INPUT);
}

void loop()
{
  ADC_TDS = analogRead(pinTDS);
  PPM = (2.1353*ADC_TDS)+6.2311;
  Serial.print("ADC :");
  Serial.println(ADC_TDS );
  delay(100);
}

Done compiling.
```

Gambar 4. 5 Program Arduino Untuk Konversi Nilai ADC Menjadi PPM Dengan Persamaan Regresi Linier

Persamaan regresi linier yang telah didapatkan dari grafik pada gambar 4.4 dimasukkan ke dalam program Arduino dengan nilai  $x$  adalah hasil pembacaan nilai analog sensor, yaitu *variable* “ADC\_TDS”. Program inilah yang akan digunakan pada proses pengujian untuk melihat tingkat akurasi pembacaan nilai PPM pada sensor TDS.

#### 4.2.2 Pengujian Sensor TDS

Proses pengujian ini dilakukan untuk melihat nilai keakuratan dari pembacaan sensor TDS. Hasil pembacaan sensor TDS yang telah dikonversi menjadi nilai PPM akan dibandingkan dengan pembacaan nilai PPM pada TDS Meter. Nilai selisih dari kedua hasil pembacaan akan dijadikan nilai eror untuk

menentukan seberapa akurat pembacaan sensor TDS. Berikut merupakan data hasil pengujian.

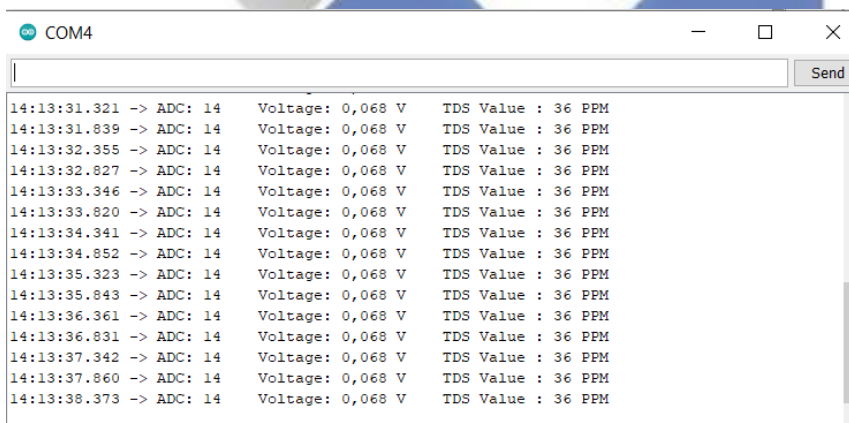
Tabel 4. 2 Data pengujian sensor TDS

No	Sensor TDS (PPM)	TDS Meter (PPM)	Akurasi (%)
1	16	18	88,89
2	36	38	94,74
3	110	112	98,21
4	277	271	97,79
5	285	287	99,30
6	330	328	99,39
7	422	424	99,53
8	800	790	99,73
9	894	940	95,11
10	997	1181	84,42
Rata-rata			95,61

Berdasarkan data hasil pengujian yang ditampilkan pada tabel 4.2, dapat disimpulkan bahwa sensor TDS ini mampu untuk mengukur nilai PPM dengan tingkat akurasi rata-rata 95,61%.



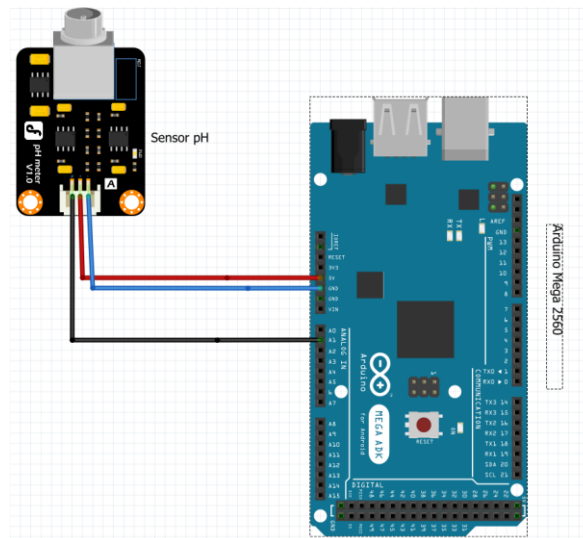
Gambar 4. 6 Pengujian Kadar PPM Menggunakan Sensor TDS dan TDS Meter



Gambar 4. 7 Pembacaan Sensor TDS pada Serial Monitor

### 4.2.3 Kalibrasi Sensor pH

Pada tahap ini dilakukan komparasi terhadap nilai pembacaan ADC dari sensor pH dan nilai pH pada alat ukur pH meter. Proses kalibrasi dilakukan menggunakan 3 jenis larutan *buffer*. Berikut merupakan rangkaian listrik dan program Arduino pada pengujian ini.



Gambar 4. 8 Rangkaian Skematik Pengujian Sensor pH

Pin *output* analog dari sensor pH dihubungkan dengan pin Analog *input* A1 pada Arduino. Pin VCC dan *ground* dari sensor dihubungkan juga ke pin VCC dan *ground* dari Arduino. Berikut ini merupakan program Arduino yang akan digunakan pada proses kalibrasi sensor pH.

```

#define pinpH A1
int ADC_pH;
void setup()
{
  Serial.begin(9600);
  pinMode(pinpH, INPUT);
}

void loop()
{
  ADC_pH = analogRead(pinpH);
  Serial.print("ADC :");
  Serial.println(ADC_pH );
  delay(100);}

```

Gambar 4. 9 Program Arduino Kalibrasi Sensor pH

Pembacaan nilai ADC dari sensor pH menggunakan perintah “analogRead” pada pin A1 arduino yang telah didefinisikan sebagai *input* analog

dari sensor. Tipe data yang digunakan pada pembacaan analog adalah bilangan bulat (*integer*). Hasil pembacaan sensor akan ditampilkan dalam serial monitor dengan memanggil fungsi “Serial.begin”.



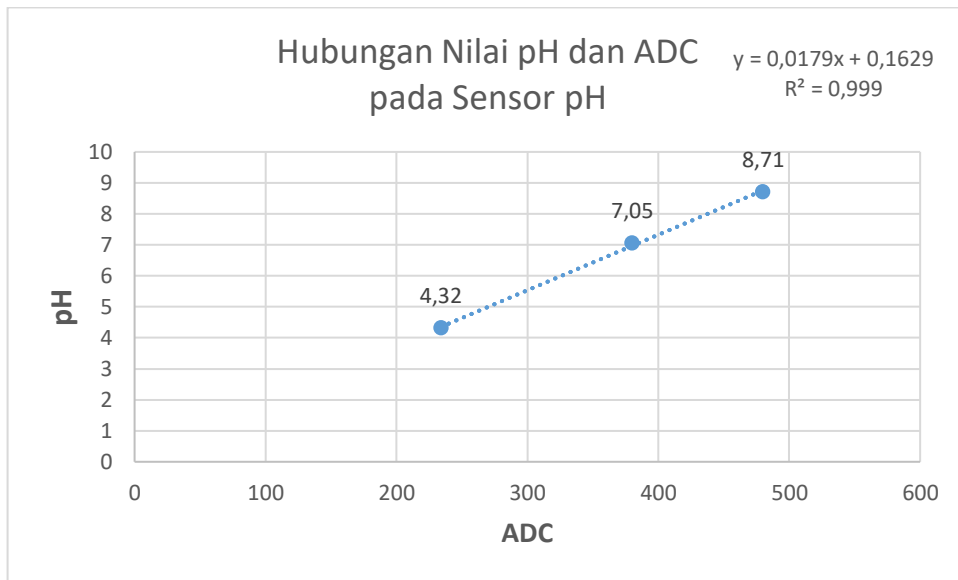
Gambar 4. 10 Sampel Air Kalibrasi Sensor pH

Hasil pengukuran nilai ADC dan tegangan dari sensor pH dan pembacaan nilai pH dari alat ukur pH meter ditampilkan pada tabel dibawah.

Tabel 4. 3 Perbandingan nilai ADC sensor pH dan nilai pH

No	Sensor pH (ADC)	pH Meter (pH)
1	234	4,32
2	380	7,05
3	480	8,71

Berdasarkan data pada tabel 4.3, didapatkan grafik hubungan antara nilai pembacaan pH pada pH meter terhadap nilai tegangan dan ADC dari pembacaan sensor pH.



Gambar 4. 11 Grafik Hubungan nilai PPM dan ADC

Dari grafik persamaan 4.11 diatas, dapat kita lihat bahwa hasil pembacaan sensor pH dan pH meter memiliki linieritas yang tinggi. Hal ini ditunjukkan oleh nilai koefisien determinasi ( $R^2$ ) dengan nilai 99,9%. Nilai ini menunjukkan bahwa sensor pH yang digunakan pada alat ini dapat berfungsi dengan baik, karena memiliki tingkat linieritas tinggi. Persamaan regresi linier yang didapat dari grafik diatas, akan digunakan sebagai persamaan untuk konversi nilai pembacaan sensor yang berupa tegangan ADC menjadi nilai pH. Persamaan yang akan digunakan pada program Arduino adalah sebagai berikut:

```

#define pinpH A1
int ADC_pH;
float pH_value, tegangan;
void setup()
{
  Serial.begin(9600);
  pinMode(pinpH, INPUT);
}

void loop()
{
  ADC_pH = analogRead(pinpH);
  tegangan = (ADC_pH * 5.0)/1024; // konversi ADC ke tegangan
  pH_value = (0.0179*ADC_pH)+0.1629; // Regresi Linier
  Serial.print("ADC :");
  Serial.print(ADC_pH );
  Serial.print("voltage :");
  Serial.print(tegangan);
  Serial.print("pH :");
  Serial.println(pH_value);
  delay(100);
}

```

Gambar 4. 12 Program Arduino Pembacaan Nilai pH dengan Regresi Linier

Gambar 4.12 merupakan program arduino untuk pembacaan sensor pH dengan menggunakan persamaan regresi linier untuk merubah nilai ADC sensor pH menjadi nilai pH. Jenis data yang digunakan untuk pembacaan nilai pH adalah *floating point*. Hal ini dilakukan untuk menampilkan data pembacaan sensor pH yang memiliki bentuk pecahan.

#### 4.2.4 Pengujian Sensor pH

Tahap pengujian ini dilakukan untuk melihat kakuratan hasil pembacaan dari sensor pH. Hasil pembacaan sensor pH yang telah dikonversi menjadi nilai pH akan dibandingkan dengan pembacaan pada pH Meter. Nilai selisih dari kedua hasil pembacaan akan dijadikan nilai eror untuk menentukan seberapa akurat pembacaan sensor pH. Berikut merupakan data hasil pengujian.

Tabel 4. 4 Data pengujian sensor pH

Jenis Larutan	Sensor pH (pH)	pH Meter (pH)	Akurasi (%)
Air Jeruk Nipis	3,06	3,04	99,34
Larutan buffer 1	4,29	4,34	98,85
Larutan buffer 2	6,76	6,93	97,55
Larutan buffer 3	9,02	8,99	99,67
Air Sabun	9,01	9,02	99,89
Rata-rata			99,06

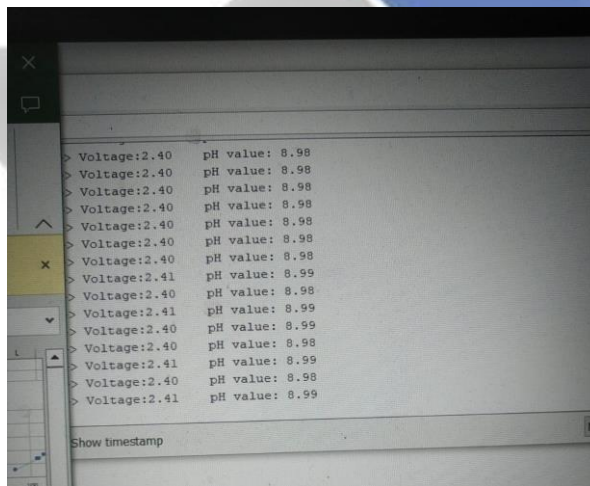
Berdasarkan data hasil pengujian yang ditampilkan pada tabel 4.4, dapat disimpulkan bahwa sensor pH ini mampu untuk mengukur nilai pH dengan tingkat akurasi rata-rata 99,06 %.





Gambar 4. 13 Sampel Air Pengujian Sensor pH

Gambar diatas merupakan sampel air yang digunakan dalam proses pengujian sensor pH. Terdapat lima sampel air yang terdiri dari larutan air jeruk, larutan air sabun, dan tiga jenis larutan *buffer*.



Gambar 4. 14 Tampilan Serial Monitor Pengujian Sensor pH

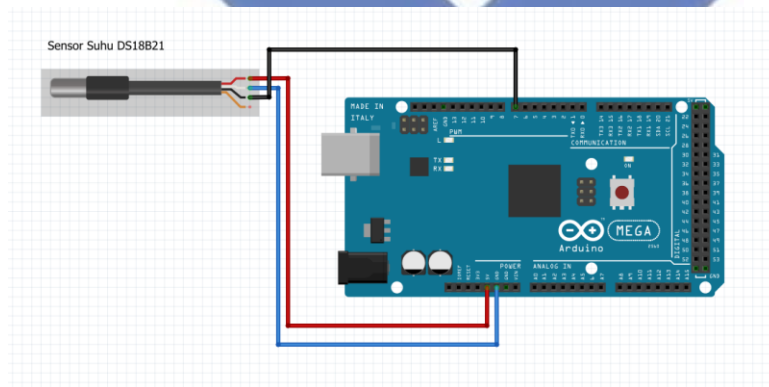
Hasil pembacaan sensor pH ditampilkan pada serial monitor Arduino seperti pada Gambar 4.14. Nilai yang ditampilkan merupakan nilai Tegangan ADC dan nilai konversi pH.



Gambar 4. 15 Pengujian Sensor pH dan pH Meter

#### 4.2.5 Kalibrasi Sensor Suhu DS18B20

Pada proses kalibrasi sensor suhu DS18B20 ini dilakukan, dikarenakan *output* dari sensor DS18B20 merupakan keluaran digital, penulis menggunakan program Arduino yang berasal *library OneWire* yang dikembangkan oleh DFRobot. Berikut ini merupakan gambar rangkaian listrik dan program Arduino dari proses kalibrasi sensor suhu DS18B20.



Gambar 4. 16 Rangkaian Skematik Pengujian Sensor Suhu DS18B20

Hasil pembacaan sensor suhu dan termometer akan dibuatkan grafik hubungan antara kedua nilai pembacaan tersebut dengan menggunakan metode regresi linier. Berikut ini merupakan program arduino untuk pembacaan sensor suhu DS18B20.

```

Sensor_suhu_dfrobot$
#include <OneWire.h>

int DS18S20_Pin = 7; |
//Temperature chip i/o
OneWire ds(DS18S20_Pin);

void setup(void) {
  Serial.begin(115200);
}

void loop(void) {
  float temperature = getTemp();
  Serial.println(temperature);
  delay(100);
}

```

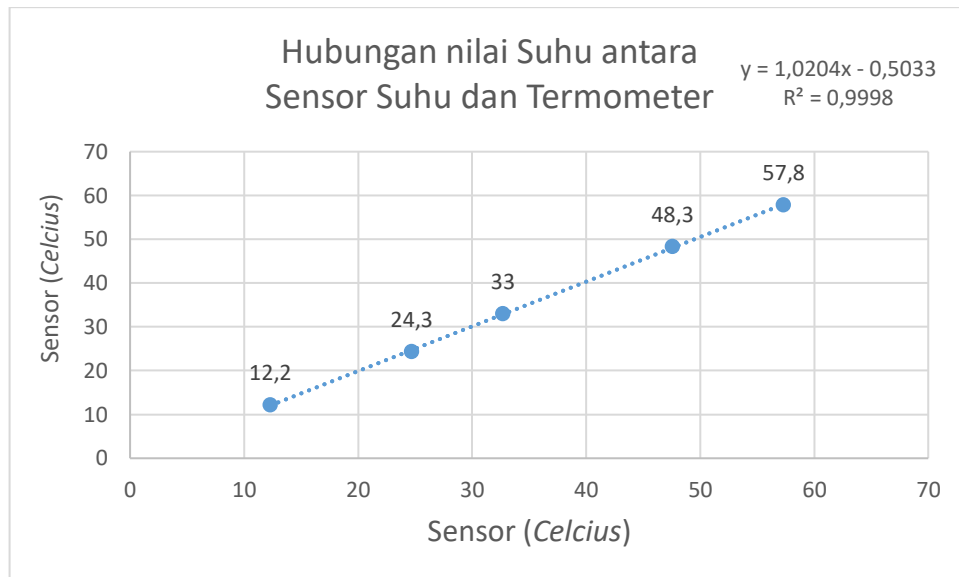
Gambar 4. 17 Program Arduino Pembacaan Sensor Suhu DS18B20

Proram pada gambar 4.17 diatas merupakan program untuk pembacaan suhu pada sensor DS18B20. Program pembacaan sensor suhu menggunakan *library* dari “OneWire” dengan input pembacaan pada pin digital arduino. Pada proses kalibrasi ini pin *input* pembacaan sensor suhu didefinikan pada pin 7 arduino. Data hasil pembacaan sensor DS18B20 dan termometer akan ditampilkan pada tabel berikut.

Tabel 4. 5 Data kalibrasi sensor Suhu

No	Sensor Suhu DS18B20 (C <sup>o</sup> )	Termometer (C <sup>o</sup> )
1	12,31	12,2
2	24,69	24,3
3	32,69	33
4	47,56	48,3
5	57,31	57,8

Dari data pada tabel 4.5, didapatkan grafik hubungan antara nilai pembacaan sensor suhu DS18B20 dan termometer.



Gambar 4. 18 Grafik Hubungan nilai Suhu antara sensor Suhu dan Termometer

Berdasarkan grafik persamaan diatas, dapat dilihat bahwa kedua nilai pembacaan suhu memiliki hubungan linieritas tinggi, yang ditunjukkan oleh nilai koefisien determinasi ( $R^2$ ) dengan nilai 99,98%. Rumus Persamaan yang didapatkan dari grafik hubungan diatas akan digunakan untuk merubah nilai pembacaan awal sensor suhu agar mendekati nilai pembacaan alat ukur. Berikut ini merupakan persamaan yang akan digunakan pada program Arduino sensor DS18B20.

```

Sensor_suhu_dfrobot $
#include <OneWire.h>

int DS18S20_Pin = 7;
int suhu;
//Temperature chip i/o
OneWire ds(DS18S20_Pin);

void setup(void) {
  Serial.begin(115200);
}

void loop(void) {
  float temperature = getTemp();
  suhu = 1.0204*temperature - 0.5033;//.....Regresi linier DS18S20
  Serial.println(suhu);

  delay(100);
}

```

Gambar 4. 19 Program Arduino Regresi Linier Sensor Suhu

Program pada gambar 4.19 diatas merupakan program arduino yang akan digunakan untuk proses pengujian sensor suhu DS18B20 yang telah menggunakan rumus persamaan regresi linier.

#### 4.2.6 Pengujian Sensor Suhu DS18B20

Proses pengujian ini dilakukan untuk membandingkan hasil pembacaan dari sensor DS18B20 yang telah di kalibrasi sebelumnya, dengan pembacaan suhu pada termometer. Nilai selisih dari kedua hasil pembacaan tersebut akan dijadikan nilai eror untuk menentukan seberapa presisi pembacaan sensor suhu. Hasil pengujian akan ditampilkan pada tabel dibawah.

Tabel 4. 6 Data pengujian sensor Suhu

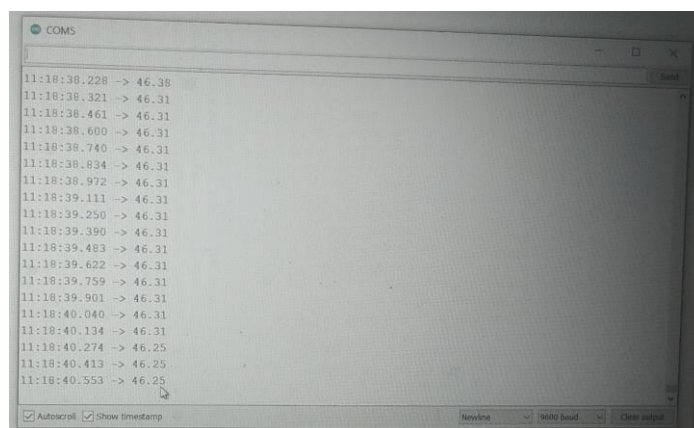
No	Sensor Suhu ( $C^{\circ}$ )	Termometer ( $C^{\circ}$ )	Akurasi (%)
1	7,2	7,1	98,59
2	13,59	13,6	99,93
3	24,68	24,3	98,40

4	27,28	27,3	99,93
5	32,69	33	99,06
6	46,25	46,9	99,61
7	53,4	53,6	99,63
8	57,31	57,8	99,15
Rata-rata			99,27

Berdasarkan hasil pengujian Sensor DS18B20 yang telah disajikan diatas, dapat disimpulkan bahwa sensor Suhu DS18B20 ini mampu mengukur suhu air dengan tingkat akurasi rata-rata di angka 99,27%.



Gambar 4. 20 Proses Pengujian Suhu dengan Sensor Suhu DS18B20 dan Termometer



Gambar 4. 21 Hasil Pembacaan Sensor Suhu DS18B20 pada Serial Monitor

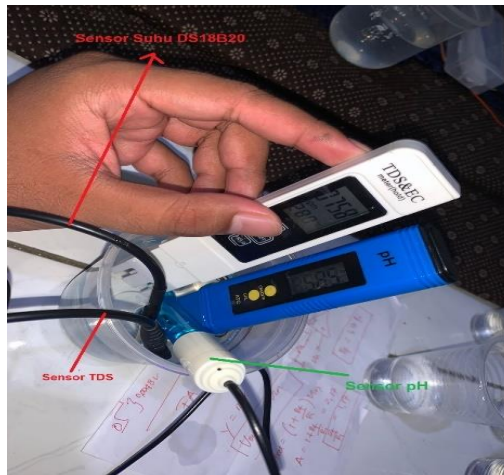
#### 4.2.7 Pengujian Sensor TDS, Sensor Suhu, dan Sensor pH Secara Bersamaan

Pengujian ini dilakukan untuk melihat bagaimana kinerja dari ketiga sensor apabila bekerja pada saat bersamaan dalam satu wadah yang sama. Pengujian dilakukan dengan melihat tingkat akurasi dari pembacaan ketiga sensor pada saat bekerja bersamaan. Pengujian dilakukan menggunakan 5 sampel air dan ditempatkan pada wadah yang berisikan air panas untuk mendapatkan pembacaan pada suhu tinggi, serta ditempatkan pada wadah untuk mendapatkan pengujian pada suhu rendah. Berikut ini data tabel hasil pengujian.

Tabel 4. 7 Data pengujian sensor TDS, sensor Suhu, dan sensor pH secara bersamaan

No	Suhu ( $C^{\circ}$ )			TDS (PPM)			pH			
	Sensor Suhu	Termometer	Akurasi (%)	Sensor TDS	TDS Meter	Akurasi (%)	Sensor pH	pH Meter	Akurasi (%)	
1	8,72	8,9	97,94	124	131	94,35	3,34	2,92	12,57	
2	14,36	14,5	99,03	235	243	96,60	4,21	3,55	15,68	
3	27,53	27,6	99,75	376	388	96,81	6,98	6,46	7,45	
4	44,76	45,1	99,24	563	581	96,80	7,13	6,51	8,70	
5	57,42	57,8	99,34	859	910	94,06	9,12	8,48	7,02	
Rata-rata			99,06				95,72			

Dari hasil pengujian, didapati bahwa terjadi penurunan akurasi pada pembacaan sensor pH yang cukup signifikan. Dimana pada kalibrasi awal sensor pH memiliki akurasi 99,06 %, dan turun menjadi 89,72%.

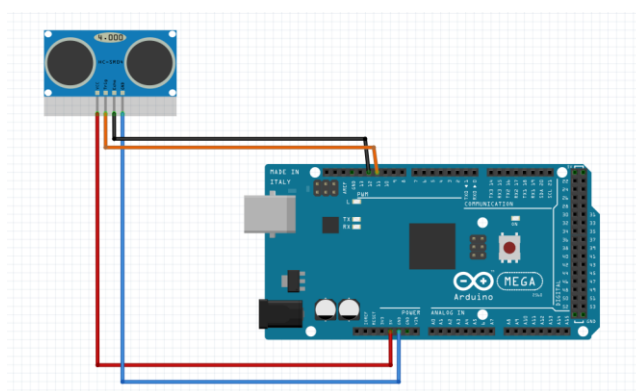


Gambar 4. 22 Pengujian Sensor TDS, pH, dan Suhu bersamaan

Gambar diatas merupakan proses pengujian ketiga sensor dalam satu wadah pada waktu bersamaan. Dari data tabel 4.7 didapati kesimpulan bahwa akurasi pembacaan sensor pH mengalami penurunan pada saat ketiga sensor bekerja bersamaan.

#### 4.2.8 Kalibrasi Sensor Ultrasonik

Proses kalibrasi sensor ultrasonik ini bertujuan untuk mendapatkan rumus konversi terhadap nilai pembacaan sensor, yang akan diubah menjadi nilai jarak dalam satuan sentimeter (cm). Proses kalibrasi dilakukan dengan membandingkan hasil pembacaan sensor ultrasonik (*echotime*) dan jarak pada alat ukur meteran terhadap benda. Berikut merupakan gambar rangkaian sensor ultrasonik dan data hasil pembacaan jarak dari sensor ultrasonik dan meteran.



Gambar 4. 23 Rangkaian Skematik Pengujian Sensor Ultrasonik



Gambar 4.23 diatas merupakan rangkaian skematik yang digunakan pada proses kalibrasi sensor ultrasonic ini. Pin VCC dan GND dari sensor dihubungkan ke pin VCC +5V dan GND dari arduino. Kemudian pin *trigger* dan *echo* sensor dihubungkan ke input digital pin 11 dan 12 arduino.

```

ultrasonik_raw $
int trig_pin = 11;
int echo_pin = 12;
long echotime;
void setup() {
  Serial.begin(9600);
  pinMode(trig_Pin, OUTPUT);
  pinMode(echo_Pin, INPUT);
}

void loop() {
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);
  echotime = pulseIn(echo_pin, HIGH);
  Serial.print("echo :");
  Serial.println(echotime);
}
Done Saving.

```

Gambar 4. 24 Program Arduino Pembacaan Sensor Ultrasonik

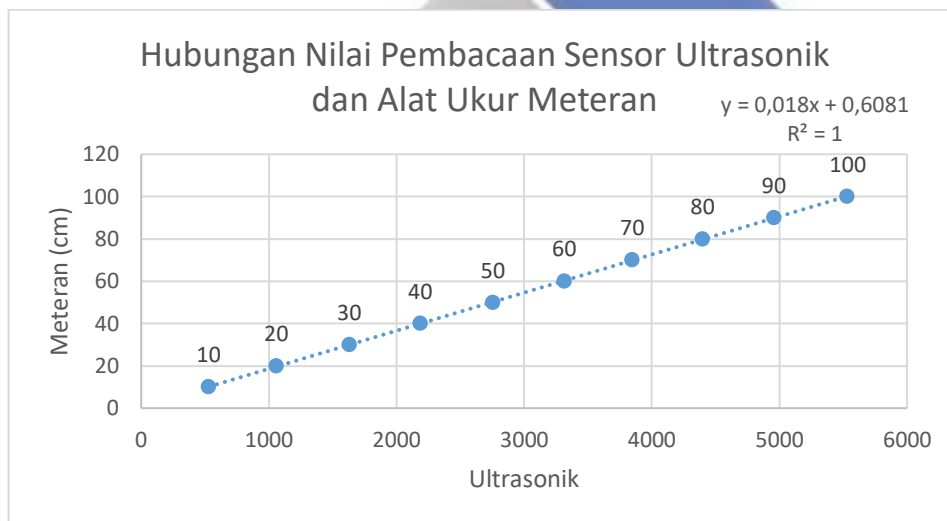
Program diatas merupakan program arduino untuk pembacaan sensor ultrasonic. Hasil dari pembacaan program pada sensor ultrasonic diatas masih berupa data waktu pengiriman dan penerimaan dari sensor ultrasonic. Berikut ini merupakan data hasil kalibrasi sensor ultrasonic.

Tabel 4. 8 Data kalibrasi sensor Ultrasonik

No	Sensor Ultrasonik	Meteran (cm)
1	530,4	10
2	1061	20
3	1631,6	30

4	2189	40
5	2757,2	50
6	3315,4	60
7	3848,6	70
8	4398,6	80
9	4958,6	90
10	5533,2	100

Dari data pada tabel diatas, didapatkan grafik regresi linier hubungan antara nilai pembacaan sensor ultrasonik dan nilai pada meteran.



Gambar 4. 25 Grafik Hubungan nilai jarak antara sensor Ultrasonik dan Meteran

Berdasarkan grafik persamaan diatas, dapat dilihat bahwa hasil pembacaan dari kedua alat memiliki hubungan linieritas tinggi, yang ditunjukkan oleh nilai  $R^2$  dengan nilai 100%. Rumus persamaan yang didapatkan dari grafik hubungan diatas akan digunakan untuk konversi nilai pembacaan awal sensor ultrasonik menjadi satuan jarak (cm). Berikut ini merupakan program arduino yang telah menggunakan rumus persamaan regresi linier.

```

ultrasonik_raw
int trig_pin = 11;
int echo_pin = 12;
long echotime;
float distance;
void setup() {
  Serial.begin(9600);
  pinMode(trig_Pin, OUTPUT);
  pinMode(echo_Pin, INPUT);
}

void loop() {
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);
  echotime = pulseIn(echo_pin, HIGH);
  distance = (0.018*float(echotime)) + 0.6081; //regresi linier
  Serial.print("jarak :");
  Serial.println(distance);
}

```

Done Saving.

Gambar 4. 26 Program Arduino Sensor Ultrasonik Regresi Linier

Program arduino diatas merupakan program untuk mengubah nilai pembacaan sensor ultrasonic yang awalnya berupa data waktu menjadi satuan jarak dengan menerapkan rumus regresi linier yang telah didapatkan sebelumnya.

#### 4.2.9 Pengujian Sensor Ultrasonik

Proses pengujian ini dilakukan untuk membandingkan hasil pembacaan dari sensor ultrasonik yang telah di kalibrasi sebelumnya, dengan pembacaan alat ukur meteran. Nilai selisih dari kedua hasil pembacaan tersebut akan dijadikan nilai eror untuk menentukan seberapa presisi pembacaan sensor ultrasonik. Hasil pengujian akan ditampilkan pada tabel dibawah.

Tabel 4. 9 Data Pengujian sensor Ultrasonik

No	Sensor Ultrasonik (cm)	Alat Ukur Meteran (cm)	Akurasi (%)
1	10,12	10	98,81
2	20,05	20	99,75
3	31	30	96,77

4	39,87	40	99,67
5	50,14	50	99,72
6	59,98	60	99,97
7	70,05	70	99,93
8	80,11	80	99,86
9	89,68	90	99,64
10	98,97	100	98,96
Rata-rata			99,31

Berdasarkan hasil pengujian sensor ultrasonik yang telah disajikan diatas, dapat disimpulkan bahwa sensor ultrasonik ini mampu mengukur jarak dengan tingkat akurasi rata-rata di angka 99,31 %.



Gambar 4. 27 Pengujian Pembacaan Jarak dengan Sensor Ultrasonik dan Meteran

#### 4.2.10 Kalibrasi Sensor Ultrasonik Terhadap Pembacaan Volume Air

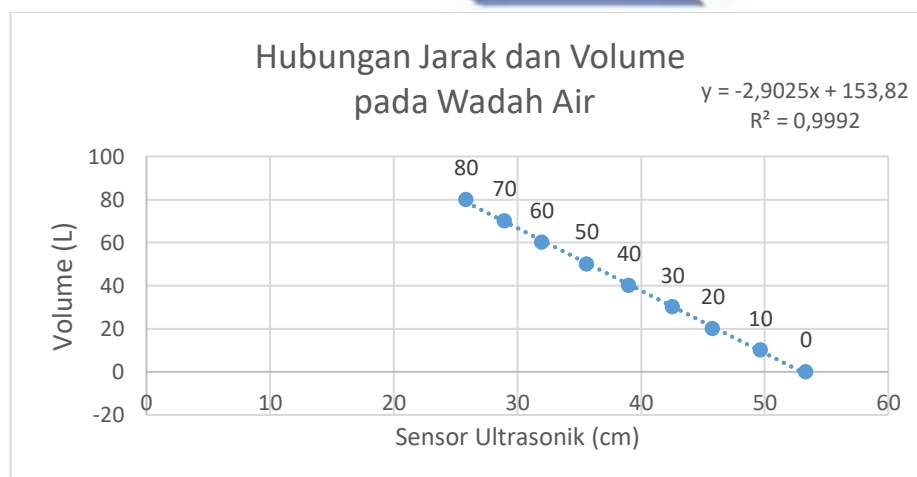
Pada tahap ini proses kalibrasi dilakukan dengan membandingkan nilai volume air yang ada pada wadah dengan pembacaan sensor ultrasonik. Pengujian dilakukan dengan cara menambahkan air pada wadah secara bertahap, dan

membandingkannya dengan hasil pembacaan sensor ultrasonik. Berikut merupakan data hasil kalibrasi.

Tabel 4. 10 Tabel Kalibrasi Sensor Ultrasonik untuk Pembacaan Volume Air

No	Sensor Ultrasonik (cm)	Volume air (L)
1	53,34	0
2	49,68	10
3	45,82	20
4	42,57	30
5	39,04	40
6	35,63	50
7	32,00	60
8	28,97	70
9	25,88	80

Dari data diatas, didapatkan grafik persamaan regresi linier hubungan pembacaan jarak pada sensor ultrasonic dan volume air pada wadah. Berikut merupakan grafik persamaan regresi linier.



Gambar 4. 28 Grafik Hubungan Jarak dan Volume air

Berdasarkan grafik persamaan diatas, dapat dilihat bahwa pembacaan nilai jarak pada sensor ultrasonik memiliki hubungan yang linier terhadap volume air yang ada didalam wadah. Hal ini ditunjukkan oleh nilai  $R^2$  dengan nilai 99,92%. Persamaan regresi linier diatas akan digunakan untuk konversi nilai pembacaan sensor ultrasonik menjadi satuan volume air pada program arduino. Berikut merupakan program Arduino yang digunakan untuk membaca volume air yang ada pada bak menggunakan sensor ultrasonic.

```
int trig_pin = 11;
int echo_pin = 12;
long echotime;
float distance, volume_air;
void setup() {
  Serial.begin(9600);
  pinMode(trig_Pin, OUTPUT);
  pinMode(echo_Pin, INPUT);
}

void loop() {
  digitalWrite(trig_pin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig_pin, LOW);
  echotime = pulseIn(echo_pin, HIGH);
  distance = (0.018*float(echotime)) + 0.6081; //regresi jarak
  volume_air = (-2.9025*distance) + 153,82; //regresi volume
  Serial.print("volume :");
  Serial.println(volume_air);
}

Done Saving.
```

Gambar 4. 29 Program Arduino Konversi Nilai Jarak Menjadi Volume Air

Program diatas merupakan bentuk penerapan dari persamaan regresi yang telah didapat pada grafik gambar 4.28. Nilai dari variable x dari persamaan regresi linier diubah menjadi variable “distance” dari pembacaan jarak pada sensor ultrasonik.

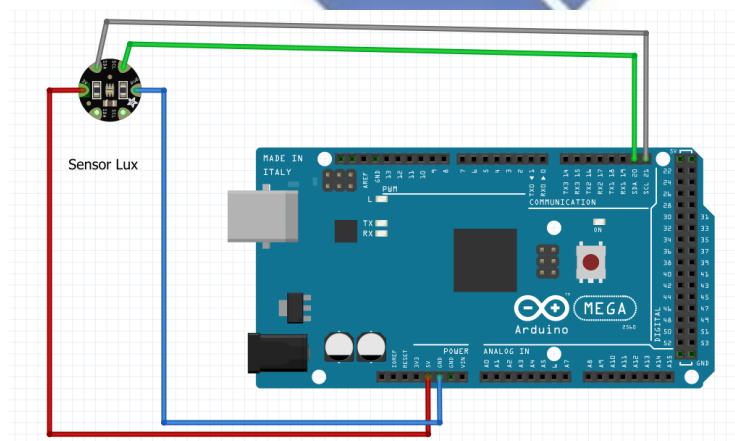


Gambar 4. 30 Kalibrasi Sensor Ultrasonik terhadap Pembacaan Volume Air

Pengujian dilakukan dengan menambahkan air secara bertahap ke dalam bak hidroponik. Proses ini dilakukan bertahap dengan penambahan air setiap 10 liter menggunakan gelas ukur. Hasil pembacaan jarak pada sensor ultrasonic akan dibandingkan dengan volume air yang ada pada bak hidroponik.

#### 4.2.11 Kalibrasi Sensor Lux

Pada proses kalibrasi sensor lux bertujuan untuk mendapatkan nilai konversi yang mendekati pembacaan alat ukur lux meter. Berikut ini merupakan gambar rangkaian skematik pada proses kalibrasi sensor lux.



Gambar 4. 31 Rangkaian Skematik Pengujian Sensor Lux

Sensor lux menggunakan jenis komunikasi I2C pada proses pembacaan nilai intensitas cahaya. Pin SCL dan SDA pada sensor dihubungkan ke pin SCL dan SDA pada Arduino. Kemudian pin VCC dan GND pada sensor dihubungkan pada pin VCC dan GND Arduino. Berikut ini merupakan program arduino untuk pembacaan nilai sensor lux.

```

#include <Wire.h>
#include <BH1750.h>
BH1750 lightMeter;
float lux;
void setup(){
  Serial.begin(9600);
  // Initialize the I2C bus (BH1750 library doesn't do this automatically)
  Wire.begin();
  // On esp8266 you can select SCL and SDA pins using Wire.begin(D4, D3);
  // For Wemos / Lolin D1 Mini Pro and the Ambient Light shield use Wire.begin(D02, D01);
  lightMeter.begin();
  Serial.println(F("BH1750 Test begin"));
}
void loop() {
  lux = lightMeter.readLightLevel();
  Serial.print("Light: ");
  Serial.print(lux);
  Serial.println(" lux");
  delay(500);
}

```

Gambar 4. 32 Program Arduino Pembacaan Sensor Lux

Pemrograman sensor lux BH1750 FVI ini menggunakan bantuan library “Wire.h.” dan “BH1750.h” untuk mengolah data dari komunikasi I2C sensor lux. Namun hasil pembacaan dari sensor lux ini belum akurat, sehingga dilakukan proses kalibrasi dengan membandingkan hasil pembacaan pada lux meter. Data hasil pembacaan sensor lux dan alat ukur lux meter akan ditampilkan pada tabel berikut.

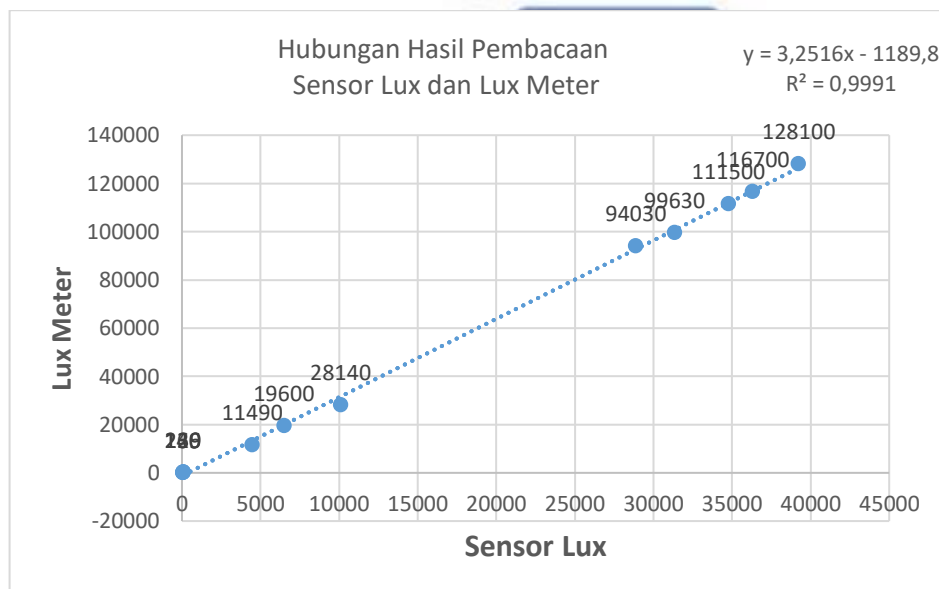
Tabel 4. 11 Data kalibrasi sensor Lux

No	Sensor lux	Lux Meter
1	55	146
2	61,67	160
3	86,67	229
4	92,5	239
5	4485	11490



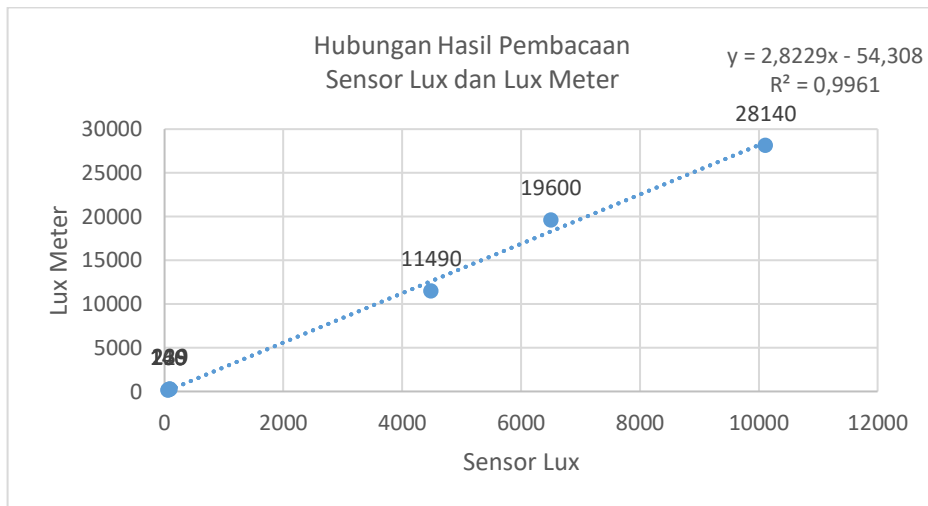
6	6500	19600
7	10110	28140
8	28894	94030
9	34773	111500
10	36321	116700
11	39238	128100

Berdasarkan data pada tabel 4.11 diatas, didapatkan grafik hubungan antara nilai pembacaan sensor lux dan lux meter sebagai berikut..



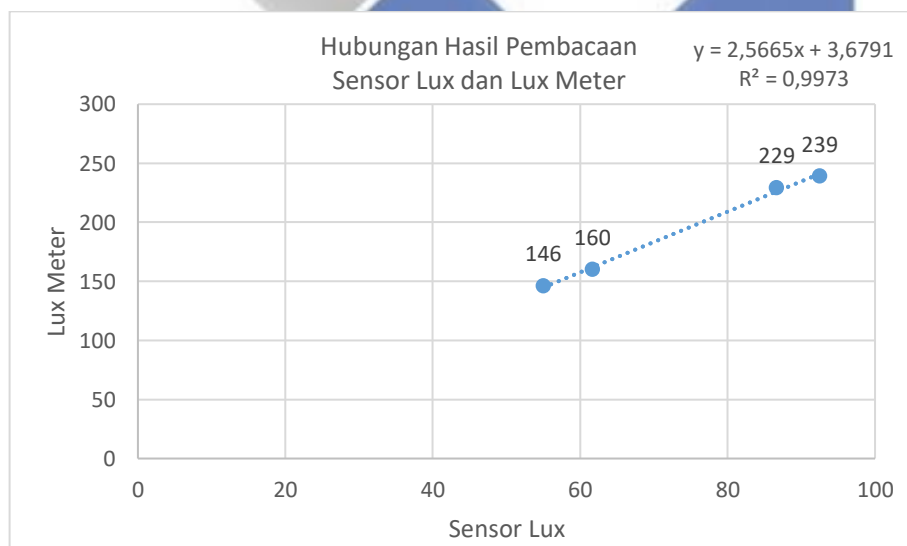
Gambar 4. 33 Grafik Hubungan Nilai Lux Pembacaan Sensor dan Lux Meter (1)

Grafik persamaan diatas merupakan grafik hubungan pembacaan sensor lux dan lux meter yang mewakili keseluruhan data dari tabel 4.11. Persamaan dari grafik pada gambar 4.33 akan digunakan untuk konversi nilai pembacaan sensor lux dengan nilai lebih dari 30.000 lux.



Gambar 4. 34 Grafik Hubungan Nilai Lux Pembacaan Sensor dan Lux Meter (2)

Grafik persamaan diatas merupakan grafik hubungan pembacaan sensor lux dan lux meter yang mewakili tujuh data pertama dari tabel 4.11. Persamaan grafik pada gambar 4.34 akan digunakan pada konversi nilai lux dengan kisaran 1.000-30.000 lux.



Gambar 4. 35 Grafik Hubungan Nilai Lux Pembacaan Sensor dan Lux Meter (3)

Grafik persamaan diatas merupakan grafik hubungan pembacaan sensor lux dan lux meter yang mewakili empat data pertama dari tabel 4.11. Persamaan grafik

pada gambar 4.35 akan digunakan untuk konversi nilai lux dengan kisaran 0-1.000 lux

Nilai persamaan yang didapatkan dari grafik hubungan diatas akan digunakan untuk merubah nilai pembacaan awal sensor Lux agar mendekati nilai pembacaan alat ukur. Pembagian rumus konversi sensor lux menjadi tiga ini dilakukan karena persamaan regresi linier dengan sampel keseluruhan data (gambar 4.31), memiliki nilai konstanta yang besar, yaitu sebesar -1189,8, sehingga dapat menyebabkan pembacaan nilai lux pada *range* yang lebih kecil akan memiliki nilai akurasi yang sangat kecil. Berikut ini merupakan program arduino dengan rumus regresi linier diatas.

```
//SENSOR CAHAYA
lux = lightMeter.readLightLevel();
if (lux <= 1000){
  hasillux = (2.5665*lux) - 3.6791; //kurang dari 1000
}
else if (lux > 1000 && lux < 30000){
  hasillux = (2.8229*lux) - 54.308; //pembacaan 1000-30000
}
else if (lux >= 30000){
  hasillux = (3.2516*lux) - 1189.8; //pembacaan lebih dari 30000
}
```

Gambar 4. 36 Program Kalibrasi Sensor Lux dengan Rumus Regresi

#### 4.2.12 Pengujian Sensor Lux

Tahap ini untuk mendapatkan nilai akurasi dengan cara membandingkan hasil pembacaan nilai lux dari sensor lux yang telah dikonversi sebelumnya, dengan hasil pembacaan dari lux meter. Nilai selisih dari kedua hasil pembacaan akan dijadikan nilai *error* untuk menentukan tingkat akurasi pembacaan sensor. Data hasil pengujian akan ditampilkan pada tabel dibawah.

Tabel 4. 12 Tabel Pengujian Sensor Lux

No	Sensor Lux	Lux Meter	Akurasi (%)
1	476,25	633	67,09
2	4753,21	5211	90,37
3	23476,52	21280	90,64
4	35324,07	31540	89,29
5	68310,31	61310	89,75
6	52371,30	47890	91,44
7	26357,23	29350	88,65
8	12365,17	13560	90,34
9	31346,45	34210	90,86
10	32147,73	29140	90,65
Rata-rata			87,91

Berdasarkan data pengujian diatas, nilai rata-rata akurasi dari pembacaan sensor lux adalah 87,91 %. Hal ini menunjukkan bahwa sensor dapat mengukur nilai intensitas cahaya dengan baik.

### 4.3 Pengujian Pompa Air

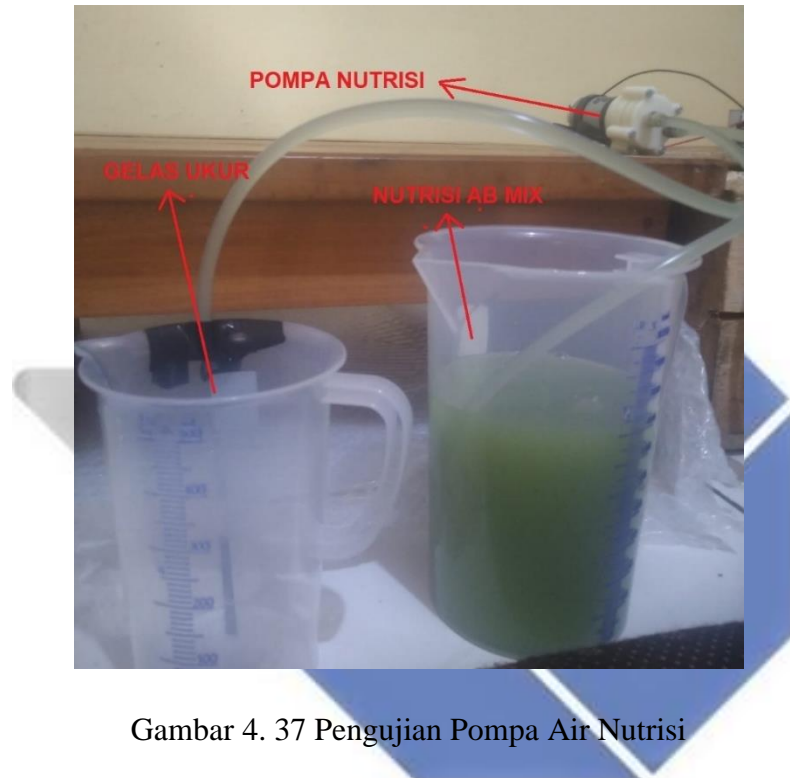
Pada tahap ini dilakukan pengujian yang ditujukan untuk mendapatkan nilai Debit (Q) dari pompa air. Nilai debit ini akan digunakan pada proses pembuatan formula program sistem pengisian air dan nutrisi oromatis. Metode yang digunakan pada pengujian debit popa air ini adalah dengan melakukan pengukuran secara manual menggunakan gelas ukur dan stopwatch. Adapun rumus yang digunakan adalah rumus debit air, yaitu:

$$Q = \frac{V}{T}$$

Q = Debit (L/s)  
V = Volume (L)  
T = Waktu (s)

### 4.3.1 Pengujian Pompa Air Nutrisi

Pengujian ini dilakukan untuk mendapatkan nilai debit air yang nantinya akan digunakan dalam perhitungan waktu aktif pompa nutrisi pada sistem pengisian air dan nutrisi otomatis. Hasil pembagian dari volume air dengan waktu nyala pompa merupakan nilai debit dari pompa nutrisi.



Gambar 4. 37 Pengujian Pompa Air Nutrisi

Proses pengujian dilakukan dengan cara mengaktifkan pompa air nutrisi dan memompa nutrisi ke dalam gelas ukur hingga mencapai batas 500 ml dan 300 ml. Waktu yang dibutuhkan oleh pompa untuk mencapai batas tersebut akan diambil dan digunakan dalam perhitungan nilai debit pompa. Berikut ini merupakan data hasil pengujian Pompa nutrisi.

Tabel 4. 13 Data pengujian Pompa Air Nutrisi

No	Pompa DC 1					
	Volume Air (ml)	Waktu (detik)	Debit (ml/detik)	Volume Air (ml)	Waktu (detik)	Debit (ml/detik)
1	500	18,12	27,594	500	17,96	27,840

2	500	17,95	27,855	500	17,92	27,902
3	500	18,14	27,563	500	17,82	28,058
4	500	17,98	27,809	500	17,93	27,886
5	500	18,21	27,457	500	17,98	27,809
6	500	18,1	27,624	500	17,86	27,996
7	500	18,15	27,548	500	17,97	27,824
8	500	18,19	27,488	500	18,02	27,747
9	500	18,18	27,503	500	17,92	27,902
10	500	17,94	27,871	500	18,05	27,701
11	300	10,98	27,322	300	11,11	27,003
12	300	10,97	27,347	300	11,09	27,051
13	300	10,94	27,422	300	11,12	26,978
14	300	10,88	27,574	300	11,14	26,930
15	300	10,92	27,473	300	11,15	26,906
16	300	10,97	27,357	300	11,09	27,051
17	300	10,89	27,548	300	11,12	26,978
18	300	10,87	27,599	300	11,18	26,834
19	300	10,91	27,498	300	11,08	27,076
20	300	10,86	27,624	300	11,12	26,978
	Rata-rata		27,553			27,422

Nilai debit pompa air yang telah didapatkan dari hasil pengujian ini yang kemudian akan digunakan dalam program pengisian nutrisi otomatis, nilai debit ini akan digunakan untuk menentukan lama waktu hidup pompa nutrisi.

#### 4.3.2 Pengujian Debit Pompa Air Sumur

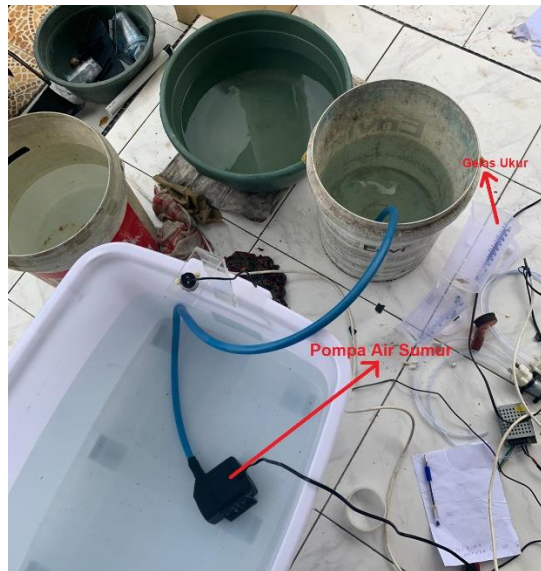
Pengujian ini ditujukan untuk mendapatkan nilai dari debit pompa air sumur yang akan digunakan pada sistem pengisian air dan nutrisi otomatis. Nilai dari debit tersebut digunakan pada pembuatan program sistem pengisian air dan nutrisi otomatis. Proses pengujian dilakukan dengan menghidupkan pompa selama 1 menit menggunakan *relay* arduino, kemudian air yang telah dipompa tersebut akan diukur

volume nya menggunakan gelas ukur. Data hasil pengujian pompa air akan ditampilkan sebagai berikut:

Tabel 4. 14 Data pengujian debit Pompa Air Sumur

No	Volume Air (Liter)	Waktu (Menit)	Debit (L/menit)
1	10,60 L	1	10,60 L
2	11,71 L	1	11,71
3	10,97 L	1	10,97
4	11,20 L	1	11,20
5	10,45 L	1	10,45
6	10,98 L	1	10,98
7	11,92 L	1	11,92
8	11,67 L	1	11,67
9	11,28 L	1	11,28
10	11,58 L	1	11,58
Rata-rata			11,24

Dari data tabel pengujian diatas, didapatkan debit rata-rata dari pompa air nutrisi yaitu 11,24 L/menit. Nilai ini akan digunakan dalam pembuatan program sistem pengisian air dan nutrisi otomatis.



Gambar 4. 38 Pengujian Nilai Debit Pompa Air Sumur

Proses pengujian diatas dilakukan dengan cara mengaktifkan pompa air sumur dalam waktu satu menit. Air yang telah dipompa ke dalam wadah yang lebih kecil akan diukur volumenya menggunakan gelas ukur.

#### 4.4 Pengujian Hubungan Penambahan Nutrisi A-B Mix Terhadap Kenaikan Nilai PPM

Pada tahap ini dilakukan pengujian untuk mengetahui pengaruh penambahan nutrisi A-B *Mix* terhadap penambahan nilai PPM air. Pengujian dilakukan dengan cara menambahkan tiap 0,5 ml nutrisi A dan 0,5 ml nutrisi B dalam 1 Liter air secara bertahap. Pengukuran nilai PPM dilakukan menggunakan TDS Meter. Dari hasil pengujian, akan diamati bagaimana kenaikan nilai PPM air tiap kali nutrisi A-B *Mix* ditambahkan. Data hasil percobaan akan ditampilkan berikut.

Tabel 4. 15 Hubungan Pemanbahan Nutrisi A-B *Mix* terhadap Kenaikan PPM

No	A-B <i>Mix</i> (ml)	TDS Meter (PPM)	Kenaikan PPM ( $\Delta$ PPM)
1	0	56	-
2	0,5	100	44
3	1	148	48



4	1,5	189	41
5	2	231	42
6	2,5	269	38
7	3	312	43
8	3,5	370	58
9	4	393	23
10	4,5	436	43
11	5	470	34
12	5,5	522	52
13	6	557	35
14	6,5	590	33
15	7	636	46
16	7,5	676	40
17	8	712	36
18	8,5	753	41
19	9	786	33
Rata-rata			40,55

Dari data pengujian tersebut, didapatkan rata-rata pada setiap penambahan 0,5 ml nutrisi A dan 0,5 ml nutrisi B pada 1 liter air, sebesar 40,55 PMM. Nilai kenaikan rata-rata ini akan digunakan pada pembuatan program sistem pengisian air dan nutrisi otomatis.



Gambar 4. 39 Pengujian Hubungan Penambahan Nutrisi AB *Mix* Terhadap Nilai PPM

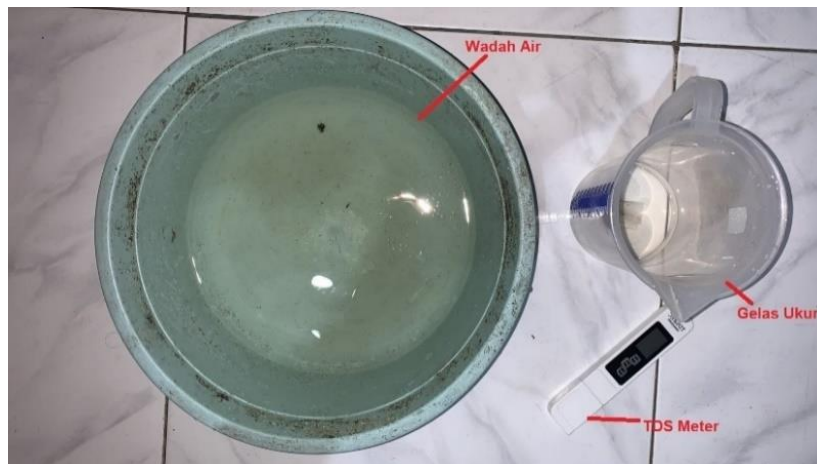
#### 4.5 Pengujian Hubungan Penambahan Air Terhadap Penurunan Nilai PPM

Pada tahap ini dilakukan pengujian untuk mengetahui pengaruh penambahan nutrisi air terhadap penurunan nilai PPM air. Pengujian dilakukan dengan cara menambahkan 100 ml air ke dalam 1 liter larutan nutrisi 1000 PPM secara bertahap. Pengukuran nilai PPM dilakukan menggunakan TDS meter. Dari hasil pengujian, akan diamati bagaimana penurunan nilai PPM pada larutan tiap kali dilakukan pengenceran dengan air. Data hasil percobaan akan ditampilkan berikut.

Tabel 4. 16 Hubungan Penambahan Air Terhadap Penurunan Nilai PPM

No	Penambahan Air (ml)	TDS Meter (PPM)	Penurunan PPM ( $\Delta$ PPM)
1	0	1009	-
2	100	916	94
3	200	855	61
4	300	796	59
5	400	761	35
6	500	720	40
7	600	677	43
8	700	636	41
9	800	617	20
10	900	582	35
11	1000	555	26
Rata-rata			45

Dari data tabel pengujian 4.16 diatas, didapatkan rata-rata penurunan nilai PPM dalam larutan pada tiap penambahan 100 ml air, yaitu 45 PPM. Nilai penurunan rata-rata ini akan digunakan dalam proses pengenceran larutan pada sistem pengisian air dan nutrisi hidroponik.



Gambar 4. 40 Pengujian Hubungan Penambahan Air Terhadap Nilai PPM

#### 4.6 Pengujian Data Waktu Pengosongan Air Pada Pipa Tanaman

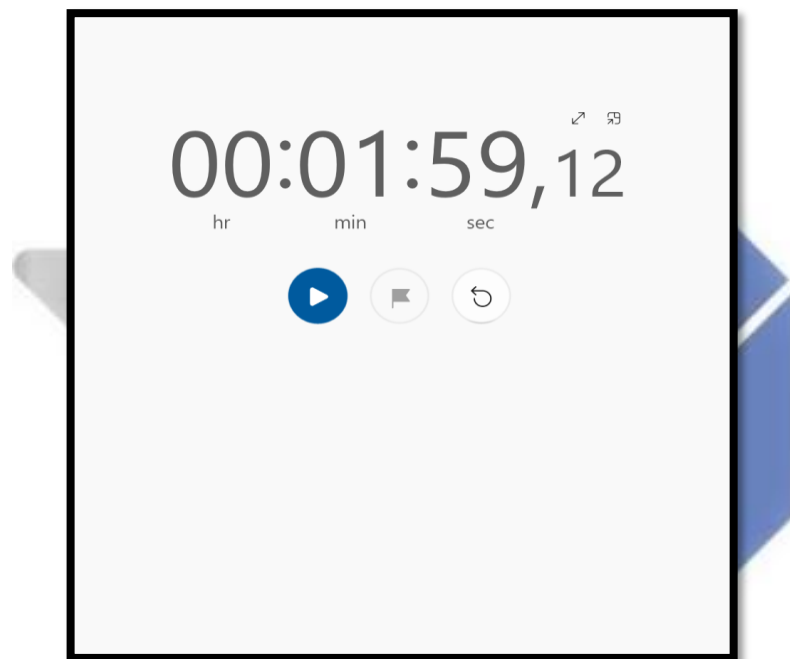
Tahap ini merupakan tahap pengujian untuk mendapatkan nilai dari lamanya waktu pada proses pengosongan air yang ada di dalam pipa hidroponik. Proses pengosongan air merupakan proses pada saat pompa distribusi air ke pipa tanaman telah terhenti, namun masih terdapat air didalam pipa hidroponik yang masih mengalir menuju bak hidropoonik. Pengujian dilakukan dengan menghitung waktu pengosongan menggunakan *stopwatch*. Berikut ini merupakan data hasil pengujian.

Tabel 4. 17 Data Pengujian Waktu Pengosongan Air pada Pipa Tanaman

Pengujian ke-	Waktu (detik)
1	119
2	120
3	119
4	121
5	121
6	120
7	122
8	120
9	119

10	120
Rata-rata	120

Dari tabel pengujian 4.17 diatas, didapatkan waktu rata-rata yang dibutuhkan pada saat pengosongan air dari pipa hidroponik adalah 120 detik. Data ini akan digunakan untuk memberikan *delay* pada program Arduino saat proses pengisian air dan nutrisi otomatis.



Gambar 4. 41 Proses Pengujian Data Waktu Pengosongan Air dengan *Stopwatch*

#### 4.7 Pengujian Sistem Pengisian Air dan Nutrisi Otomatis

Pengujian dilakukan untuk melihat seberapa akurat program yang telah dibuat dalam membuat air nutrisi yang akan digunakan pada sistem hidroponik. Parameter untuk pengujian dianggap berhasil adalah, sistem mampu membuat air nutrisi pada bak hidroponik dengan kisaran nilai 600-800 PPM.

Cara kerja dari sistem ini adalah pada saat sensor ultrasonik mendeteksi ketinggian air dibawah volume 30 liter, maka sistem akan otomatis mengisi kembali

air dan nutrisi dengan target kadar nutrisi 600-800 PPM. Berikut data hasil pengujian sistem pengisian air dan nutrisi otomatis.

Tabel 4. 18 Data pengujian sistem pengisian Air dan Nutrisi otomatis

Pengisian ke-	Hasil Pengisian Ulang (PPM)	Status ( ✓ )	Akurasi (%)
1	745	✓	100
2	765	✓	100
3	715	✓	100
4	729	✓	100
5	708	✓	100

Dari pengujian pada tabel 4.18 yang telah dilakukan sebanyak 5 kali, didapati jumlah keberhasilan sebanyak 5 kali. Berdasarkan data diatas, bisa dikatakan bahwa sistem dapat bekerja dengan baik, dengan tingkat keberhasilan 100%.



Gambar 4. 42 Pengujian Sistem Pengisian Air dan Nutrisi Otomatis

## 4.8 Pengujian Aplikasi

Pengujian aplikasi bertujuan untuk melihat apakah aplikasi *smartphone* dapat *memonitoring* pembacaan sensor pada sistem dengan benar, sebagaimana fungsi LCD. Serta dapat melakukan pengontrolan pada lampu *Growlight* dari jarak jauh melalui *smartphone*. Pada pengujian kontrol aplikasi ini, penulis menggunakan *relay* sebagai pengganti *Growlight*. Berikut data hasil pengujian.

### 4.8.1 Pengujian Pembacaan Sensor Pada Aplikasi

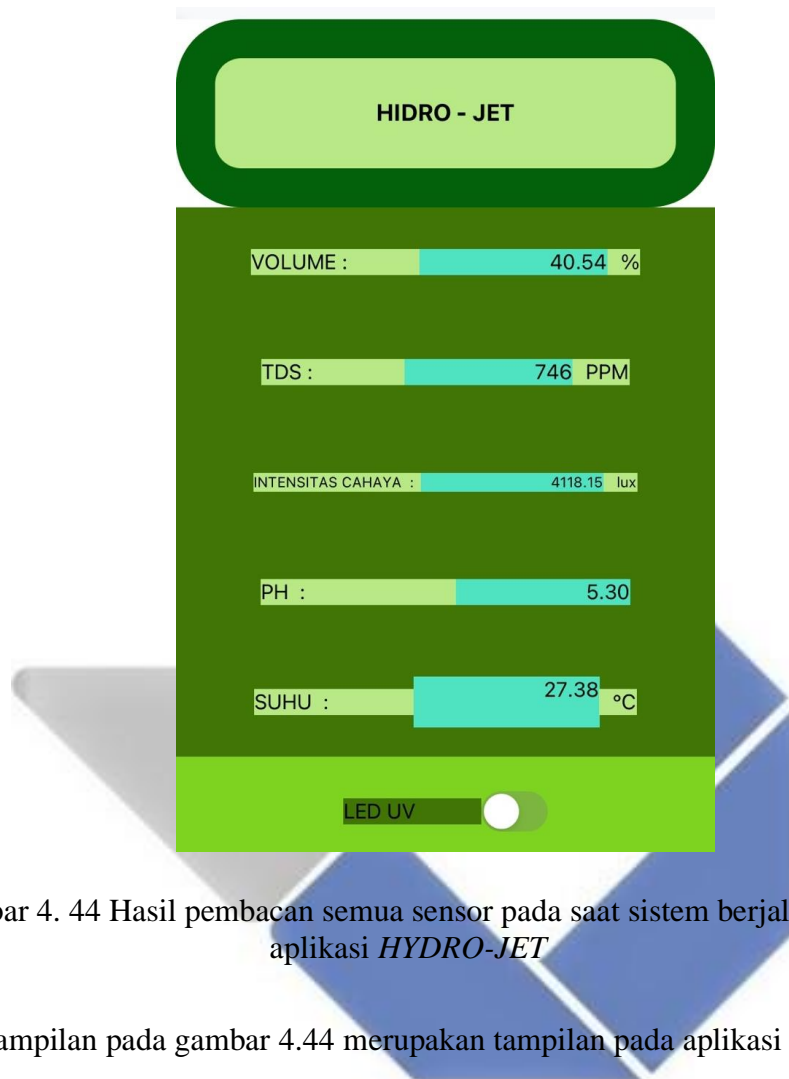
Berikut ini merupakan data hasil pengujian pembacaan sensor pada aplikasi *smatphone*. Pengujian dilakukan dengan membandingkan tampilan pembacaan sensor pada LCD dan aplikasi. Nilai sensor yang ditampilkan diantaranya adalah PPM, pH, intensitas cahaya (lux), volume air, dan suhu air.

Tabel 4. 19 Data pengujian sensor aplikasi *HYDRO-JET*

No	LCD					Aplikasi Smarhphone				
	PPM	pH	Lux	Volume Air	Suhu	PPM	pH	Lux	Volume Air	Suhu
1	746	5,30	4014,6	40,5%	27,37	746	5,30	4014,6	40,5 %	27,37
2	565	6,12	3583,1	35,2%	27,42	565	6,12	3583,1	35,2%	27,42
3	746	5,56	3890,5	35,8%	27,36	746	5,56	3890,5	35,8%	27,36
4	812	6,67	2512,4	51,2%	27,68	812	6,67	2512,4	51,2%	27,68
5	698	5,78	2918	37,5%	27,29	698	5,78	2918	37,5%	27,29



Gambar 4. 43 Hasil pembacaan semua sensor pada saat sistem berjalan pada LCD



Gambar 4. 44 Hasil pembacaan semua sensor pada saat sistem berjalan pada aplikasi *HYDRO-JET*

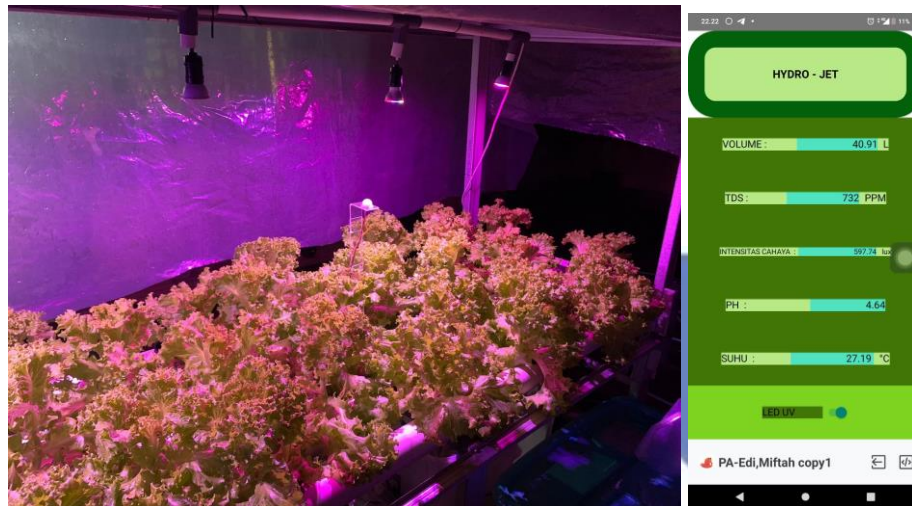
Tampilan pada gambar 4.44 merupakan tampilan pada aplikasi smatphone. Aplikasi dapat menampilkan hasil pembacaan dari kelima sensor secara *realtime*.

#### 4.8.2 Pengujian Kontrol Lampu *Growlight* Pada aplikasi

Tabel 4. 20 Data pengujian kontrol manual LED UV

No.	Kondisi saklar pada aplikasi	Kondisi <i>Growlight</i>	Keberhasilan (✓)	Akurasi (%)
1	Mati	Mati	✓	100
2	Hidup	Hidup	✓	100
3	Mati	Mati	✓	100
4	Hidup	Hidup	✓	100
5	Mati	Mati	✓	100

Dari data tabel pengujian 4.20 diatas, didapatkan hasil bahwa aplikasi *smartphone* dapat berfungsi sesuai dengan semestinya dengan tingkat akurasi 100%. Tampilan aplikasi dapat menampilkan hasil pembacaan sensor sebagaimana tampilan pada LCD. Sistem pengontrol lampu pada aplikasi juga dapat dikendalikan dengan baik.



Gambar 4. 45 Pengujian Kontrol Lampu dari Aplikasi *Smartphone*

#### 4.9 Pengujian Keseluruhan Alat pada Sistem Hidroponik

Tahap ini merupakan tahap pengujian terhadap keseluruhan alat dan sistem. Pengujian dibagi menjadi 3, yaitu pengujian sistem *monitoring* pada LCD, pengujian sistem *monitoring* pada aplikasi, dan pengujian pada kontrol lampu pada aplikasi. Tahap ini merupakan tahap akhir dari segala pengujian pada alat ini. Pengujian keseluruhan alat ini, meliputi pengujian sistem *monitoring*, yaitu kemampuan aplikasi IoT untuk dapat menampilkan hasil pembacaan sensor. Kemudian pengujian pada sistem kontrol otomatis untuk mengukur keberhasilan sistem dalam membuat larutan nutrisi pada bak hidroponik, dengan kadar nutrisi 600-800 PPM. Serta pengujian terhadap sistem kontrol lampu *growlight*.

##### 4.9.1 Pengujian Pembacaan Sensor Pada LCD

Pengujian ini merupakan pengujian hasil pembacaan sensor pada saat sistem berjalan keseluruhan. Hasil pembacaan dari masing-masing sensor akan



dibandingkan dengan hasil pembacaan alat ukur. Berikut merupakan data pembacaan sensor.

Tabel 4. 21 Data pengujian sensor keseluruhan sistem

No	PPM		Suhu ( C <sup>o</sup> )		pH		Lux	
	Sensor	Alat Ukur	Sensor	Alat Ukur	Sensor	Alat Ukur	Sensor	Alat Ukur
1	657	705	27,3	27,4	4,73	4,98	52451	47230
2	712	793	26,77	26,9	5,13	5,31	55582	52740
3	678	732	27,34	27,3	5,02	5,27	36359	32540
4	704	778	27,22	27,2	5,32	5,70	31610	28430
5	673	613	26,88	26,8	5,78	6,18	35323	32540
6	701	755	27,3	27,4	5,98	6,37	68310	62310
7	741	798	27,56	27,5	5,73	6,02	52371	47890
8	773	704	29,0	29,1	5,56	5,86	23289	20740

Tabel 4. 22 Tabel Tingkat Akurasi Pembacaan Sensor Keseluruhan Sistem

No	Akurasi (%)			
	TDS	Suhu	pH	Lux
1	89,65	99,63	94,71	90,05
2	88,62	99,51	94,43	94,89
3	92,04	99,85	95,02	89,50
4	89,49	99,93	92,86	89,94
5	91,08	99,70	93,08	92,12
6	95,15	99,63	93,48	91,22
7	90,83	99,78	94,94	91,44
8	91,07	99,66	94,60	89,05
Rata-Rata	91,07	99,71	94,14	91,03

Dari data pengujian kedua tabel 4.21 dan 4.22 diatas, didapatkan nilai akurasi rata-rata dari pembacaan sensor, yaitu sensor TDS dengan akurasi 91,07%, sensor suhu dengan akurasi 99,71%, sensor pH dengan akurasi 94,14%, dan sensor

lux dengan akurasi 91,03 %. Hal ini menunjukkan bahwa alat dapat membaca nilai PPM, pH, suhu, dan Lux dengan cukup baik.



Gambar 4. 46 Tampilan LCD Pengujian Sensor pada Keseluruhan Sistem

Pengujian dilakukan dengan untuk membandingkan hasil pembacaan sensor yang ditampilkan pada LCD, apakah sesuai dengan hasil pembacaan alat ukur. Pada gambar 4.47 menunjukkan proses pengujian nilai PPM dan Suhu dengan alat ukur.



Gambar 4. 47 Pengujian nilai PPM dan Suhu Air



Gambar 4. 48 Pengujian nilai pH Air dan Intensitas Cahaya

Pada proses pengujian ini, didapatkan juga bahwa pembacaan nilai PPM pada sensor TDS membutuhkan waktu respon untuk mencapai nilai pembacaan yang diinginkan. Sehingga dilakukan pula pengujian terhadap waktu respon pembacaan sensor TDS dengan cara memindahkan *probe* sensor TDS dari wadah yang berisikan 2 air dengan PPM berbeda. Berikut merupakan data hasil pengujian waktu respon pembacaan sensor TDS.

Tabel 4. 23 Data Pengujian Waktu Respon Pembacaan Sensor TDS

No.	PPM Awal	PPM Akhir	Waktu Respon (detik)
1	77	712	25
2	78	713	25
3	77	712	24
4	78	712	25
5	77	712	24
6	712	78	26

7	712	78	25
8	713	78	26
9	712	78	26
10	712	77	27
Rata-rata			25,3

Berdasarkan data hasil pengujian pada tabel 4.23, didapatkan hasil nilai waktu respon pada pembacaan sensor TDS yaitu 25,3 detik untuk mencapai pembacaan yang stabil.

#### 4.9.2 Pengujian *Monitoring* Pada Aplikasi

Pada tahap ini, pengujian dilakukan dengan membandingkan hasil pembacaan sensor yang ditampilkan pada LCD dengan tampilan pembacaan sensor pada aplikasi. Data hasil pembacaan sensor akan ditampilkan pada tabel dibawah.

Tabel 4. 24 Perbandingan Pembacaan Sensor pada LCD dan Aplikasi *Smartphone*

	Tampilan LCD	Tampilan Aplikasi
Sensor TDS (PPM)	746 PPM	746 PPM
Sensor pH (pH)	5,30	5,30
Sensor Suhu ( $C^{\circ}$ )	27,37 $C^{\circ}$	27,37 $C^{\circ}$
Volume air	40,50 %	40,50 %
Sensor Lux (lux)	4014,60 lux	4014,60 lux

Dari data tabel 4.24, menunjukkan bahwa aplikasi dapat menampilkan hasil pembacaan sensor dengan baik, sebagaimana fungsi dari LCD. Namun, pembacaan sensor pada aplikasi terdapat delay yang lebih besar daripada LCD. Hal ini dengan sengaja dilakukan agar tidak memberatkan kerja dari ESP32 dalam proses komunikasi serial dari Arduino.



Gambar 4. 49 Tampilan pembacaan sensor pada LCD

Pada pengujian ini hasil *monitoring* pembacaan sensor akan dibandingkan, antara pembacaan pada LCD seperti pada gambar 4.49 dengan hasil pembacaan pada aplikasi seperti pada gambar 4.50.



Gambar 4. 50 Tampilan pembacaan sensor pada aplikasi

### 4.9.3 Pengujian Kontrol Lampu Jarak Jauh Dengan Aplikasi

Tahap ini merupakan tahap pengujian untuk melihat apakah fungsi dari kontrol lampu dapat berfungsi dengan baik. Pengujian dilakukan pada beberapa jarak berbeda.

Tabel 4. 25 Data Hasil Pengujian Kontrol Lampu dengan Aplikasi *Smartphone*

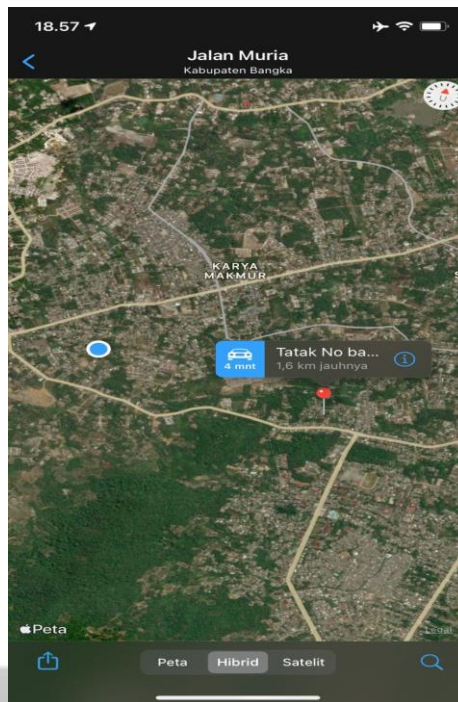
	Jarak Pengujian (m)	Fungsi Lampu <i>Growlight</i>		Akurasi (%)
		ON	OFF	
1	1 m	✓	✓	100
2	2 m	✓	✓	100
3	3 m	✓	✓	100
4	4 m	✓	✓	100
5	5 m	✓	✓	100
6	6 m	✓	✓	100
7	7 m	✓	✓	100
8	8 m	✓	✓	100
9	9 m	✓	✓	100
10	10 m	✓	✓	100
11	750 m	✓	✓	100
12	1600 m	✓	✓	100
13.	4400m	✓	✓	100

Berdasarkan data hasil pengujian diatas, kontrol lampu menggunakan aplikasi dapat bekerja dengan baik dengan tingkat akurasi 100%, meskipun pengujian berada pada jarak yang jauh sekalipun.



Gambar 4. 51 Pengujian Kontrol Lampu *Growlight*

Gambar 4.51 merupakan pengujian kontrol lampu *growlight* dari aplikasi smatphone. Tampilan pertama merupakan keadaan awal pada saat lampu dalam keadaan mati, sedangkan tampilan kedua merupakan keadaan pada saat lampu grow light diaktifkan. Pengujian diatas dilakukan pada malam hari.



Gambar 4. 52 Pengujian kontrol lampu berbasis aplikasi dari jarak jauh

Gambar 4.52 diatas merupakan tangkapan layar berupa jarak dari proses pengujian kontrol lampu berbasis aplikasi *smartphone*. Pengujian diatas dilakukan pada jarak 1,6 km dari kebun hidroponik.

#### 4.9.4 Hasil Pertanian

Tahap ini merupakan pengambilan data terhadap hasil pertanian dari sistem hidroponik yang telah dikontrol oleh sistem otomatis terintegrasi IoT. Data yang akan diambil adalah berat sayuran, tinggi sayuran, dan jumlah daun. Berikut ini merupakan data tabel hasil pertanian hidroponik.

Tabel 4. 26 Data Sayuran Hasil Pertanian Hidroponik

No.	Berat Sayuran (gr)	Tinggi Sayuran (cm)	Jumlah Daun
1	183	37	18
2	172	35	17
3	197	37	20
4	187	38	18



5	191	36	19
6	180	37	17
7	182	36	18
8	187	36	17
9	184	37	17
10	192	37	18
Rata rata	186	37	18

Berdasarkan data tabel 4.26 diatas, sistem hidroponik yang dilengkapi dengan sistem *monitoring* dan kontrol berbasis IoT dapat menghasilkan sayuran selada yang berusia 37 hari dengan berat rata-rata 186 gr, tinggi 37 cm, dan jumlah, daun 18 helai.



Gambar 4. 53 Proses Pengukuran Tinggi Sayuran

Gambar 4.53 merupakan proses pengukuran tinggi sayuran selada menggunakan alat ukur meteran. Berdasarkan pengujian yang telah dilakukan, didapatkan tinggi rata-rata dari tanaman selada adalah 37 cm.



Gambar 4. 54 Proses Menimbang Berat Sayuran Hasil Pertanian

Pada gambar 4.54 menunjukkan proses penimbangan berat dari sayuran selada menggunakan timbangan digital. Berdasarkan pengujian untuk menimbang berat sayur, didapatkan berat rata-rata dari sayur seberat 186 gr.

## BAB V PENUTUP

### 5.1 Kesimpulan

Berdasarkan hasil pengujian alat yang berjudul “ Implementasi Kebun Cerdas IMPLEMENTASI KEBUN CERDAS PADA PERKEBUNAN HIDROPONIK SISTEM *DEEP FLOW TECHNIQUE* (DFT) TERINTEGRASI IoT” diperoleh kesimpulan sebagai berikut:

1. Berdasarkan data hasil pengujian 3 buah sensor dalam satu waktu, didapati bahwa pembacaan dari sensor pH mengalami gangguan pada proses pembacaan. Hal ini disebabkan oleh karakteristik dari sensor TDS yang menghantarkan elektron dalam air, sehingga dapat mengganggu pembacaan sensor pH. Namun hal ini dapat diatasi dengan meletakkan sensor pH berjauhan dengan sensor TDS atau menggunakan *analog isolator* pada salah satu sensor tersebut.
2. Hasil pembacaan semua sensor yang telah dikalibrasi dan diuji cenderung menunjukkan tingkat akurasi yang tinggi, dimana nilai akurasi seluruh sensor berada diatas 90%. Hal ini menunjukkan bahwa sensor dapat digunakan dengan baik pada proses *monitoring*.
3. Sistem pengisian air dan nutrisi secara otomatis dapat bekerja dengan baik dengan tingkat akurasi 100%. Sistem mampu untuk membuat larutan nutrisi pada bak hidroponik dengan *range* kadar PPM yang diinginkan, yaitu 600-800 PPM.
4. Data sensor dari Arduino dapat dikirimkan ke dalam *firebase* dan berhasil ditampilkan pada aplikasi *smartphone*. Serta kontrol lampu melalui aplikasi dapat berjalan dengan tingkat akurasi 100% bahkan dari jarak jauh sekalipun.
5. Tampilan hasil pembacaan sensor pada aplikasi memiliki delay yang lebih besar dibandingkan LCD, hal ini dilakukan untuk menghindari eror pada saat proses komunikasi serial dari Arduino ke ESP32 dan tidak membebani ESP32.

6. Hasil pertanian dari sistem hidroponik DFT dengan menerapkan konsep kebun cerdas didapatkan sayuran selada dengan karakteristik berat rata-rata 186 gr , tinggi 37 cm, dan jumlah daun 18 helai.

## 5.2 Saran

Saran untuk pengembangan alat lebih lanjut dapat dengan menambahkan beberapa hal, antara lain :

1. Pengembangan pada sistem kontrol *growlight* yang dapat hidup dan mati sesuai dengan waktu terbit dan terbenam matahari, dengan bantuan RTC.
2. Menambahkan sistem kontrol otomatis dan *monitoring* kelembapan udara
3. Penggunaan metode kalibrasi lain untuk mendapatkan hasil pembacaan yang lebih akurat.



## DAFTAR PUSTAKA

- [1] H. A. Karim, F. Rossi, N. A. M. Arif, A. Sali, and R. Komiya, "Multiple description coding with side information for stereoscopic 3D," in *2012 International Symposium On Telecommunication Technologies*, 2012, pp. 245–248.
- [2] T. Susanto, S. D. Riskiono, R. Rikendry, and A. Nurkholis, "Implementasi Kendali Lqr Untuk Pengendalian Sikap Longitudinal Pesawat Flying Wing," *Electro Luceat*, vol. 6, no. 2, pp. 245–254, 2020.
- [3] A. Nurkholis, M. Muhaqiqin, and T. Susanto, "Analisis Kesesuaian Lahan Padi Gogo Berbasis Sifat Tanah dan Cuaca Menggunakan ID3 Spasial (Land Suitability Analysis for Upland Rice based on Soil and Weather Characteristics using Spatial ID3)," *JUITA: Jurnal Informatika*, vol. 8, no. 2, pp. 235–244, 2020.
- [4] W. Wajiran, S. D. Riskiono, P. Prasetyawan, and M. Iqbal, "Desain Iot Untuk Smart Kumbung Dengan Thinkspeak Dan Nodemcu," *POSITIF: Jurnal Sistem Dan Teknologi Informasi*, vol. 6, no. 2, pp. 97–103, 2020.
- [5] A. Mulyanto and R. Rikendry, "Sistem Kontrol Pergerakan Robot Beroda Pemadam Api," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 2007.
- [6] A. A. Endryanto and N. E. Khomariah, "Kontrol Dan Monitoring Tanaman Hidroponik Sistem Nutrient Film Technique Berbasis IoT," *Teknik Informatika*, no. 45, 2020.
- [7] B. WN, "Kelebihan dan Kekurangan Sistem Hidroponik DFT," *HIDROPONIKPEDIA*, Aug, vol. 6, 2018.
- [8] S. Firmansyah, C. Prihantoro, H. A. Haidar, and M. Z. F. Baihaqi, "Sistem Automasi Hidroponic Berbasis IOT," *LEDGER: Journal Informatic and Information Technology*, vol. 1, no. 2, pp. 84–90, 2022.

- [9] D. Kurniawan, "Regresi linier." Statistic, 2008.
- [10] N. H. Wijaya and S. Sutrimo, "Lux Meter as A Measuring Instrument for Operating Lamp Light Intensity Based on Arduino Uno R3," *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)*, vol. 8, no. 1, pp. 1–8, 2021.
- [11] B. A. Sakti, S. Prasetya, and I. Nuriskasari, "Analisis Pemilihan Sensor dan Ketelitian pada Rancang Bangun Weather Station Sebagai Monitoring System Cuaca Area Politeknik Negeri Jakarta," in *Prosiding Seminar Nasional Teknik Mesin*, 2022, no. 1, pp. 651–655.
- [12] R. N. Ilham, D. S. Harafany, and W. M. Setiavi, "KARAKTERISASI SISTEM PENDINGIN DALAMMENINGKATKAN EFISIENSI PANEL SURYA BERBASISINTERNET OF THINGS (IOT)," *repository PNJ*, Aug. 2021.
- [13] U. M. Arief, "Pengujian sensor ultrasonik ping untuk pengukuran level ketinggian dan volume air," *Jurnal Ilmiah Elektrikal Enjiniring UNHAS*, vol. 9, no. 2, pp. 72–77, 2011.
- [14] I. Muklisin, "Pendeteksi Volume Tandon Air Secara Otomatis Menggunakan Sensor Ultrasonic Berbasis Arduino Uno R3," *Jurnal Qua Teknika*, vol. 7, no. 2, pp. 55–65, 2017.
- [15] Z. Azmi and I. Saniman, "Sistem penghitung pH air pada tambak ikan berbasis mikrokontroler," *Jurnal Ilmiah Saindikom*, vol. 15, no. 2, pp. 101–108, 2016.
- [16] F. Karoba and R. Nurjasmi, "Pengaruh Perbedaan pH terhadap Pertumbuhan dan Hasil Tanaman Kailan (*Brassica oleraceae*) Sistem Hidroponik Nft (Nutrient Film Tecnique)," *Jurnal Ilmiah Respati*, vol. 6, no. 2, 2015.
- [17] U. Syafiqoh, S. Sunardi, and A. Yudhana, "Pengembangan Wireless Sensor Network Berbasis Internet of Things untuk Sistem Pemantauan Kualitas

- Air dan Tanah Pertanian,” *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, no. 2, pp. 285–289, 2018.
- [18] D. R. Wati and W. Sholihah, “Pengontrol pH dan Nutrisi Tanaman Selada pada Hidroponik Sistem NFT Berbasis Arduino,” *Teknik Komputer, Sekolah Vokasi, IPB University*, 2021.
- [19] A. N. Pratama, “Implementasi Sensor TDS (Total Dissolved Solids) untuk Kontrol Air Secara Otomatis pada Tanaman Hidroponik.,” *Repository Dinamika*, 2017.
- [20] A. Z. NADIAH, “SISTEM KENDALI POMPA AIR DAN POMPA PUPUK BERDASARKAN PERHITUNGAN NILAI PPM PADA TANAMAN HIDROPONIK,” *Scholar.unand*, 2017.
- [21] D. R. Tisna, B. J. M. Putra, T. Maharani, and H. Hasnira, “Metode Peningkatan Akurasi pada Sensor TDS Berbasis Arduino untuk Nutrisi Air Menggunakan Regresi Linier,” *Jurnal Integrasi*, vol. 14, no. 1, pp. 61–68, 2022.
- [22] H. Effendi and R. Puspitaningrum, “Rancang Bangun Sistem Monitoring Pemakaian Air Pam Dan Mutu Air Pada Komplek Perumahan Dengan Jaringan Nirkabel Lora Berbasis Arduino Uno,” *SINUSOIDA*, vol. 23, no. 1, pp. 50–60, 2021.
- [23] I. Asmbangnirwana, E. Endryansyah, P. W. Rusimamto, and M. S. Zuhrie, “Pengendalian Suhu Air Nutrisi Pada Hidroponik NFT (Nutrient Film Technique) Berbasis Fuzzy Logic Controller,” *JURNAL TEKNIK ELEKTRO*, vol. 11, no. 1, pp. 108–116, 2022.
- [24] A. N. Fathoni and U. Y. Oktiawati, “Blackbox Testing terhadap Prototipe Sistem Monitoring Kualitas Air Berbasis IoT,” *Jurnal Nasional Teknik Elektro dan Teknologi Informasi/ Vol*, vol. 10, no. 4, 2021.

- [25] Y. Rahmanto, A. Burlian, and S. Samsugi, "Sistem Kendali Otomatis Pada Akuaponik Berbasis Mikrokontroler Arduino UNO R3," *Jurnal Teknologi Dan Sistem Tertanam*, vol. 2, no. 1, pp. 1–6, 2021.
- [26] A. K. Putra, F. Fidiyanto, B. Prakoso, R. Z. Armantya, and F. H. S. al Haris, "PERAKITAN 3D PRINTER FUSED DEPOSITE MODELING (FDM) BERBASIS ARDUINO MEGA 2560," *Jurnal Gaung Informatika*, vol. 12, no. 2, 2019.
- [27] A. R. MOHAMMAD, "PENYEBAB PENURUNAN KINERJA POMPA SENTRIFUGAL TERHADAP PENDINGIN MESIN INDUK," *KARYA TULIS*, 2021.
- [28] S. A. Pohan and O. Oktojournal, "Pengaruh Konsentrasi Nutrisi AB Mix terhadap Pertumbuhan Caisim secara Hidroponik (Drip System)," *Lambung*, vol. 18, no. 1, pp. 20–32, 2019.
- [29] F. Zulrachman, H. Hasbi, and E. A. Kuncoro, "Uji Efektivitas Lampu LED Grow Light Terhadap Perbedaan Tinggi Media Tumbuh Tanaman," *Sriwijaya University*, 2019.
- [30] A. R. Restiani, S. Triyono, A. Tusi, and R. Zahab, "Pengaruh jenis lampu terhadap pertumbuhan dan hasil produksi tanaman selada (*Lactuca sativa* L.) dalam sistem hidroponik indoor," *Jurnal teknik pertanian Lampung*, vol. 4, no. 3, pp. 219–226, 2015.
- [31] Y. Yudhanto and A. Azis, *Pengantar Teknologi Internet of Things (IoT)*. UNSPress, 2019.
- [32] T. Rachmadi and S. Kom, *Mengenal Apa Itu Internet of Things*, vol. 1. Tiga Ebook, 2020.
- [33] T. Nursyahbani, R. Munadi, and N. B. Karna, "Pengembangan Sistem Parkir Pintar Berbasis Iot," *eProceedings of Engineering*, vol. 8, no. 5, 2021.



- [34] I. W. Suriana, I. G. A. Setiawan, and I. M. S. Graha, “Rancang Bangun Sistem Pengaman Kotak Dana Punia berbasis Mikrokontroler NodeMCU ESP32 dan Aplikasi Telegram,” *Jurnal Ilmiah Telsinas Elektro, Sipil dan Teknik Informasi*, vol. 4, no. 2, pp. 75–84, 2021.
- [35] A. NURROHMAT, “Pengertian Lengkap Thunkable, Kelebihan dan Kekurangannya,” [HTTPS://WWW.KELASHIRO.COM/2019/05/PENGERTIAN-LINGKAP-THUNKABLE-KELEBIHAN.HTML](https://www.kelashiro.com/2019/05/pengertian-lengkap-thunkable-kelebihan.html), May 29, 2019.





**LAMPIRAN 1**

**DAFTAR RIWAYAT HIDUP**

## Daftar Riwayat Hidup

### 1. Data Pribadi

Nama Lengkap : Muhammad Miftah Syahfiqri  
Tempat, Tanggal Lahir : Sungailiat, 28 Oktober 2000  
Alamat Rumah : Jl. Imam Bonjol Gg. Muria No.14  
No. HP : 081366830959  
Email : miftahsyahfiqri1@gmail.com  
Jenis Kelamin : Laki-Laki  
Agama : Islam



### 2. Riwayat Pendidikan

1. SD Negeri 25 Sungailiat Lulus 2012  
2. SMP Negeri 2 Sungailiat Lulus 2015  
3. SMA Negeri 1 Sungailiat Lulus 2018  
4. Politeknik Manufaktur Negeri Bangka  
Belitung 2019 - Sekarang

### 3. Pendidikan Non Formal

Sungailiat, 16 Februari 2023

Muhammd Miftah Syahfiqri

## Daftar Riwayat Hidup

### 1. Data Pribadi

Nama Lengkap : Edi Kuswara  
Tempat, Tanggal Lahir : Sungailiat, 12 Juni 2001  
Alamat Rumah : Jl. Sambu 2 No. 47 Air Ruai  
No. HP : 08828751110 / 0895413950690  
Email : wawa26980@gmail.com  
Jenis Kelamin : Laki-Laki  
Agama : Islam



### 2. Riwayat Pendidikan

1. SD Negeri 2 Sungailiat Lulus 2013
2. SMP Negeri 1 Sungailiat Lulus 2016
3. SMA Negeri 1 Sungailiat Lulus 2019
4. Politeknik Manufaktur Negeri Bangka Belitung 2019 - Sekarang

### 3. Pendidikan Non Formal

Sungailiat, 16 Februari 2023

Edi Kuswara



**LAMPIRAN 2**

**PROGRAM ARDUINO DAN NodeMCU ESP32**

## Progam Arduino

```
// ONEmark Project
#include <OneWire.h> //sensor suhu
#include <Wire.h> //sensor lux
#include <LiquidCrystal_I2C.h> //LCD
#include <BH1750.h> //sensor lux
#include "RTCLib.h" //RTC

String dataKirim;
unsigned long previousSerialMillis = 0;

bool wPengosongan;
float WAKTU_PENGOSONGAN = 120; //119.968 detik
float WAKTU_PENYESUAIAN = 40; //31.422 detik

//RTC
RTC_DS3231 rtc;
char dataHari[7][12] = {"Minggu", "Senin", "Selasa", "Rabu",
"Kamis", "Jumat", "Sabtu"};
String hari;
int tanggal, bulan, tahun, jam, menit, detik;

bool blmDisimpan;
bool yTerpenuhi;

int simpanDetik;
int simpanMenit;
int simpanJam;
int simpanTanggal;
int simpanBulan;
int simpanTahun;
int detikBerlalu;

//LCD
#define I2C_ADDR 0x27 //or(0x3F) I2C adress, you should use the
code to scan the adress first (0x27) here
```

```

#define BACKLIGHT_PIN 3 // Declaring LCD Pins
#define En_pin 2
#define Rw_pin 1
#define Rs_pin 0
#define D4_pin 4
#define D5_pin 5
#define D6_pin 6
#define D7_pin 7

LiquidCrystal_I2C
lcd(I2C_ADDR,En_pin,Rw_pin,Rs_pin,D4_pin,D5_pin,D6_pin,D7_pin);

//Ultrasonic
int trig_pin = 11;
int echo_pin = 12;
long echotime;
float distance;

float panjangBak;
float lebarBak;
float tinggiBak;
float volumeBak;
float tinggi_sensor;
float volume_sensor;
float jarak_sensor_ke_tandon = 15;
float volume;
float simpanJarak;

//Relay
int pompa_air = 5;
int pompa_distribusi = 4;
int pompa_mix = 3;
int pompa_nutrisi = 2;
int lampu = 6;

// Lux Meter
float lux;
float konstanta_lux;

```

```

float hasillux;
BH1750 lightMeter;

// Sensor Suhu
int DS18S20_Pin = 7; //DS18S20 Signal pin on digital 7
  //Temperature chip i/o
  OneWire ds(DS18S20_Pin); // on digital pin 7
  float temperature_suhu;

//pH & TDS
#define LED 13

//Sensor pH
#define SensorPin A0 //pH meter Analog output to
Arduino Analog Input 0
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40 //times of collection
int pHArray[ArrayLenth]; //Store the average value of the sensor
feedback
int pHArrayIndex=0;
static float pHValue,voltage_ph;

//Sensor TDS
#define TDSSensorPin A1
#define banyakSampling 20
#define printInterval 800
#define timeCollection 40 //times of collection
int TDSArray[timeCollection]; //Store the average value of the
sensor feedback
int TDSArrayIndex=0;
int ADC_TDS;
int tdsValue;
double TeganganADC;

//KONVERSI PPM
float PPM_Sekarang;
float butuhPPM;

```



```

float total_butuhPPM;
float kurangPPM;
float lebihPPM;
float Pompa_nutrisi_harus_hidup;
float Pompa_nutrisi_harus_hidup_penyesuaian;
float Total_hidup_pompaNutrisi;
float simpanPPM;
float total_butuhPPM_penyesuaian;
float butuhAir;
float butuhAir_Penyesuaian;
float total_butuhAir;
float Pompa_air_harus_hidup;
float ketentuan = 700;

float nutrisiAB = 0.5; //(ml nutrisi AB)
float hasil_nutrisi = 40.51; //(PPM per 1L air, dari 0.5ml AB)
float air = 100; //(ml air)
float berkurang_PPM = 45.38; //(PPM per 1L air, dari 0.5ml AB)

float real_nutrisi_keluar = 27.42; //(detik) pompa hidup 1 detik
mengeluarkan 27,42 ml nutrisi AB
float waktu_pompa_nutrisi = 1; //(detik) pompa hidup 1 detik
mengeluarkan 27,42 ml nutrisi AB
float real_air_keluar = 11268; //(detik) pompa hidup 1 detik
mengeluarkan 11268 ml / 11.268 L air
float waktu_pompa_air = 60; //(detik) pompa hidup 60 detik
mengeluarkan 11.268 L air

void setup() {

    Serial.begin(115200);
    Serial1.begin(9600);

    pengaturan_pin();

}

void loop() {

```

```

    digitalWrite(pompa_mix, LOW); //pompa mix selalu hidup untuk
memastikan PPM merata

    ambildata_komponen(); //baca sensor

    //KONTROL ISI ULANG AIR
    // -Air Penuh (CEK POINT)
    if (distance >= 25.88 && distance <= 28.97) { //70 liter = 28.97
        cekUlang();
    }

    // -Air Habis
    else if (distance >= 42.57) { //30 liter = 42.57 ; 50 liter =
35.63
        wPengosongan = true;
        isiUlang();
    }

    //Harus distribusi
    else if (distance < 25.88) { //80 liter = 25.88
        distribusi();
    }

    while (Serial1.available()) {
        char data = (char)Serial1.read();
        Serial.println(data);
        if (data == '1') {
            digitalWrite(6, LOW);
        }
        else if (data == '0') {
            digitalWrite(6, HIGH);
        }
    }

    display();
    delay(500);
}

```

```

void pengaturan_pin() {

    //RTC
    if (! rtc.begin()) {
        //Serial.println("RTC Tidak Ditemukan");
        Serial.flush();
        abort();
    }

    //Atur Waktu, upload untuk set waktu, kemudian beri komentar
    lalu upload kembali
    //rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    //rtc.adjust(DateTime(2022,12,31,23,59,0));

    //Ultrasonic
    pinMode(trig_pin, OUTPUT);
    pinMode(echo_pin, INPUT);
    digitalWrite(trig_pin, LOW);

    //Relay
    pinMode(pompa_air, OUTPUT);
    pinMode(pompa_distribusi, OUTPUT);
    pinMode(pompa_mix, OUTPUT);
    pinMode(pompa_nutrisi, OUTPUT);
    pinMode(lampu, OUTPUT);

    //Lux
    Wire.begin();
    // On esp8266 you can select SCL and SDA pins using
    Wire.begin(D4, D3);
    // For Wemos / Lolin D1 Mini Pro and the Ambient Light shield
    use Wire.begin(D2, D1);
    lightMeter.begin();

    //TDS & PH
    pinMode(LED,OUTPUT);

    //LCD

```

```

lcd.begin (20,4);
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH); //Lighting backlight
lcd.home ();

//Kondisi awal relay
digitalWrite(pompa_air, HIGH);
digitalWrite(pompa_distribusi, LOW);
digitalWrite(pompa_mix, HIGH);
digitalWrite(pompa_nutrisi, HIGH);
digitalWrite(lampu, HIGH);

}
void ambildata_komponen() {

//SENSOR CAHAYA
lux = lightMeter.readLightLevel();
if (lux <= 5000){
    hasillux = (2.561*lux) - 4.091; //kurang dari 5000
}
else if (lux > 5000 && lux < 28000){
    hasillux = (2.8229*lux) - 54.308; //pembacaan 5000-28000
}
else if (lux >= 28000){
    hasillux = (3.2516*lux) - 1189.8; //pembacaan lebih dari 28000
}

//ULTRASONIC
digitalWrite(trig_pin, HIGH);
delayMicroseconds(10);
digitalWrite(trig_pin, LOW);

echotime = pulseIn(echo_pin, HIGH);
//distance = (float)echotime;
//distance = 0.0001*((float)echotime*340.0)/2.0;
distance = (0.0177*(float)echotime) + 0.1573; //regresi 2

//panjangBak = 67.5; //satuan cm

```

```

//lebarBak = 49.5;
//tinggiBak = 37.5;
//volumeBak = (panjangBak*lebarBak*tinggiBak)/1000; //kurleb
130 liter
//tinggi_sensor = tinggiBak - (distance -
jarak_sensor_ke_tandon); //sensor ultrasonik biasa
//volume_sensor = (panjangBak*lebarBak*tinggi_sensor)/1000;
//volume terkini
//volume_sensor = (2.9025*tinggi_sensor) + 1.4391; //regresi
volume_sensor = (-2.9025*distance) + 153.82; //regresi
volume = volume_sensor;

//RTC
DateTime now = rtc.now();
hari = dataHari[now.dayOfTheWeek()];
tanggal = now.day();
bulan = now.month();
tahun = now.year();
jam = now.hour();
menit = now.minute();
detik = now.second();

//Sensor Suhu
temperature_suhu = (1.0204* getTemp()) - 0.5033;
// Serial.print("Temp : ");
// Serial.println(temperature);
// delay(100); //just here to slow down the output so it is
easier to read

//TDS
//static
static unsigned long waktuSampel = millis();
static unsigned long waktuPrint = millis();

if(millis()-waktuSampel > banyakSampling)
{
    TDSArray[TDSArrayIndex++]=analogRead(TDSSensorPin);
    if(TDSArrayIndex==timeCollection)TDSArrayIndex=0;
}

```

```

        //voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
        ADC_TDS= avergeIdx(TDSArray, timeCollection);
        TeganganADC = (ADC_TDS*5.0)/1024.0; // convert ADC 5v 10 bit
ke Tegangan (V)
        //tdsValue = 2.1353*ADC_TDS + 6.2311; //Regresi 1
        //tdsValue = 2.1564*ADC_TDS + 0.2725; //Regresi 2
        tdsValue = 2.0384*ADC_TDS - 23.914; //sistem
        waktuSampel=millis();
    }
    if(millis() - waktuPrint > banyakSampling)
    {
        // Serial.print("ADC:");
        // Serial.print( ADC_TDS);
        // Serial.print(" Voltage:");
        // Serial.print(TeganganADC,3);
        // Serial.print(" PPMN value: ");
        // Serial.println(TDSValue);
        waktuPrint=millis();
    }

//Sensor pH
static unsigned long samplingTime = millis();
static unsigned long printTime = millis();

if(millis() - samplingTime > samplingInterval)
{
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage_ph = avergearray_pH(pHArray, ArrayLenth)*5.0/1024;
    //pHValue = (avergearray_pH(pHArray, ArrayLenth)
+8.7084)/55.783; //regresi sensor sendiri
    pHValue = (0.0186*avergearray_pH(pHArray, ArrayLenth)) -
0.2695; //regresi sensor bersamaan
    samplingTime=millis();
}
if(millis() - printTime > printInterval)
{
    // Serial.print("Voltage:");

```

```

//      Serial.print(voltage,2);
//      Serial.print("    pH value: ");
//Serial.println(pHValue,2);
        digitalWrite(LED,digitalRead(LED)^1);
        printTime=millis();
    }

    tampilan_serial();
}
void distribusi () {
    digitalWrite(pompa_distribusi, LOW);
    digitalWrite(pompa_air, HIGH);
    digitalWrite(pompa_nutrisi, HIGH);
    detikBerlalu = 0;
    blmDisimpan = true;
}
void cekUlang() {

    blmDisimpan = true;

    digitalWrite(pompa_distribusi, HIGH);
    digitalWrite(pompa_air, HIGH);

    //PPM kurang
    if (tdsValue < 700){
    waktu_penyesuaian();
        while(tdsValue < 700){
            ambildata_komponen();
            simpan_data();
            while(yTerpenuhi == true){

                ambildata_komponen();
                PPM_Sekarang = simpanPPM;

                //Perhitungan lama hidup pompa nutrisi
                kurangPPM = ketentuan - PPM_Sekarang;
                butuhPPM = (kurangPPM/hasil_nutrisi)*nutrisiAB;
                total_butuhPPM = butuhPPM*volume_sensor;
            }
        }
    }
}

```

```

        Pompa_nutrisi_harus_hidup =
(waktu_pompa_nutrisi*total_butuhPPM)/real_nutrisi_keluar;

        digitalWrite(pompa_nutrisi, LOW);
        logikaRTC();
        if (detikBerlalu > Pompa_nutrisi_harus_hidup){
            digitalWrite(pompa_nutrisi, HIGH);
            yTerpenuhi = false;
            blmDisimpan = true;
            detikBerlalu = 0;
        }

        display();
        delay(500);
    }

    //menunggu sampai teraduk rata
    waktu_penyesuaian();

    display();
    delay(500);
}

}

//-----

//PPM lebih
//cuma dilakukan sekalipenyesuaian, dikarenakan batas tandon
else if (tdsValue > 800){
    //while (tdsValue > 800){
        ambildata_komponen();
        simpan_data();
        while(yTerpenuhi == true){

            ambildata_komponen();
            PPM_Sekarang = simpanPPM;

```



```

//Perhitungan lama hidup pompa air
lebihPPM = PPM_Sekarang - ketentuan;
butuhAir = (lebihPPM/berkurang_PPM)*air;
total_butuhAir = butuhAir*volume_sensor;
Pompa_air_harus_hidup =
(waktu_pompa_air*total_butuhAir)/real_air_keluar;

digitalWrite(pompa_air, LOW);
logikaRTC();
    if (detikBerlalu > Pompa_air_harus_hidup | distance <
25.88){ //ketika 80 liter, pompa air mati
        digitalWrite(pompa_air, HIGH);
        yTerpenuhi = false;
        blmDisimpan = true;
        detikBerlalu = 0;
    }

    display();
    delay(500);
}

//menunggu sampai teraduk rata
waktu_penyesuaian();

display();
delay(500);
//}
}

//-----

//step 2 (ketika ppm sudah terpenuhi)
else if (tdsValue >= 700 && tdsValue <= 800){
    distribusi();
}
}

void isiUlang(){

```

```

blmDisimpan = true;

digitalWrite(pompa_distribusi, HIGH);
delay(100);

//step 1 (waktu Pengosongan)
if (wPengosongan == true){
    pengosongan();
    wPengosongan = false;
}

//step 2 (penyesuaian nutrisi)
//PPM kurang
if (wPengosongan == false) {
    digitalWrite(pompa_air, LOW);
}
}

void tampilan_serial() {

    if ((millis()-previousSerialMillis) > 1000) {
        previousSerialMillis = millis();

        //variabel penampung data yg dikirim
        //dataKirim = "10#30.65#29"
        dataKirim = String(volume) + "#" + String(hasillux) + "#" +
String(tdsValue) + "#" + String(pHValue) + "#" +
String(temperature_suhu);
        //kirim data
        Serial.println(dataKirim);
        Serial1.println(dataKirim);
    }

}

void display() {

    //LCD

```

```

    lcd.clear();
    lcd.setCursor(16,0);
    lcd.print(detikBerlalu);
    //-----
    lcd.setCursor(0,0);
    lcd.print("V  : ");
    lcd.setCursor(5,0);
    lcd.print(volume_sensor);
    lcd.setCursor(11,0);
    lcd.print("L");
    //-----
    lcd.setCursor(0,1);
    lcd.print("PPM: ");
    lcd.setCursor(5,1);
    lcd.print(tdsValue);
    lcd.setCursor(10,1);
    lcd.print("PPM");
    //-----
    lcd.setCursor(0,2);
    lcd.print("Lux: ");
    lcd.setCursor(5,2);
    lcd.print(hasillux);
    lcd.setCursor(14,2);
    lcd.print("lx");
    //-----
    lcd.setCursor(0,3);
    lcd.print("pH : ");
    lcd.setCursor(5,3);
    lcd.print(pHValue);
    //-----
    lcd.setCursor(11,3);
    lcd.print("C: ");
    lcd.setCursor(14,3);
    lcd.print(temperature_suhu);

}
void pengosongan() {

```

```

simpan_data();

while(yTerpenuhi == true){

    ambildata_komponen();
    logikaRTC();

    if (detikBerlalu > WAKTU_PENGOSONGAN){ //2 menit
    yTerpenuhi = false;
    blmDisimpan = true;
    detikBerlalu = 0;
    }

    //Pembacaan sensor
    display();
    delay(500);
}

}
void waktu_penyesuaian() {

simpan_data();

while(yTerpenuhi == true){

    ambildata_komponen();
    logikaRTC();

    if (detikBerlalu > WAKTU_PENYESUAIAN){ //40 detik
    yTerpenuhi = false;
    blmDisimpan = true;
    detikBerlalu = 0;
    }

    //Pembacaan sensor
    display();
    delay(500);
}
}

```

```

}
void simpan_data() {
    if(blmdisimpan == true)
    {
        simpanDetik = detik;
        simpanMenit = menit;
        simpanJam = jam;
        simpanTanggal = tanggal;
        simpanBulan = bulan;
        simpanTahun = tahun;

        simpanJarak = distance;
        simpanPPM = tdsValue;

        blmdisimpan = false; //simpan semua data setelah
pengosongan
        yTerpenuhi = true;
    }
}
void logikaRTC() {
    //Logika waktu RTC bertambah 1 detik
    if (tahun == simpanTahun){
        if (bulan == simpanBulan){
            if (tanggal == simpanTanggal){
                if (jam == simpanJam){
                    waktu();
                }
                else if (jam > simpanJam){
                    waktu();
                }
            }
            else if (tanggal > simpanTanggal){
                if (jam < simpanJam){
                    waktu();
                }
            }
        }
    }
}

```

```

else if (bulan > simpanBulan){
    if (tanggal < simpanTanggal){
        if (jam < simpanJam){
            waktu();
        }
    }
}
else if(tahun > simpanTahun){
    if(bulan < simpanBulan){
        if(tanggal < simpanTanggal){
            if (jam < simpanJam){
                waktu();
            }
        }
    }
}
}
void waktu() {
    //MENIT > SIMPANMENIT
    if (detik < simpanDetik && menit > simpanMenit)
    {
        detikBerlalu = ((detik+((menit - simpanMenit)*60)) -
simpanDetik);
    }
    else if (detik == simpanDetik && menit > simpanMenit)
    {
        detikBerlalu = (detik - simpanDetik) + ((menit -
simpanMenit)*60);
    }
    else if (detik > simpanDetik && menit > simpanMenit)
    {
        detikBerlalu = (detik - simpanDetik) + ((menit -
simpanMenit)*60);
    }

    //MENIT = SIMPANMENIT
    else if (detik > simpanDetik && menit == simpanMenit)

```

```

    {
        detikBerlalu = detik - simpanDetik;
    }

//MENIT < SIMPANMENIT
    if (detik < simpanDetik && menit < simpanMenit)
    {
        detikBerlalu = ((detik+(((menit + 60) -
simpanMenit)*60)) - simpanDetik);
    }
    else if (detik == simpanDetik && menit < simpanMenit)
    {
        detikBerlalu = (detik - simpanDetik) + (((menit + 60)
- simpanMenit)*60);
    }
    else if (detik > simpanDetik && menit < simpanMenit)
    {
        detikBerlalu = (detik - simpanDetik) + (((menit + 60)
- simpanMenit)*60);
    }
}
float getTemp() {
    //returns the temperature from one DS18S20 in DEG Celsius

    byte data[12];
    byte addr[8];

    if ( !ds.search(addr)) {
        //no more sensors on chain, reset search
        ds.reset_search();
        return -1000;
    }

    if ( OneWire::crc8( addr, 7) != addr[7]) {
        Serial.println("CRC is not valid!");
        return -1000;
    }
}

```

```

if ( addr[0] != 0x10 && addr[0] != 0x28) {
    Serial.print("Device is not recognized");
    return -1000;
}

ds.reset();
ds.select(addr);
ds.write(0x44,1); // start conversion, with parasite power on at
the end

byte present = ds.reset();
ds.select(addr);
ds.write(0xBE); // Read Scratchpad

for (int i = 0; i < 9; i++) { // we need 9 bytes
    data[i] = ds.read();
}

ds.reset_search();

byte MSB = data[1];
byte LSB = data[0];

float tempRead = ((MSB << 8) | LSB); //using two's compliment
float TemperatureSum = tempRead / 16;

return TemperatureSum;
}

double avergearray_pH(int* arr, int number) {
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/n");
        return 0;
    }

```



```

}
if(number<5){ //less than 5, calculated directly statistics
    for(i=0;i<number;i++){
        amount+=arr[i];
    }
    avg = amount/number;
    return avg;
}else{
    if(arr[0]<arr[1]){
        min = arr[0];max=arr[1];
    }
    else{
        min=arr[1];max=arr[0];
    }
    for(i=2;i<number;i++){
        if(arr[i]<min){
            amount+=min; //arr<min
            min=arr[i];
        }else {
            if(arr[i]>max){
                amount+=max; //arr>max
                max=arr[i];
            }else{
                amount+=arr[i]; //min<=arr<=max
            }
        }
    }
    avg = (double)amount/(number-2);
}
return avg;
}

double avergeIdx(int* idx, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){

```

```

    Serial.println("Error number for the array to avraging!\n");
    return 0;
}
if(number<5){ //less than 5, calculated directly statistics
    for(i=0;i<number;i++){
        amount+=idx[i];
    }
    avg = amount/number;
    return avg;
}else{
    if(idx[0]<idx[1]){
        min = idx[0];max=idx[1];
    }
    else{
        min=idx[1];max=idx[0];
    }
    for(i=2;i<number;i++){
        if(idx[i]<min){
            amount+=min; //arr<min
            min=idx[i];
        }else {
            if(idx[i]>max){
                amount+=max; //arr>max
                max=idx[i];
            }else{
                amount+=idx[i]; //min<=arr<=max
            }
        }
    }
    avg = (double)amount/(number-2);
}
return avg;
}

```

## Program ESP 32

```
#include <Arduino.h>
#include <WiFi.h>
#include <Firebase_ESP_Client.h>
#include "time.h"

// Provide the token generation process info.
#include "addons/TokenHelper.h"
// Provide the RTDB payload printing info and other helper
functions.
#include "addons/RTDBHelper.h"

// Insert your network credentials
  //#define WIFI_SSID "HSTFARM" //hp tatak
  //#define WIFI_PASSWORD "28282828"

  //#define WIFI_SSID "RZL" //biznet
  //#define WIFI_PASSWORD "28282828"

  //#define WIFI_SSID "HIDROJET" //laptop
  //#define WIFI_PASSWORD "SUNGAILIAT"

  //#define WIFI_SSID "iPhone" //hp edi
  //#define WIFI_PASSWORD "13131313"

  //#define WIFI_SSID "TIREX" //laptop andre
  //#define WIFI_PASSWORD "satusampaikosong"

#define WIFI_SSID "hotslenovo"
#define WIFI_PASSWORD "12345678"

#define RXPIN2 16 //Connect to PIN D3 Arduino
```

```

#define TXPIN2 17 //Connect to PIN D2 Arduino

// Insert Firebase project API Key
#define API_KEY "AIzaSyD-d8qKEmtyFCGfzUV8VDesL2Av2F33uJg"

// Insert Authorized Email and Corresponding Password
#define USER_EMAIL "wawa26980@gmail.com"
#define USER_PASSWORD "Kusw@ra12"

// Insert RTDB URLdefine the RTDB URL
#define DATABASE_URL "https://proyek-akhir-ca7c2-default-
rtdb.asia-southeast1.firebaseio.com/"

// Define Firebase objects
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

// Variable to save USER UID
String uid;

// Database main path (to be updated in setup with the user UID)
String databasePath;
String readPath;
String updatePath;

// Database child nodes
String volumePath = "/volume";
String tdsPath = "/tds";
String cahayaPath = "/cahaya";
String phPath = "/ph";
String suhuPath = "/suhu";

FirebaseJson json;
String dataReceived = "";
String dataParsing[5];
String volume = "0";
String tds = "0";

```

```

String cahaya = "0";
String ph = "0";
String suhu = "0";
int count = 0;
int lastOnlineButton = 9;

const char* ntpServer = "pool.ntp.org";

// Timer variables (send new readings every three minutes)
unsigned long sendDataPrevMillis = 0;
unsigned long timerDelay = 10000;

unsigned long readDataPrevMillis = 0;

// Initialize WiFi
void initWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
  Serial.println();
}

unsigned long getTime() {
  time_t now;
  struct tm timeinfo;
  if (!getLocalTime(&timeinfo)) {
    //Serial.println("Failed to obtain time");
    return(0);
  }
  time(&now);
  return now;
}

```

```

void setup(){
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RXPIN2, TXPIN2);

  initWiFi();
  configTime(0, 0, ntpServer);

  // Assign the api key (required)
  config.api_key = API_KEY;

  // Assign the user sign in credentials
  auth.user.email = USER_EMAIL;
  auth.user.password = USER_PASSWORD;

  // Assign the RTDB URL (required)
  config.database_url = DATABASE_URL;

  Firebase.reconnectWiFi(true);
  fbdo.setResponseSize(4096);

  // Assign the callback function for the long running token
  generation task */
  config.token_status_callback = tokenStatusCallback; //see
  addons/TokenHelper.h

  // Assign the maximum retry of token generation
  config.max_token_generation_retry = 5;

  // Initialize the library with the Firebase authen and config
  Firebase.begin(&config, &auth);

  // Getting the user UID might take a few seconds
  Serial.println("Getting User UID");
  while ((auth.token.uid) == "") {
    Serial.print('.');
    delay(1000);
  }
  // Print user UID

```

```

uid = auth.token.uid.c_str();
Serial.print("User UID: ");
Serial.println(uid);

// Update database path
databasePath = "/UsersData/" + uid + "/sensor";
readPath = "/UsersData/" + uid + "/command/uv";
updatePath = "/command";
}

void loop(){

while (Serial2.available()) {
  char tempData = (char)Serial2.read();
  if(tempData != '\n') {
    dataReceived += tempData;
  }
  else {
    dataParsing [0] = "";
    dataParsing [1] = "";
    dataParsing [2] = "";
    dataParsing [3] = "";
    dataParsing [4] = "";
    int index = 0;
    for (int i=0; i < dataReceived.length(); i++) {
      if (dataReceived[i] != '#') {
        dataParsing[index] += dataReceived[i];
      }
      else {
        index ++;
      }
    }
    volume = dataParsing[0];
    cahaya = dataParsing[1];
    tds = dataParsing[2];
    ph = dataParsing[3];
    suhu = dataParsing[4];
    dataReceived = "";
  }
}

```

```

        index = 0;
    }
}

// Send new readings to database
if (Firebase.ready() && (millis() - sendDataPrevMillis >
timerDelay || sendDataPrevMillis == 0)){
    sendDataPrevMillis = millis();
    Serial.println(volume);
    Serial.println(cahaya);
    Serial.println(tds);
    Serial.println(ph);
    Serial.println(suhu);

    FirebaseJson updateData;
    updateData.add(volumePath, volume);
    updateData.add(cahayaPath, cahaya);
    updateData.add(tdsPath, tds);
    updateData.add(phPath, ph);
    updateData.add(suhuPath, suhu);
    Serial.printf("Set data json... %s\n",
Firebase.RTDB.updateNode(&fbdo, databasePath.c_str(), &updateData)
? "ok" : fbdo.errorReason().c_str());
}

if ((millis()-readDataPrevMillis) > 1000) {
    readDataPrevMillis = millis();
    if (Firebase.RTDB.getString(&fbdo, readPath.c_str())) {
        //if (fbdo.dataTypeEnum() == fb_esp_rtdb_data_type_string) {
        String fbdataString = fbdo.to<String>();
        int fbdata = fbdataString.toInt();
        if (lastOnlineButton != fbdata) {
            lastOnlineButton = fbdata;
            Serial2.print(fbdata);
            Serial.println(fbdata);
        }
    }
    //}
}
}

```



```
else {  
    Serial.println(fbdo.errorReason());  
}  
}  
}
```

