



# **Perancangan dan Implementasi Sistem Kontrol Nutrisi Otomatis pada Greenhouse Menggunakan Algoritma PID Berbasis IoT di Politeknik Negeri Batam**

## **Tugas Akhir**

**Oleh:  
Muhammad Baihaqi Adnan (3232301019)**

**Program Studi Teknik Instrumentasi  
Jurusan Teknik Elektro  
Politeknik Negeri Batam  
2025**

## Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul "**Implementasi Sistem Akuisisi Data Multi-Sensor dan Pengembangan Aplikasi *Mobile Android* untuk *Monitoring* dan Kontrol Otomatis Nutrisi pada *Greenhouse* Berbasis *IoT* di Politeknik Negeri Batam**" adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Muhammad Baihaqi  
Adnan  
NIM: 3232301019

# Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat  
memperoleh gelar  
Ahli Madya Teknik (AMd.T.)  
di  
Politeknik Negeri Batam

Oleh:  
Muhammad Baihaqi Adnan (3232301019)

Tanggal Sidang:  
11 Agustus 2025

Disetujui oleh :

Dosen Penguji 1



1. Muhammad Baihaqi Adnan Wimbang  
Wicaksono, S.T., M.T  
NIK: 122272

Dosen Pembimbing 1



1. Rahmi Mahdaliza S.Si., M.Si.  
NIK: '198712302019032011

Dosen Penguji 2



2. Ahmad Syafi'i, S.Pd., M.T.,  
NIK: 199003232024061001

Dosen Pembimbing 2



2. Ardian Budi Kusuma Atmaja  
S.Tr., M.T.  
NIK: 214172

# **[Implementasi Sistem Akuisisi Data Multi-Sensor dan Pengembangan Aplikasi Mobile Android untuk Monitoring dan Kontrol Otomatis Nutrisi pada Greenhouse Berbasis IoT di Politeknik Negeri Batam]**

## **Abstrak**

Pertanian hidroponik modern menuntut efisiensi dalam pemantauan dan pemberian nutrisi secara *real-time* untuk mendukung produktivitas tanaman. Penelitian ini bertujuan merancang dan mengimplementasikan sistem *monitoring* dan kontrol nutrisi otomatis pada *greenhouse* menggunakan metode *Nutrient Film Technique* (NFT). Sistem ini dirancang dengan mikrokontroler ESP32 yang terhubung dengan sensor suhu dan kelembapan (DHT11), sensor pH (SEN0161), sensor TDS, serta sensor ketinggian air (HC-SR04). Kendali nutrisi dilakukan secara otomatis menggunakan algoritma PID untuk menjaga kestabilan konsentrasi larutan nutrisi. Data sensor dikirim secara *real-time* ke *Firebase Realtime Database* dan ditampilkan melalui aplikasi *Android* dan *website* guna memudahkan pemantauan jarak jauh. Berdasarkan pengujian yang dilakukan selama periode pukul 19.34 hingga 20.24, sistem berhasil mencatat data dari seluruh sensor secara *real-time* dan konsisten, dengan nilai pH berkisar antara 3,1 hingga 4,9 (kondisi asam), nilai TDS relatif stabil di sekitar 1380 ppm mendekati *setpoint*, tinggi air berubah dari 36,1 cm menjadi 49,6 cm, suhu stabil di sekitar 40 °C, dan kelembapan menurun dari 29% menjadi 23%. Hasil ini menunjukkan bahwa sistem mampu menjaga parameter nutrisi tetap stabil dan merespon perubahan nilai TDS secara cepat dengan pengaturan kecepatan pompa yang presisi. Sistem ini juga terbukti dapat menghemat penggunaan air dan nutrisi, meningkatkan efisiensi pemeliharaan tanaman, serta memberikan kemudahan bagi pengguna dalam mengelola kondisi *greenhouse*. Dengan kemampuannya dalam integrasi teknologi *IoT* dan kontrol otomatis, sistem ini berpotensi menjadi solusi pertanian cerdas yang aplikatif untuk kebutuhan edukasi dan penelitian di lingkungan akademik seperti Politeknik Negeri Batam.

Kata kunci: *Greenhouse*, Hidroponik NFT, PID, *IoT*, *Firebase*.

# ***["Implementation of Multi-Sensor Data Acquisition System and Android Mobile Application Development for Automatic Nutrition Monitoring and Control in IoT-Based Greenhouse at Politeknik Negeri Batam"]***

## **Abstract**

*Modern hydroponic farming demands efficiency in real-time monitoring and nutrient delivery to support crop productivity. This study aims to design and implement an automatic nutrient monitoring and control system in a greenhouse using the Nutrient Film Technique (NFT) method. The system is built with an ESP32 microcontroller connected to a temperature and humidity sensor (DHT11), a pH sensor (SEN0161), a TDS sensor, and a water level sensor (HC-SR04). Nutrient control is carried out automatically using a PID algorithm to maintain the stability of the nutrient solution concentration. Sensor data is transmitted in real-time to the Firebase Realtime Database and displayed through an Android application and website to facilitate remote monitoring. Based on testing conducted between 19:34 and 20:24, the system successfully recorded data from all sensors in real-time and consistently, with pH values ranging from 3.1 to 4.9 (acidic condition), TDS values remaining relatively stable at around 1380 ppm close to the setpoint, water level changing from 36.1 cm to 49.6 cm, temperature stable at around 40 °C, and humidity decreasing from 29% to 23%. These results indicate that the system is capable of maintaining nutrient parameters at stable levels and responding quickly to TDS changes with precise pump speed control. The system has also proven to reduce water and nutrient usage, improve plant maintenance efficiency, and provide convenience for users in managing greenhouse conditions. With its capability to integrate IoT technology and automatic control, this system has the potential to serve as a practical smart farming solution for educational and research purposes in academic environments such as Politeknik Negeri Batam.*

**Keywords:** *Greenhouse, NFT Hydroponics, PID, IoT, Firebase*

# Kata Pengantar

Segala puji dan hormat saya panjatkan ke hadirat Tuhan Yesus Kristus, Sang Sumber kehidupan dan segala hikmat, yang telah dengan kasih dan kemurahan-Nya membimbing setiap langkah perjalanan saya hingga akhirnya saya dapat menyelesaikan Proyek Akhir ini dengan baik. Tanpa kasih karunia dan penyertaan-Nya, karya ini tidak akan pernah terselesaikan. Proyek Akhir yang berjudul **"Implementasi Sistem Akuisisi Data Multi-Sensor dan Pengembangan Aplikasi Mobile Android untuk Monitoring dan Kontrol Otomatis Nutrisi pada Greenhouse Berbasis IoT di Politeknik Negeri Batam"** disusun sebagai bagian dari pemenuhan syarat untuk memperoleh gelar Ahli Madya di Jurusan Teknik Elektro, Program Studi Teknik Instrumentasi, Politeknik Negeri Batam.

Penyusunan laporan ini bukan hanya sekadar proses akademik, melainkan sebuah perjalanan panjang yang penuh tantangan, air mata, ketekunan, dan doa.

Saya menyadari sepenuhnya bahwa keberhasilan ini bukanlah hasil usaha pribadi semata, melainkan buah dari doa, dukungan, bimbingan, dan cinta dari banyak hati yang peduli. Dengan segala kerendahan hati, saya ingin menyampaikan rasa terima kasih dan penghargaan yang tulus kepada:

1. **Kedua orang tua dan seluruh keluarga tercinta**, yang telah menjadi tiang kekuatan dan tempat saya bersandar sepanjang proses ini. Terima kasih atas cinta yang tak pernah habis, doa-doa yang senantiasa mengalir, dan dukungan dalam segala bentuk — baik moral, spiritual, hingga materi. Kalian adalah alasan saya tetap kuat, tetap berjuang, dan tidak pernah menyerah.
2. **Ibu Rahmi Mahdaliza, S.Si., M.Si.**, selaku Dosen Pembimbing I yang telah dengan sabar membimbing, mengarahkan, dan memberikan dukungan yang tidak ternilai selama proses penyusunan laporan ini.
3. **Bapak Ardian Budi Kusuma Atmaja, S.Tr., M.T.**, selaku Dosen Pembimbing II yang telah memberikan wawasan teknis dan arahan berharga yang sangat membantu dalam pengembangan proyek ini.
4. **Teman-teman WS 6.1**, yang telah memberikan dukungan moral, motivasi, dan bantuan selama proses pengerjaan Proyek Akhir ini, **serta teman-teman lain yang tidak dapat saya sebutkan satu per satu**, namun kehadiran dan bantuannya sangat berarti bagi Penulis.
5. **Seluruh tim Proyek Akhir "Sistem Monitoring dan Kontrol pada Greenhouse Menggunakan Logika PID di Politeknik Negeri Batam"**,

atas kekompakan, kerja sama yang luar biasa, dan dedikasi dalam menyelesaikan proyek ini secara bersama-sama.

Saya menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, saya membuka hati seluas-luasnya untuk menerima kritik dan saran yang membangun demi perbaikan di masa mendatang.

Akhir kata, saya berharap karya sederhana ini dapat memberi manfaat, menjadi berkat bagi sesama, dan turut berkontribusi dalam kemajuan teknologi pertanian berbasis *IoT* di Indonesia, khususnya di lingkungan akademik.

Batam, 15 Juli 2025

Muhammad Baihaqi Adnan

# Daftar Isi

|                                                   |     |
|---------------------------------------------------|-----|
| Pernyataan Keaslian Tugas Akhir .....             | i   |
| Abstrak.....                                      | iii |
| Abstract.....                                     | iv  |
| Kata Pengantar .....                              | v   |
| Daftar Isi.....                                   | vii |
| Daftar Gambar.....                                | ix  |
| Daftar Tabel .....                                | x   |
| Bab 1. Pendahuluan .....                          | 1   |
| 1.1. Latar Belakang .....                         | 1   |
| 1.2. Rumusan Masalah.....                         | 2   |
| 1.3. Tujuan .....                                 | 2   |
| 1.4. Manfaat .....                                | 2   |
| 1.5. Batasan .....                                | 3   |
| 1.6. <i>Work Breakdown Structure</i> .....        | 3   |
| Bab 2. Tinjauan Pustaka .....                     | 4   |
| 2.1. Landasan Teori .....                         | 4   |
| 2.1.1. Kajian Pustaka .....                       | 4   |
| 2.2. Teknologi yang Digunakan .....               | 5   |
| 2.2.1. Teknologi <i>Hardware</i> .....            | 5   |
| 2.2.2. Teknologi <i>Software</i> .....            | 10  |
| 2.3. Penelitian Terdahulu .....                   | 11  |
| Bab 3. Metode Pelaksanaan .....                   | 17  |
| 3.1. Teknik dan Cara Pelaksanaan.....             | 17  |
| 3.1.1. Perancangan .....                          | 17  |
| 3.1.2. Perancangan <i>Design</i> Elektrikal ..... | 18  |
| 3.1.3. Perancangan <i>Design</i> Mekanikal.....   | 19  |
| 3.1.4. Pembuatan Program.....                     | 20  |
| 3.1.5. Pengembangan Aplikasi <i>Mobile</i> .....  | 20  |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 3.1.5. Pengembangan <i>Website</i> .....                                      | 21 |
| 3.1.6. Pembuatan Firebase Realtime Database.....                              | 22 |
| 3.1.7. Pembuatan Kontrol dengan Metode PID .....                              | 22 |
| 3.2. Alat dan Bahan .....                                                     | 23 |
| 3.3. Pengujian .....                                                          | 24 |
| 3.3.1. Pengujian Sensor pH 0161 .....                                         | 24 |
| 3.3.2. Pengujian Sensor Gravity: TDS Analog .....                             | 24 |
| 3.3.3. Pengujian Sensor Ultrasonic HC-SR04.....                               | 25 |
| 3.3.4. Pengujian Sensor <i>DHT11</i> .....                                    | 25 |
| 3.3.5. Pengujian Peristaltic Pump Dosing Pump 12v DC .....                    | 25 |
| 3.4. Cara Kerja Sistem.....                                                   | 26 |
| Bab 4. Hasil dan Pembahasan .....                                             | 27 |
| 4.1. Data Hasil Penelitian .....                                              | 27 |
| 4.1.1. Data Hasil Pengujian Sensor pH 0161 .....                              | 27 |
| 4.1.2. Data Hasil Pengujian Sensor <i>Gravity: TDS</i> Analog.....            | 29 |
| 4.1.3. Data Hasil Pengujian Sensor Ultrasonic HC-SR04.....                    | 31 |
| 4.1.4. Data Hasil Pengujian Sensor DHT 11.....                                | 32 |
| 4.1.5. Data Hasil Pengujian PID.....                                          | 32 |
| 4.1.6. Pengujian sistem Secara <i>Real-Time</i> pada <i>Greenhouse</i> . .... | 37 |
| Bab 5. Kesimpulan dan Saran.....                                              | 39 |
| 5.1. Kesimpulan.....                                                          | 39 |
| 5.2. Saran.....                                                               | 39 |
| Daftar Pustaka.....                                                           | 41 |
| Biodata.....                                                                  | 43 |
| Lampiran .....                                                                | 44 |
| Lampiran A. Program Sistem.....                                               | 44 |
| Lampiran B. Dokumentasi Pengujian .....                                       | 63 |

## Daftar Gambar

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Gambar 2. 1 ESP 32.....                                              | 5  |
| Gambar 2. 2 <i>Driver</i> Motor L298N.....                           | 6  |
| Gambar 2. 3 Sensor pH 0161 .....                                     | 6  |
| Gambar 2. 4 <i>Gravity</i> : Analog TDS Sensor .....                 | 7  |
| Gambar 2. 5 Sensor <i>Ultrasonic</i> HC-SR04.....                    | 8  |
| Gambar 2. 6 Sensor DHT11.....                                        | 8  |
| Gambar 2. 7 <i>Peristaltic Pump</i> Dosing <i>Pump</i> 12v DC.....   | 9  |
| Gambar 2. 8 LCD 20×4 i2c.....                                        | 9  |
| Gambar 2. 9 <i>Power Supply</i> 12V 5 A.....                         | 9  |
| Gambar 2. 10 <i>Step Down</i> 12v to 5v.....                         | 10 |
| Gambar 3. 1 <i>Flowchart</i> Tahap perancangan sistem.....           | 17 |
| Gambar 3. 2 Koneksi Pin Sensor dan Komponen.....                     | 18 |
| Gambar 3. 3 <i>Design</i> Elektrikal Monitor dan Kontrol.....        | 18 |
| Gambar 3. 4 Design Mekanikal Tandon, Ember Nutrisi & Panel Box ..... | 19 |
| Gambar 3. 5 <i>Mockup</i> Aplikasi Android .....                     | 20 |
| Gambar 3. 6 <i>Mockup Website</i> .....                              | 21 |
| Gambar 3. 7 <i>Firebase Realtime Database</i> .....                  | 22 |
| Gambar 3. 8 Cara Kerja Sistem .....                                  | 26 |
| Gambar 3. 12 Cara Kerja Sistem .....                                 | 26 |
| Gambar 4. 1 Percobaan Pertama PID.....                               | 33 |
| Gambar 4. 2 Percobaan Kedua PID .....                                | 34 |
| Gambar 4. 3 Percobaan Ketiga PID.....                                | 35 |
| Gambar 4. 4 Percobaan Keempat PID .....                              | 36 |
| Gambar 4. 5 Percobaan Kelima PID.....                                | 37 |

## Daftar Tabel

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| Tabel 1. 1 <i>Work Breakdown Structure</i> .....                                   | 3  |
| Tabel 2. 1 Penelitian Terdahulu .....                                              | 11 |
| Tabel 2. 1 Penelitian Terdahulu .....                                              | 11 |
| Tabel 3. 1 Alat dan Bahan.....                                                     | 23 |
| Tabel 3. 2 Pengujian Sensor pH 0161 .....                                          | 24 |
| Tabel 3. 3 Pengujian Sensor Gravity: TDS Analog .....                              | 25 |
| Tabel 4. 1 Tegangan Sensor Ph Sebelum Linierisasi.....                             | 27 |
| Tabel 4. 2 Sensor Ph Sebelum Linierisasi .....                                     | 27 |
| Tabel 4. 3 Sensor Ph Setelah Linierisasi .....                                     | 28 |
| Tabel 4. 4 Data Pengujian Sensor Gravity: TDS Analog sebelum Linierisasi.....      | 29 |
| Tabel 4. 5 Data Pengujian Tegangan: TDS Analog sebelum Linierisasi .....           | 29 |
| Tabel 4. 6 Data Hasil Pengujian Sensor Gravity: TDS Analog Setelah Linierisasi ... | 30 |
| Tabel 4. 7 Data Hasil Pengujian Sensor Ultrasonic HC-SR04 Sebelum Linierisasi .    | 31 |
| Tabel 4. 8 Data Hasil Pengujian Sensor Ultrasonic HC-SR04 Setelah Linierisasi .... | 31 |
| Tabel 4. 9 Data Hasil Pengujian Sensor DHT 11.....                                 | 32 |
| Tabel 4. 10 Analisis Percobaan Pertama Pid .....                                   | 33 |
| Tabel 4. 11 Analisis Percobaan Kedua Pid.....                                      | 34 |
| Tabel 4. 12 Analisis Percobaan Ketiga Pid.....                                     | 35 |
| Tabel 4. 13 Analisis Percobaan Keempat Pid.....                                    | 36 |
| Tabel 4. 14 Analisis Percobaan Kelima Pid.....                                     | 37 |
| Tabel 4. 15 Pengujian sistem secara iReal-Time.....                                | 37 |

# Bab 1. Pendahuluan

## 1.1. Latar Belakang

Pertanian tradisional sering kali membutuhkan waktu dan tenaga yang besar untuk pemeliharaan, pengelolaan, dan pemantauan tanaman. Keterbatasan waktu dan tenaga kerja dapat mengakibatkan keterlambatan dalam penanganan hama, penyakit, atau kebutuhan nutrisi tanaman. Teknologi canggih seperti sistem pemantauan otomatis dapat meningkatkan efisiensi, namun menjadikannya memerlukan investasi dan pelatihan yang signifikan [1]. Tanpa teknologi yang tepat, pengguna greenhouse menghadapi tantangan dalam mengelola waktu dan tenaga secara efektif.

Perkembangan IPTEK (Ilmu Pengetahuan dan Teknologi) pada era modern saat ini berkembang dengan pesat, termasuk pada bidang pertanian. Salah satu teknik budidaya tanaman yang semakin populer adalah teknik hidroponik. Hidroponik merupakan metode budidaya tanaman dengan menggunakan air yang telah dilarutkan nutrisi sebagai media tumbuh, menggantikan penggunaan tanah. Teknik ini memungkinkan pemanfaatan sumber daya alam hayati untuk menghasilkan bahan pangan, bahan industri, sumber energi, serta menjaga kelestarian lingkungan [2].

Sistem hidroponik Nutrient Film Technique (NFT) adalah salah satu metode budidaya tanaman tanpa tanah yang menggunakan aliran tipis larutan nutrisi untuk menyediakan unsur hara kepada akar tanaman. Dalam metode ini, larutan nutrisi secara terus-menerus dialirkan dalam lapisan tipis di atas akar tanaman yang ditempatkan dalam saluran atau kanal. Sistem NFT dikenal karena efisiensinya dalam penggunaan air dan nutrisi, serta kemampuannya untuk mendukung pertumbuhan tanaman dengan cepat. Penelitian tersebut menunjukkan bahwa sistem NFT memberikan kontrol yang baik atas kondisi nutrisi dan oksigen yang diterima oleh tanaman, sehingga dapat meningkatkan hasil panen dan mengurangi risiko penyakit tanah [3].

Pengendalian sistem hidroponik telah dilakukan oleh peneliti terdahulu dengan berbagai metode. Perancangan sistem pemantauan dan pengendalian nutrisi pada tanaman hidroponik sistem NFT menggunakan kontrol PID (Proportional-Integral-Derivative) telah terbukti efektif dalam menjaga stabilitas sistem [4]. Penelitian yang dilakukan oleh mahasiswa Politeknik Negeri Batam juga menunjukkan sistem monitoring pada greenhouse di Politeknik Negeri Batam memberikan hasil yang baik dalam pemantauan kondisi tanaman.

Berdasarkan kajian penelitian terdahulu, maka pada penelitian ini dirancang sistem monitoring dan kontrol nutrisi pada sistem hidroponik menggunakan kendali PID dengan integrasi aplikasi Android dan *website*. Keuntungan menggunakan metode PID memberikan respons cepat terhadap perubahan mendadak dalam kebutuhan nutrisi dan menjaga kestabilan sistem [5]. Dengan

penambahan monitoring melalui aplikasi *mobile* dan *web*, sistem ini berpotensi meningkatkan produktivitas dan efisiensi dalam pengelolaan *greenhouse* di Politeknik Negeri Batam.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat diidentifikasi beberapa permasalahan yang perlu diatasi dalam pengembangan sistem monitoring dan kontrol *greenhouse* hidroponik, yaitu:

1. Bagaimana membangun sistem akuisisi data multi-sensor untuk mengumpulkan data parameter lingkungan *greenhouse* secara *real-time*?
2. Bagaimana menerapkan metode kontrol PID untuk mengontrol sistem nutrisi *greenhouse* secara otomatis?
3. Bagaimana mengembangkan aplikasi *mobile* Android serta *Website* untuk *monitoring* dan kontrol *greenhouse* yang dapat diakses dari jarak jauh?
4. Bagaimana mengintegrasikan sistem sensor, kontrol PID, dan aplikasi *mobile* menggunakan teknologi IoT?
5. Bagaimana menguji kinerja sistem yang telah dibangun dalam mengontrol parameter nutrisi pada *greenhouse* hidroponik?

## 1.3. Tujuan

Berdasarkan rumusan masalah yang telah diidentifikasi, maka tujuan dari penelitian ini adalah:

1. Melakukan akuisisi data multi-sensor untuk mengumpulkan data parameter lingkungan *greenhouse* secara *real-time*.
2. Menerapkan metode kontrol PID untuk mengontrol sistem nutrisi *greenhouse* secara otomatis.
3. Mengembangkan aplikasi *mobile* Android serta *Website* untuk *monitoring* dan kontrol *greenhouse* yang dapat diakses dari jarak jauh.
4. Mengintegrasikan sistem sensor, kontrol PID, dan aplikasi *mobile* menggunakan teknologi IoT.
5. Menguji kinerja sistem yang telah dibangun dalam mengontrol parameter nutrisi pada *greenhouse* hidroponik.

## 1.4. Manfaat

Adapun manfaat yang diharapkan dari kegiatan ini adalah sebagai berikut:

1. Memberikan kemudahan dalam memantau kondisi suhu, kelembapan, dan kadar nutrisi tanaman secara *real time* melalui *smartphone* dan *website*.

2. Memudahkan pengguna dalam mengatur pemberian nutrisi tanaman secara otomatis dan sesuai kebutuhan melalui sistem kontrol berbasis PID.
3. Meningkatkan pertumbuhan dan kualitas tanaman dengan pemberian nutrisi yang lebih tepat dan terukur.
4. Menyediakan alat peraga berupa *prototipe* sistem *greenhouse* otomatis untuk kebutuhan praktikum dan penelitian di Politeknik Negeri Batam.
5. Mendorong penerapan teknologi *IoT* dan *smart farming* di kalangan mahasiswa dan masyarakat umum sebagai solusi pertanian modern yang efisien dan berkelanjutan.

## 1.5. Batasan

Untuk menjaga fokus dan ruang lingkup penelitian, batasan-batasan berikut ditetapkan:

1. Sistem dirancang khusus untuk *greenhouse* hidroponik dengan metode NFT (Nutrient Film Technique).
2. Parameter yang dimonitor terbatas pada suhu, kelembaban, pH, Ketinggian Air pada Tandon dan kadar nutrisi (TDS/EC) menggunakan empat jenis sensor.
3. Kontrol otomatis hanya diterapkan pada pemberian nutrisi, dengan metode kendali PID untuk menjaga kestabilan larutan nutrisi.
4. Sistem menggunakan mikrokontroler ESP32 sebagai *platform* utama untuk pemrosesan data dan konektivitas *IoT*.
5. Pengujian dilakukan di *greenhouse* Politeknik Negeri Batam, dalam kondisi lingkungan normal dan tidak mencakup pengujian pada kondisi ekstrem.
6. Evaluasi performa sistem terbatas pada parameter yang dapat diukur oleh sensor, dan sistem dirancang beroperasi optimal pada suhu 20–35°C dan kelembaban 60–90%.

## 1.6. Work Breakdown Structure

**Tabel 1. 1 Work Breakdown Structure**

| No | Nama                   | Tugas dan Tanggung Jawab dalam Tim                           |
|----|------------------------|--------------------------------------------------------------|
| 1  | Jelita Agnesia Purba   | Koordinator, Akuisisi Data dan Pengembangan Aplikasi Android |
| 2  | Muhammad Reza          | Pengembangan <i>Website</i>                                  |
| 3  | Muhammad Baihaqi Adnan | Kontrol Pid dan Rancang Bangun                               |

## Bab 2. Tinjauan Pustaka

### 2.1. Landasan Teori

#### 2.1.1. Kajian Pustaka

Sistem hidroponik adalah metode budidaya tanaman tanpa menggunakan tanah sebagai media tanam, melainkan menggunakan larutan nutrisi yang mengandung unsur hara yang dibutuhkan tanaman. Menurut [3] Sistem ini bekerja dengan prinsip menyediakan nutrisi melalui larutan yang mengandung unsur hara makro dan mikro. Unsur hara makro yang dibutuhkan tanaman meliputi nitrogen (N), fosfor (P), kalium (K), kalsium (Ca), magnesium (Mg), dan sulfur (S), sedangkan unsur hara mikro meliputi besi (Fe), mangan (Mn), seng (Zn), tembaga (Cu), boron (B), dan molibdenum (Mo).

Salah satu teknik hidroponik yang paling populer adalah Nutrient Film Technique (NFT), yaitu sistem di mana larutan nutrisi dialirkan dalam bentuk lapisan tipis melalui saluran yang berisi akar tanaman. Menurut [6], sistem NFT memiliki karakteristik aliran nutrisi yang kontinu dengan ketebalan 2–3 mm, akar tanaman sebagian tercelup dalam larutan, oksigenasi akar yang baik, serta efisiensi penggunaan air dan nutrisi yang tinggi. Komponen utama NFT meliputi reservoir nutrisi, pompa sirkulasi, saluran tumbuh, sistem drainase, dan sistem aerasi, dengan kemiringan saluran sekitar 1:30 hingga 1:40 untuk menjaga aliran nutrisi tetap optimal.

Dalam pengaturan sistem ini, kontrol PID (Proportional-Integral-Derivative) sering digunakan untuk mengatur parameter seperti pH, konsentrasi nutrisi (EC/TDS), suhu larutan, dan kestabilan aliran nutrisi. Menurut [7], kontrol PID menggunakan tiga parameter, yaitu proporsional ( $K_p$ ), integral ( $K_i$ ), dan derivatif ( $K_d$ ), dengan fungsi transfer  $G(s) = K_p + K_i/s + K_d.s$ . Kontrol ini memberikan respon yang cepat, mengurangi *error steady-state*, dan menjaga kestabilan sistem.

Untuk meningkatkan efisiensi dan kemudahan pengawasan, teknologi Internet of Things (IoT) juga diterapkan dalam sistem pertanian modern. Menurut [8], IoT memungkinkan pemantauan dan pengendalian kondisi pertanian secara real-time melalui internet. Komponen utama *IoT* dalam pertanian meliputi sensor, mikrokontroler, komunikasi nirkabel, *platform cloud*, dan aplikasi *mobile* atau *web* untuk *monitoring* dan kontrol. Implementasi *IoT* dalam pertanian terbukti mampu meningkatkan produktivitas hingga 30% dan mengurangi penggunaan air hingga 50%, menjadikannya solusi inovatif dalam pengembangan sistem hidroponik modern.

## 2.2. Teknologi yang Digunakan

### 2.2.1. Teknologi *Hardware*

#### 2.2.1.1. ESP 32

Pada Gambar 2.1 Pada Gambar 2.1 merupakan mikrokontroler yang digunakan sebagai komunikasi yaitu ESP 32. ESP 32 adalah komunikasi nirkabel (*wireless*) yang memberikan informasi jarak jauh tanpa menggunakan tegangan listrik atau kawat dimana komunikasi nirkabel menggunakan media udara sebagai tempat transmisinya. Mikrokontroler ESP32 ini dapat diprogram dengan menggunakan C++, C, Python, Lua, dll. untuk menjalankan program mikrokontroler ESP32 ini memerlukan suatu *software* pemrograman.



**Gambar 2. 1 ESP 32**

#### 2.2.1.2. *Driver Motor* L298N

Pada Gambar 2.2 *Driver Motor* L298N merupakan modul pengendali motor DC yang digunakan untuk mengatur arah putaran dan kecepatan motor, termasuk pada pompa nutrisi dalam sistem otomasi. Modul ini bekerja sebagai jembatan antara mikrokontroler, seperti ESP32, dan motor DC, dengan menerima sinyal logika dari mikrokontroler untuk mengaktifkan atau menonaktifkan pompa secara otomatis. L298N memiliki dua *channel* yang masing-masing dapat mengendalikan satu motor DC, mendukung tegangan kerja hingga 35V, dan arus maksimum sekitar 2A per *channel*. Modul ini dilengkapi dengan rangkaian proteksi serta terminal koneksi yang memudahkan pemasangan, sehingga banyak digunakan pada berbagai proyek otomasi. Dalam aplikasi sistem kontrol nutrisi, L298N memastikan pompa dapat beroperasi sesuai perintah sistem, baik dalam mode manual maupun otomatis, untuk menjaga suplai nutrisi tetap optimal.



**Gambar 2. 2 Driver Motor L298N**

#### **2.2.1.3. Sensor pH 0161**

Pada Gambar 2.3 Sensor pH kit SEN0161 merupakan modul pengukur derajat keasaman atau kebasaan larutan yang banyak digunakan pada penelitian dan praktikum laboratorium. Kit ini terdiri dari *probe* pH berbahan kaca dengan konektor BNC dan papan pengkondisi sinyal yang mengubah sinyal elektrokimia menjadi tegangan analog yang dapat dibaca mikrokontroler. Prinsip kerjanya didasarkan pada reaksi membran kaca terhadap konsentrasi ion hidrogen ( $H^+$ ) di dalam larutan, yang menghasilkan beda potensial sesuai Persamaan Nernst, dengan tegangan keluaran sekitar 2,5 V pada pH netral (pH 7) dan berubah secara proporsional terhadap nilai pH. Sensor ini direkomendasikan hanya untuk lingkungan laboratorium karena sensitif terhadap suhu ekstrem, cairan korosif, dan kondisi lapangan yang tidak terkontrol. Untuk memperoleh akurasi tinggi, diperlukan proses kalibrasi menggunakan larutan buffer standar pH 4, 7, dan 9, serta perawatan rutin seperti pembilasan dengan air bebas ion dan penyimpanan ujung probe dalam larutan penyimpanan khusus. Pada sistem monitoring berbasis mikrokontroler, sensor ini dihubungkan melalui pin *VCC*, *GND*, dan *Aout*, dengan pembacaan data dilakukan melalui pin analog serta pengolahan sinyal untuk mengurangi *noise* dan mengonversi tegangan menjadi nilai pH yang terkalibrasi.



**Gambar 2. 3 Sensor pH 0161**

#### **2.2.1.4. Gravity : Analog TDS Sensor**

Pada Gambar 2.4 Sensor Gravity: Analog TDS merupakan modul yang digunakan untuk mengukur jumlah total zat padat terlarut (*Total Dissolved Solids/TDS*) dalam air atau larutan, yang biasanya dinyatakan dalam satuan ppm

(*parts per million*). Pengukuran TDS penting untuk memantau kualitas air, terutama pada sistem hidroponik, karena nilai TDS yang terlalu rendah menunjukkan kurangnya nutrisi, sedangkan nilai yang terlalu tinggi dapat berdampak buruk bagi pertumbuhan tanaman. Sensor ini bekerja dengan mengukur konduktivitas listrik larutan, di mana semakin tinggi konduktivitas, semakin besar kandungan zat terlarut. Modul ini terdiri dari *probe* konduktivitas dan papan pengkondisi sinyal yang menghasilkan keluaran tegangan analog yang sebanding dengan nilai TDS. Tegangan ini kemudian dibaca oleh pin analog pada mikrokontroler seperti Arduino atau ESP32, dan melalui proses konversi dengan rumus kalibrasi, nilai tegangan dapat diubah menjadi nilai TDS dalam ppm. Sensor ini umumnya digunakan pada suhu cairan yang terkontrol dan memerlukan kalibrasi untuk menjaga akurasi pengukuran.



**Gambar 2. 4 Gravity : Analog TDS Sensor**

#### **2.2.1.5. Sensor Ultrasonic HC-SR04**

Pada Gambar 2.5 Sensor Ultrasonic HC-SR04 merupakan sensor jarak yang digunakan untuk mengukur ketinggian atau permukaan air pada tandon dengan memanfaatkan prinsip pantulan gelombang ultrasonik. Sensor ini memiliki dua komponen utama, yaitu pemancar (*trigger*) yang mengirimkan gelombang ultrasonik dan penerima (*echo*) yang menangkap gelombang pantul dari permukaan objek, dalam hal ini permukaan air. Waktu tempuh gelombang dari pemancar ke permukaan air dan kembali ke penerima diukur, kemudian dikonversi menjadi jarak menggunakan kecepatan rambat suara di udara. HC-SR04 menghasilkan sinyal digital yang dapat dibaca oleh mikrokontroler seperti Arduino atau ESP32 untuk kemudian diolah menjadi informasi ketinggian air. Sensor ini memiliki jangkauan pengukuran sekitar 2 cm hingga 400 cm dengan tingkat akurasi yang baik, sehingga efektif digunakan pada sistem monitoring ketinggian air secara *real-time*.



**Gambar 2. 5 Sensor *Ultrasonic* HC-SR04**

#### **2.2.1.6. Sensor DHT11**

Pada Gambar 2.6 Sensor DHT11 merupakan modul sensor digital yang berfungsi untuk mengukur suhu dan kelembaban udara di lingkungan, seperti pada *greenhouse*. Di dalamnya terdapat dua elemen pengukuran, yaitu termistor yang mendeteksi suhu dan sensor kelembaban kapasitif yang mengukur kadar uap air di udara. Kedua hasil pengukuran tersebut diproses oleh chip internal dan dikirimkan dalam bentuk sinyal digital melalui satu jalur data, sehingga memudahkan integrasi dengan mikrokontroler seperti Arduino atau ESP32. Sensor ini memiliki rentang pengukuran suhu 0–50 °C dengan akurasi sekitar  $\pm 2$  °C, serta kelembaban 20–90 % RH dengan akurasi  $\pm 5$  % RH. Meskipun memiliki resolusi dan akurasi yang tergolong standar, DHT11 cukup andal untuk pemantauan kondisi lingkungan secara *real-time* dalam sistem otomatisasi *greenhouse* guna menjaga suhu dan kelembaban sesuai kebutuhan tanaman.



**Gambar 2. 6 Sensor DHT11**

#### **2.2.1.7. *Peristaltic Pump Dosing Pump 12v DC***

Pada Gambar 2.7 *Peristaltic Pump 12V DC* merupakan jenis pompa yang digunakan untuk mengalirkan larutan nutrisi ke dalam tandon utama dengan tingkat presisi yang tinggi. Prinsip kerjanya didasarkan pada mekanisme peristaltik, di mana serangkaian *roller* atau roda penekan secara bergantian menekan dan melepas selang fleksibel, sehingga mendorong cairan bergerak maju tanpa kontak langsung antara komponen pompa dan cairan yang dipompa. Desain ini membuat pompa aman digunakan untuk cairan nutrisi karena mencegah risiko kontaminasi dan memudahkan proses pembersihan. Pompa ini memiliki tegangan kerja 12V DC dan dapat diintegrasikan dengan sistem otomatisasi berbasis mikrokontroler, sehingga aliran nutrisi dapat dikontrol sesuai kebutuhan. Keunggulan lainnya adalah kemampuannya untuk mengatur laju aliran secara

stabil, menjadikannya ideal untuk aplikasi hidroponik dan sistem pemberian nutrisi yang memerlukan akurasi tinggi.



**Gambar 2. 7 Peristaltic Pump Dosing Pump 12v DC**

#### **2.2.1.8. LCD 20×4 i2c**

Pada Gambar 2.8 ditunjukkan LCD 20×4 I2C, yaitu modul tampilan yang digunakan untuk menampilkan data hasil pembacaan sensor secara langsung. LCD ini mampu menampilkan hingga 4 baris dengan 20 karakter per baris, dan menggunakan komunikasi I2C sehingga hanya memerlukan dua pin pada mikrokontroler seperti Arduino atau ESP32. Modul ini memudahkan pemantauan data suhu, kelembaban, pH, Ketinggian Air pada Tandon dan Nutrisi secara *real time* di lokasi *greenhouse*.



**Gambar 2. 8 LCD 20×4 i2c**

#### **2.2.1.9. Power Supply 12V 5 A**

Pada Gambar 2.9 ditunjukkan *Power Supply* 12V 5A, yaitu perangkat yang digunakan untuk menyuplai tegangan ke seluruh sistem, termasuk mikrokontroler dan pompa nutrisi. *Power supply* ini memberikan arus yang stabil dengan tegangan 12 volt dan arus maksimal 5 ampere, sehingga mampu mendukung kinerja komponen elektronik dan aktuator dalam sistem secara optimal.



**Gambar 2. 9 Power Supply 12V 5 A**

#### **2.2.1.10. Step Down 12v to 5v**

Pada Gambar 2.10 ditunjukkan *Step Down* 12V to 5V, yaitu modul yang digunakan untuk mengonversi tegangan 12 volt menjadi 5 volt sebelum masuk ke

mikrokontroler. Modul ini berfungsi untuk menjaga agar tegangan yang diterima oleh ESP32 tetap stabil dan sesuai batas kerja, sehingga mencegah kerusakan pada sistem.



**Gambar 2. 10 Step Down 12v to 5v**

## **2.2.2. Teknologi Software**

### **2.2.2.1. Visual Studios Code**

Visual Studio Code (VS Code) adalah editor kode sumber (*source code editor*) yang populer, dikembangkan oleh Microsoft. VS Code mendukung berbagai bahasa pemrograman seperti Python, JavaScript, C++, Java, HTML, CSS, dan banyak lagi. Editor ini bersifat gratis dan *open-source*, serta bisa berjalan di berbagai sistem operasi seperti Windows, macOS, dan Linux. *Visual Studios Code* ini digunakan untuk membuat tampilan *web* dan *mobile*.

### **2.2.2.2. Firebase Realtime Database**

*Firebase Realtime Database* yaitu *platform* berbasis *cloud* yang digunakan untuk menyimpan dan menampilkan data sensor secara *online*. *Firebase* memungkinkan sistem mengirim dan menerima data secara *real time*, sehingga pengguna dapat memantau kondisi *greenhouse* dari jarak jauh melalui aplikasi *mobile* atau *website*. *Platform* ini sangat cocok untuk sistem berbasis *IoT* karena mendukung komunikasi cepat dan efisien.

### **2.2.2.3. Android Studio**

Android Studio yaitu *software* yang digunakan untuk membuat aplikasi *mobile* berbasis Android. Aplikasi ini digunakan sebagai antarmuka (*interface*) untuk menampilkan data sensor secara *real time* yang tersimpan di *Firebase*. Dengan Android Studio, pengguna dapat memantau kondisi *greenhouse* melalui *smartphone*, sehingga pengawasan dapat dilakukan dari jarak jauh dengan lebih praktis dan efisien.

### **2.2.2.4. SolidWorks**

*SolidWorks* yaitu perangkat lunak yang digunakan untuk merancang desain mekanikal dari sistem *greenhouse*. *SolidWorks* membantu dalam membuat model 3D dari komponen seperti rangka *greenhouse*,udukan sensor, dan posisi pompa secara presisi. Dengan desain ini, proses perakitan sistem menjadi lebih terstruktur dan efisien.

### 2.2.2.5. EasyEda

*EasyEDA* yaitu perangkat lunak berbasis *web* yang digunakan untuk merancang rangkaian elektrik dari sistem. *EasyEDA* memudahkan dalam pembuatan skematik dan *layout PCB* secara digital, sehingga rangkaian sistem monitoring dan kontrol dapat dibuat dengan rapi, presisi, dan siap untuk proses perakitan atau produksi.

### 2.2.2.6. Arduino IDE

Arduino IDE yaitu *software* yang digunakan untuk memprogram mikrokontroler seperti ESP32. Arduino IDE menggunakan bahasa pemrograman C/C++ dan menyediakan antarmuka yang sederhana untuk menulis, mengunggah, dan menguji kode program. Dalam sistem ini, Arduino IDE digunakan untuk mengatur logika pembacaan sensor, kendali pompa, dan pengiriman data ke *Firebase*.

### 2.2.2.7. GitHub

GitHub yaitu *platform* penyimpanan berbasis *cloud* yang digunakan untuk menyimpan seluruh program dari sistem sebagai arsip. GitHub memudahkan pengelolaan versi kode, kolaborasi tim, serta dokumentasi program secara online. Dengan GitHub, semua file proyek dapat diakses kembali dengan mudah dan aman kapan saja.

## 2.3. Penelitian Terdahulu

Tabel 2. 1 Penelitian Terdahulu

| No . | Nama Peneliti                                 | Judul                                                                                          | Persamaan                                                                                                                                                                                                                                                                                                  | Perbedaan                                                                                                                                                                                              |
|------|-----------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.   | Dhoni Setyanto dan Nur Sultan Salahuddin. [9] | <i>Prototipe Monitor dan Kontrol Otomatis Iklim Mikro Greenhouse dengan Platform IoT Blynk</i> | sama-sama menggunakan teknologi <i>IoT</i> untuk memantau kondisi <i>greenhouse</i> secara <i>real-time</i> . Keduanya menggunakan sensor DHT11 untuk suhu dan kelembaban udara, serta sensor HC-SR04 untuk mendeteksi ketinggian air. Mikrokontroler yang digunakan juga mendukung Wi-Fi, serta sama-sama | Perbedaannya , sistem kami dirancang khusus untuk <i>greenhouse</i> hidroponik metode NFT, sedangkan penelitian referensi menggunakan media tanah. Sistem kamu menggunakan kontrol PID untuk pemberian |

|    |                                               |                                                        |                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----|-----------------------------------------------|--------------------------------------------------------|------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                               |                                                        | <p>diprogram menggunakan Arduino IDE dan diakses melalui aplikasi Android.</p>     | <p>nutrisi dan TDS sensor, sementara referensi hanya pakai sistem <i>on/off</i> dan <i>soil moisture</i> sensor. Platform <i>IoT</i> yang digunakan juga berbeda—kamu memakai <i>Firebase</i> dan aplikasi Android buatan sendiri, mereka menggunakan Blynk. Dari sisi desain, sistem kamu juga dibuat dengan bantuan SolidWorks dan <i>EasyEDA</i>, serta seluruh kode disimpan di GitHub untuk dokumentasi proyek.</p> |
| 2. | Hamzah Nashirudin & Triyanto Pangaribowo [10] | <i>Prototype Automasi Greenhouse Dengan Monitoring</i> | sama-sama menggunakan mikrokontroler ESP32 sebagai otak sistem, serta memanfaatkan | Penelitian kami menggunakan sistem hidroponik                                                                                                                                                                                                                                                                                                                                                                            |

|  |  |                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--|--|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  | <p>Menggunakan <i>Smartphone</i> dan <i>ESP32</i> Berbasis <i>Internet of Things</i></p> | <p>teknologi <i>IoT</i> untuk <i>monitoring</i> kondisi <i>greenhouse</i> secara <i>real-time</i>. Kedua sistem juga menggunakan sensor suhu dan kelembaban (<i>DHT series</i>), serta dilengkapi aplikasi <i>mobile</i> untuk pemantauan dari jarak jauh. Sistem pada penelitian tersebut maupun sistemmu sama-sama mendukung mode otomatis dan manual, dan mengatur kontrol penyiraman tanaman secara otomatis sesuai parameter lingkungan. Keduanya juga menggunakan kontrol <i>PID</i> untuk menjaga kestabilan suhu di dalam <i>greenhouse</i>.</p> | <p>dengan metode <i>NFT</i>, sedangkan penelitian referensi menggunakan media tanah. Kami memanfaatkan <i>TDS</i> sensor untuk mengukur kadar nutrisi, sementara mereka menggunakan sensor kelembaban tanah. Platform <i>IoT</i> yang digunakan juga berbeda, penelitian kami memakai <i>Firebase Realtime Database</i> dan aplikasi <i>Android</i> buatan sendiri, sedangkan mereka menggunakan <i>Blynk</i>. Selain itu, aktuator pada sistem kami menggunakan dosing <i>pump</i> dengan</p> |
|--|--|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|    |                                                                     |                                                                                     |                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                    |
|----|---------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                                     |                                                                                     |                                                                                                                                                                        | kontrol PID untuk pengaturan nutrisi, sementara sistem mereka menggunakan pompa <i>sprayer</i> dan kipas untuk penyiraman dan pendingin. Desain perangkat keras pada penelitian kami dibuat dengan bantuan <i>SolidWorks</i> dan <i>EasyEDA</i> , serta seluruh kode disimpan di GitHub untuk dokumentasi proyek, yang tidak dijelaskan pada penelitian referensi. |
| 3. | Adevindi Dwi Agustya, Rini Handayani, dan Muhammad Ikhsan Sani [11] | Sistem Kendali dan <i>Monitoring</i> Lingkungan Rumah Kaca dengan Metode <i>IoT</i> | Penelitian kami dan jurnal referensi sama-sama merancang sistem otomasi dan <i>monitoring greenhouse</i> berbasis <i>IoT</i> . Keduanya menggunakan sensor DHT11 untuk | Penelitian kami dirancang untuk sistem <i>greenhouse</i> hidroponik metode <i>NFT</i> , sedangkan referensi                                                                                                                                                                                                                                                        |

|  |  |  |                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  |  | <p>membaca suhu dan kelembaban udara serta mengimplementasikan kontrol PID untuk mengatur parameter lingkungan. Data dari sensor dikirim ke <i>platform cloud</i> dan dapat dipantau secara <i>real-time</i> melalui <i>smartphone</i>.</p> | <p>menggunakan media tanah. Mikrokontroler yang digunakan berbeda, kami memakai ESP32, sementara mereka menggunakan NodeMCU (ESP8266). Sistem kami menggunakan <i>Firebase Realtime Database</i> dan aplikasi Android buatan sendiri, sedangkan mereka menggunakan <i>Thingspeak</i> dan <i>interface web</i>. Selain itu, kami menggunakan TDS sensor dan dosing <i>pump</i> untuk kontrol nutrisi cair, sedangkan referensi tidak mencakup parameter nutrisi. Kami juga melibatkan</p> |
|--|--|--|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

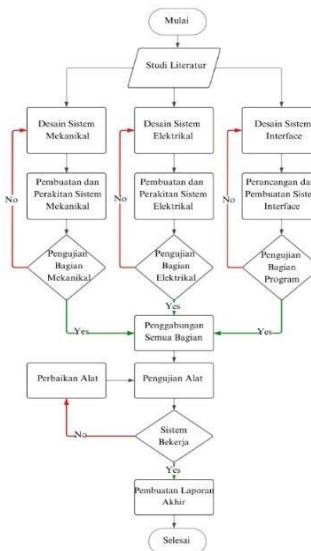
|  |  |  |  |                                                                                                                                                                    |
|--|--|--|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  |  |  | <p>perancangan mekanikal (<i>SolidWorks</i>) dan elektrikal (<i>EasyEDA</i>), serta dokumentasi melalui GitHub, yang tidak dijelaskan dalam penelitian mereka.</p> |
|--|--|--|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Bab 3. Metode Pelaksanaan

### 3.1. Teknik dan Cara Pelaksanaan

#### 3.1.1. Perancangan

Pada Gambar 3.1 Tahap perancangan sistem dapat dilihat, yang menjelaskan secara umum urutan pelaksanaan yang akan dilakukan. Tahap perencanaan dimulai dari persiapan untuk menyusun kerangka penelitian yang sistematis, menentukan daerah sasaran, dan melakukan observasi lapangan. Setelah itu, sosialisasi terhadap pemberian pakan ikan secara otomatis dan perancangan fabiasi setelah itu melakukan pengecekan alat setelah selesai pengecekan selanjutnya tahap terakhir yaitu pembuatan laporan. menyertakan data-data yang sudah didapatkan.



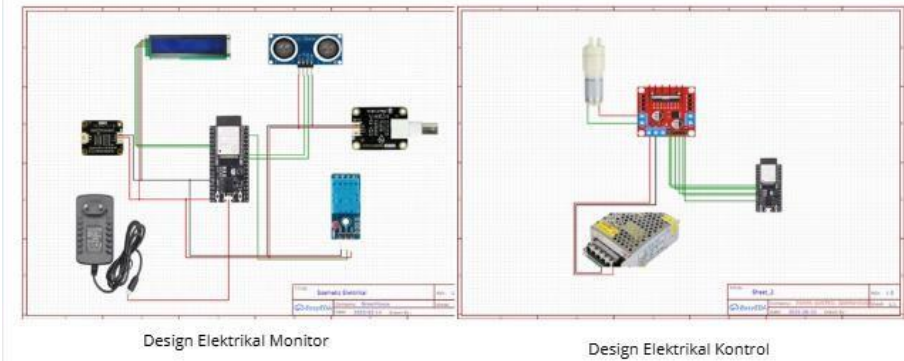
Gambar 3. 1 *Flowchart* Tahap perancangan sistem

3.1.2. Perancangan *Design* Elektrikal

Gambar 3.2, 3.3, menunjukkan proses perancangan elektrikal dari sistem monitoring dan kontrol nutrisi otomatis pada *greenhouse* di Politeknik Negeri Batam. Perancangan ini dilengkapi dengan panduan wiring antar komponen untuk memastikan koneksi yang tepat dan kestabilan sistem. Pemilihan pin yang sesuai menjadi bagian penting agar setiap komponen dapat bekerja dengan optimal. Dalam proses ini, referensi *wiring* diperlukan agar pengkabelan dilakukan dengan benar dan sesuai dengan kebutuhan sistem. Desain dibuat menggunakan *software* draw.io yang memvisualisasikan hubungan antar komponen seperti sensor DHT11, sensor pH SEN0161, sensor TDS SEN0244, sensor ultrasonik HC-SR04, ESP32, motor servo, dan kabel jumper. Semua komponen dihubungkan langsung ke mikrokontroler ESP32 dengan penempatan pin yang telah disesuaikan agar pembacaan data dan proses kontrol berjalan maksimal. Adapun tabel pin yang digunakan dalam desain elektrikal ini disajikan berikutnya sebagai acuan utama dalam proses perakitan sistem.

| PIN yang terhubung | Komponen  | PIN yang terhubung | PIN yang terhubung | Komponen     | PIN yang terhubung |
|--------------------|-----------|--------------------|--------------------|--------------|--------------------|
| VCC                | DHT11     | 3.3V               | VCC                | ULTRA SONIK  | VIN                |
| GND                |           | GND                | GND                |              | GND                |
| DATA               | pH        | D4 (GPIO4)         | TRIGG              |              | D6 (GPIO6)         |
| VCC                |           | 3.3V               | ECHO               |              | D18 (GPIO18)       |
| GND                |           | GND                | VCC                | TDS          | 3.3V               |
| ANALOG             |           | D34 (GPIO34)       | GND                |              | GND                |
| VCC                | LCD       | VIN                | ANALOG             |              | D35 (GPIO35)       |
| GND                |           | GND                | PIN 1              | DRIVER MOTOR | D25                |
| SDA                |           | D21 (GPIO21)       | PIN 2              |              | D26                |
| SCL                |           | D22 (GPIO22)       | PIN 3              |              | D27                |
| GND                |           | GND                | GND                |              | GND                |
| VIN                | STEPPDOWN | VIN                |                    |              |                    |

Gambar 3. 2 Koneksi Pin Sensor dan Komponen

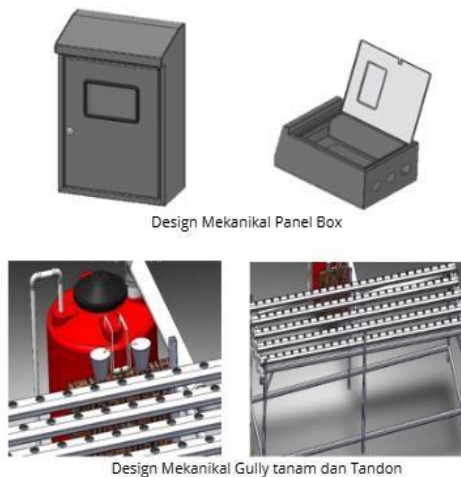


Gambar 3. 3 *Design* Elektrikal Monitor dan Kontrol

### 3.1.3. Perancangan *Design* Mekanikal

Gambar 3.4 menampilkan desain mekanikal sistem hidroponik di *greenhouse* Politeknik Negeri Batam yang terdiri dari panel *box*, tandon nutrisi, ember nutrisi, dan gully tanam. Panel *box* berukuran 30 × 20 × 17 cm digunakan untuk menempatkan komponen elektronik seperti sensor, mikrokontroler ESP32, modul driver, dan LCD, serta dilengkapi penutup engsel dan lubang ventilasi kecil untuk melindungi perangkat dari debu, kelembapan, dan panas. Tandon nutrisi memiliki kapasitas sekitar 60–80 liter dengan dimensi perkiraan diameter 40 cm dan tinggi 60 cm, terbuat dari plastik HDPE yang tahan terhadap bahan kimia dan sinar *UV*, serta dilengkapi penutup rapat dan pipa distribusi nutrisi. Ember nutrisi berkapasitas 5–10 liter dengan diameter 25 cm dan tinggi 30 cm digunakan sebagai wadah cadangan atau pencampuran larutan, dilengkapi tutup transparan dan saluran penghubung.

*Gully* tanam memiliki panjang 4 meter dengan tinggi rak sekitar 1 meter dari lantai, dirancang miring dengan kemiringan 2–3% untuk memastikan aliran larutan nutrisi lancar kembali ke tandon. Lubang tanam dibuat presisi menggunakan *SOLIDWORKS* dengan diameter ±5 cm dan jarak tanam 20–25 cm, sehingga dalam satu gully dapat memuat 16–20 tanaman. Rangka penopang *gully* dibuat dari besi hollow galvanis 4 × 4 cm yang dilapisi cat anti karat untuk memastikan kekuatan dan ketahanan terhadap kondisi lingkungan.

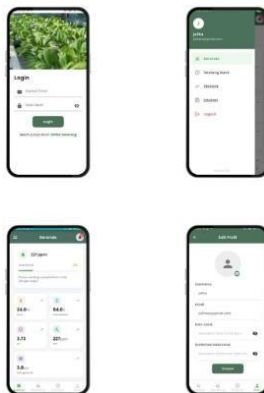


**Gambar 3. 4 *Design* Mekanikal Tandon, Ember Nutrisi & Panel Box**

### 3.1.4. Pembuatan Program

Perancangan program pada sistem *monitoring* dan kontrol nutrisi otomatis ini dilakukan menggunakan *software* Arduino IDE. Program ditujukan untuk membaca data dari sensor DHT11 untuk mengukur suhu dan kelembaban udara, sensor pH SEN0161 untuk mengetahui tingkat keasaman larutan nutrisi, sensor TDS SEN0244 untuk mengukur konsentrasi total padatan terlarut, serta sensor ultrasonik HC-SR04 yang digunakan untuk memantau ketinggian larutan dalam tangki nutrisi. Seluruh pembacaan sensor diprogram agar hasilnya dapat ditampilkan melalui *serial monitor*, LCD I2C, dan dikirimkan secara *real-time* ke *Firebase* untuk pemantauan jarak jauh. Pemrograman dilakukan dengan memperhatikan spesifikasi dari masing-masing sensor, termasuk penggunaan *library* yang sesuai dan penyesuaian pin sesuai dengan rancangan desain elektrikal yang telah dibuat sebelumnya. Hal ini penting agar setiap sensor dapat berfungsi dengan optimal dan sistem dapat berjalan secara akurat dan stabil.

### 3.1.5. Pengembangan Aplikasi *Mobile*



**Gambar 3. 5 Mockup Aplikasi Android**

Gambar 3.5 menunjukkan *mockup* aplikasi Android yang digunakan untuk sistem *monitoring* dan kontrol nutrisi otomatis pada *greenhouse*. Aplikasi ini menampilkan antarmuka pengguna yang dirancang sederhana agar mudah digunakan, dengan fitur utama seperti pemantauan data suhu, kelembapan, pH, TDS, dan ketinggian larutan secara *real-time* yang dikirimkan dari mikrokontroler ESP32 melalui *Firebase Realtime Database*. Selain itu, aplikasi dapat memberikan notifikasi jika terdeteksi penyimpangan kondisi nutrisi, sehingga pengguna dapat segera melakukan penyesuaian. Desain *mockup* ini dibuat menggunakan

*framework Flutter* di *Visual Studio Code* dan *Android Studio*, menampilkan halaman utama berisi ringkasan data sensor, halaman pengaturan, serta tampilan notifikasi, yang semuanya mendukung proses pemantauan tanaman di *greenhouse* menjadi lebih efisien.

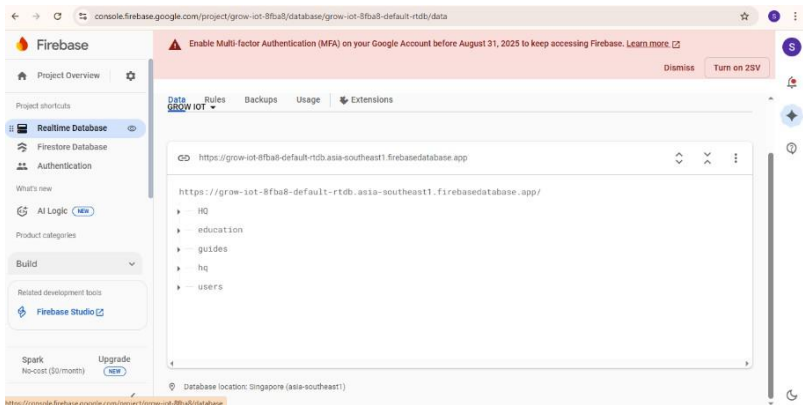
### 3.1.5. Pengembangan *Website*



**Gambar 3. 6 Mockup Website**

Gambar 3.6 menunjukkan *mockup website* untuk sistem *monitoring* dan kontrol nutrisi otomatis pada *greenhouse*. *Website* ini dirancang sebagai alternatif pemantauan berbasis *web* yang terhubung langsung ke *Firebase Realtime Database*, memungkinkan pengguna memantau kondisi nutrisi secara *online* tanpa perlu mengakses aplikasi *mobile*. Antarmuka *website* menampilkan data sensor secara *real-time*, meliputi suhu, kelembapan, pH, TDS, dan ketinggian larutan. Desain dibuat responsif sehingga dapat diakses dari berbagai perangkat, baik komputer maupun ponsel, dengan tampilan yang sederhana dan mudah dipahami. Pengembangan *website* dilakukan menggunakan *Visual Studio Code* dengan bahasa pemrograman HTML, CSS, dan JavaScript, sehingga pengguna dapat memantau dan mengelola sistem nutrisi tanaman dari mana saja secara efisien.

### 3.1.6. Pembuatan *Firebase Realtime Database*



**Gambar 3. 7 *Firebase Realtime Database***

Gambar 3.7 menunjukkan tampilan *Firebase Realtime Database* yang digunakan sebagai media penyimpanan dan pengelolaan data pada sistem monitoring dan kontrol nutrisi otomatis. *Database* ini berfungsi untuk menerima data dari berbagai sensor, seperti DHT11, SEN0161, TDS, dan HC-SR04, yang membaca parameter suhu, kelembapan, pH, TDS, dan ketinggian larutan. Data analog dari sensor diolah melalui ADC, sedangkan data digital dibaca langsung oleh mikrokontroler ESP32, yang kemudian melakukan proses kalibrasi, *filtering*, dan kompensasi sebelum disusun dalam format JSON lengkap dengan informasi waktu dan identitas perangkat. Selanjutnya, data dikirim melalui koneksi Wi-Fi ke *Firebase Realtime Database* menggunakan API key dan URL yang telah dikonfigurasi. *Database* ini menyimpan data secara terstruktur sehingga dapat diakses secara *real-time* baik melalui aplikasi *mobile* maupun *website*. Selain itu, sistem mendukung pembaruan otomatis tanpa perlu *refresh*, memudahkan pengguna memantau kondisi nutrisi secara langsung dan mengambil tindakan cepat jika terjadi penyimpangan, sehingga proses pengendalian nutrisi pada *greenhouse* lebih efisien dan responsif.

### 3.1.7. Pembuatan Kontrol dengan Metode PID

Pembuatan sistem kontrol dengan metode PID (*Proportional-Integral-Derivative*) bertujuan untuk mengatur kerja aktuator (misalnya pompa nutrisi) agar nilai parameter yang diukur, seperti TDS (*Total Dissolved Solids*), dapat mendekati nilai setpoint secara otomatis. PID merupakan metode kontrol umpan balik yang menghitung selisih antara nilai *setpoint* dan nilai sensor (*error*), lalu

menghasilkan *output* berupa sinyal PWM yang akan digunakan untuk mengendalikan aktuator.

Dalam implementasinya, tiga komponen utama yaitu Kp (*proportional*), Ki (*integral*), dan Kd (*derivative*) diatur melalui proses *tuning*. Nilai-nilai ini dimasukkan dalam program mikrokontroler (ESP32), dan perhitungan PID dilakukan secara periodik dalam *loop* program. *Output* hasil perhitungan digunakan untuk mengatur kecepatan atau durasi kerja pompa, agar penambahan nutrisi dilakukan secara presisi. Metode ini memungkinkan sistem beradaptasi secara dinamis terhadap perubahan lingkungan atau kondisi larutan, sehingga proses pengontrolan menjadi lebih stabil dan efisien dibandingkan dengan metode kontrol konvensional seperti *ON/OFF*.

### 3.2. Alat dan Bahan

**Tabel 3. 1 Alat dan Bahan**

| Alat dan Bahan                                                      | Kuantitas | Harga Satuan (Rp) | Total (Rp) |
|---------------------------------------------------------------------|-----------|-------------------|------------|
| esp32 - ESP 32 WROOM 32<br>ESP32 WiFi BLE Development Board + Cable | 2 Buah    | 135.000 -,        | 270.000 -, |
| Sensor pH                                                           | 1 Buah    | 555.000 -,        | 555.000 -, |
| Sensor TDS Analog                                                   | 1 Buah    | 85,000            | 85,000     |
| Sensor DHT 11                                                       | 1 Buah    | 35,000            | 35,000     |
| Sensor Ultrasonic HC-SR                                             | 1 Buah    | 120,000           | 120,000    |
| StepDown12 V to 5 V                                                 | 1 Buah    | 45,000            | 45,000     |
| Driver Motor L298N                                                  | 1 Buah    | 250,000           | 250,000    |
| 12V DC Dosing Pump<br>Peristaltic ead wit Connector                 | 1 Buah    | 150,000           | 150,000    |
| LCD 20x4 I2C                                                        | 1 Buah    | 40,000            | 40,000     |
| Power supply PSU 12V 10A                                            | 1 Buah    | 160,000           | 160,000    |
| Socket JST                                                          | 30 Buah   | 1,500             | 15,000     |
| Probe BNC male - female                                             | 1 Buah    | 25,000            | 25,000     |
| Terminal 2 Pin                                                      | 2 Buah    | 20,000            | 20,000     |
| Jumper Male To Female                                               |           | 20,000            | 20,000     |
| Jumper Male To male                                                 |           | 15,000            | 15,000     |
| Gully Tanam NFT 2 m                                                 |           | 20,000            | 20,000     |
| Lem Tembak                                                          |           | 150,000           | 150,000    |

|                            |        |         |         |
|----------------------------|--------|---------|---------|
| Lem Pipa                   |        | 100,000 | 100,000 |
| Panel Box Outdoor 20×30×12 | 1 Buah | 200,000 | 200,000 |
| Cable Spiral               |        | 30,000  | 30,000  |

### 3.3. Pengujian

Dalam pengujian ini, terdapat beberapa tahapan yang akan dilakukan. Sebelum melakukan pengujian secara keseluruhan, dilakukan beberapa tahap pengujian sebagai berikut:

#### 3.3.1. Pengujian Sensor pH 0161

Tabel 3.2 Pengujian Sensor pH 0161 menunjukkan hasil pengujian sensor pH 0161 yang digunakan dalam sistem *monitoring* dan kontrol nutrisi otomatis pada *greenhouse*. Pengukuran pH sangat penting untuk memastikan tanaman menerima nutrisi secara optimal. Sensor pH 0161 berfungsi memantau tingkat keasaman atau kebasaan larutan nutrisi, di mana nilai pH memengaruhi kemampuan akar tanaman dalam menyerap unsur hara. Umumnya, tanaman hidroponik tumbuh optimal pada rentang pH 5,5 hingga 6,5. Nilai pH yang terlalu rendah (asam) atau terlalu tinggi (basa) dapat menyebabkan defisiensi nutrisi meskipun unsur hara tersedia. Oleh karena itu, pengukuran dan penyesuaian secara otomatis sangat dibutuhkan agar pertumbuhan tanaman tetap optimal.

**Tabel 3. 2 Pengujian Sensor pH 0161**

| No | Kondisi Larutan Nutrisi | Nilai pH | Keterangan    |
|----|-------------------------|----------|---------------|
| 1  | pH terlalu asam         | 4.0      | Terlalu Asam  |
| 2  | pH optimal              | 7.0      | Netral        |
| 3  | pH terlalu basa         | 10.0     | Terlalu biasa |

#### 3.3.2. Pengujian Sensor Gravity: TDS Analog

Tabel 3.3 Pengujian Sensor *Gravity*: TDS Analog menunjukkan hasil pengukuran jumlah padatan terlarut (*Total Dissolved Solids* / TDS) dalam larutan nutrisi di sistem hidroponik *greenhouse*. Nilai TDS menunjukkan konsentrasi nutrisi yang tersedia bagi tanaman. Pengukuran TDS sangat penting karena kelebihan atau kekurangan unsur hara dapat memengaruhi pertumbuhan, produktivitas, dan kualitas hasil panen tanaman. Dalam sistem *monitoring* dan kontrol nutrisi otomatis pada *greenhouse* Politeknik Negeri Batam, sensor TDS membantu memastikan bahwa konsentrasi nutrisi tetap berada pada batas optimal. Nilai TDS yang terlalu rendah menunjukkan larutan kekurangan nutrisi, sedangkan nilai yang terlalu tinggi dapat menyebabkan penumpukan garam yang membahayakan tanaman.

**Tabel 3. 3 Pengujian Sensor Gravity: TDS Analog**

| No | Kondisi Larutan Nutrisi | Nilai TDS | Keterangan |
|----|-------------------------|-----------|------------|
| 1  | Larutan terlalu encer   | 300       | Rendah     |
| 2  | Larutan Optimal         | 1000      | ideal      |
| 3  | Larutan terlalu pekat   | 1800      | Tinggi     |

### 3.3.3. Pengujian Sensor Ultrasonic HC-SR04

Sensor *Ultrasonic HC-SR04* digunakan untuk mengukur ketinggian air pada tandon utama sistem hidroponik. Sensor ini bekerja dengan memancarkan gelombang *ultrasonik* dan mengukur waktu pantulan gelombang tersebut untuk menentukan jarak antara permukaan air dan sensor. Dalam sistem *greenhouse* dengan kapasitas 40 lubang hidroponik, kebutuhan air nutrisi cukup besar untuk menjaga aliran tetap stabil.

Berdasarkan perhitungan rata-rata, setiap lubang membutuhkan sekitar 1–1.5 liter larutan nutrisi. Maka, total kebutuhan minimal air adalah sekitar 40–60 liter. Untuk itu, tandon utama sebaiknya memiliki tinggi air minimal 20–25 cm tergantung pada diameter dan bentuk tandon, agar mampu menampung *volume* air yang dibutuhkan sekaligus menjaga tekanan aliran tetap stabil. Sensor *ultrasonic* ditempatkan pada bagian atas tandon untuk memantau ketinggian air secara *real-time* dan memberikan peringatan jika *volume* air mendekati batas minimum.

### 3.3.4. Pengujian Sensor DHT11

Sensor *DHT11* digunakan untuk memantau suhu dan kelembapan udara dalam *greenhouse*. Kedua parameter ini berperan penting dalam mendukung pertumbuhan tanaman, terutama untuk sistem hidroponik yang membutuhkan kondisi lingkungan yang stabil. Sensor *DHT11* mengukur suhu menggunakan *termistor* dan kelembapan menggunakan sensor *kapasitor*. Data yang diperoleh digunakan untuk mengatur sistem otomatis seperti kipas, *mist maker*, atau pemanas. Suhu ideal untuk tanaman seperti selada berada pada kisaran 20–28°C, dan kelembapan optimal berada di rentang 60–80% RH.

### 3.3.5. Pengujian Peristaltic Pump Dosing Pump 12v DC

*Peristaltic pump 12V DC* digunakan untuk menambahkan larutan nutrisi ke dalam tandon utama berdasarkan pembacaan sensor *TDS*. Pompa ini tidak dikontrol secara *ON/OFF*, melainkan melalui pengaturan kecepatan menggunakan sinyal *PWM* dari mikrokontroler. Semakin rendah nilai *TDS* dari ambang target (sekitar 1000 ppm), semakin cepat pompa bekerja. Saat nilai *TDS* mendekati target, kecepatan pompa dikurangi secara bertahap untuk mencegah kelebihan dosis.

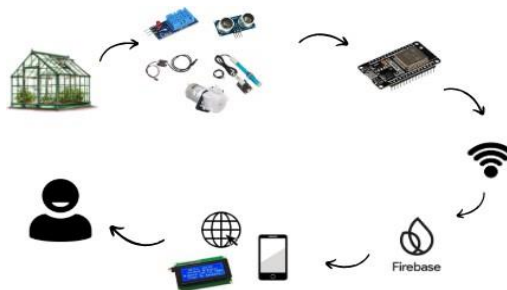
Pengujian menunjukkan bahwa metode pengaturan kecepatan ini lebih presisi dan efisien dibandingkan sistem *ON/OFF*. Pompa dapat menjaga kestabilan konsentrasi nutrisi di dalam tandon tanpa menyebabkan fluktuasi tajam, sehingga sangat cocok digunakan dalam sistem hidroponik otomatis di *greenhouse*.

### 3.4. Cara Kerja Sistem

Seperti pada Gambar 3.8 Sistem *monitoring* dan kontrol nutrisi otomatis ini diterapkan pada *greenhouse* di Politeknik Negeri Batam. Proses dimulai dari area *greenhouse* sebagai tempat pertumbuhan tanaman hidroponik. Di dalam *greenhouse*, terpasang sejumlah sensor yaitu sensor *pH 0161* untuk mengukur tingkat keasaman larutan, sensor *TDS Gravity Analog* untuk memantau konsentrasi nutrisi, sensor *DHT11* untuk mengukur suhu dan kelembapan udara, serta sensor *ultrasonik HC-SR04* untuk mengetahui ketinggian air pada tandon utama. Data dari semua sensor dikirim ke mikrokontroler *ESP32* yang berfungsi sebagai pusat pengendali.

*ESP32* memproses informasi secara *real-time* dan mengatur kerja *peristaltic pump 12V DC* untuk menambahkan larutan nutrisi sesuai kebutuhan. Pompa dikontrol melalui sistem *PWM*, di mana kecepatan aliran disesuaikan berdasarkan nilai *TDS*—semakin rendah nilai *TDS*, semakin cepat pompa bekerja. Selain mengontrol pompa, *ESP32* juga menampilkan data hasil pengukuran pada *LCD 16x2* agar dapat dipantau langsung oleh operator di lapangan. Di saat yang sama, *ESP32* mengirimkan data ke *Firebase Realtime Database* melalui koneksi *WiFi*, sehingga semua informasi tersimpan secara *online* dan dapat dipantau dari jarak jauh.

Data yang tersimpan di *Firebase* dapat diakses oleh pengguna melalui dua cara: melalui *website* dan melalui aplikasi *mobile Android*. Dengan kedua media ini, pengguna dapat melihat kondisi sistem secara *real-time* dan mengambil tindakan jika diperlukan, baik secara lokal maupun dari lokasi yang jauh.



Gambar 3. 8 Cara Kerja Sistem

## Bab 4. Hasil dan Pembahasan

### 4.1. Data Hasil Penelitian

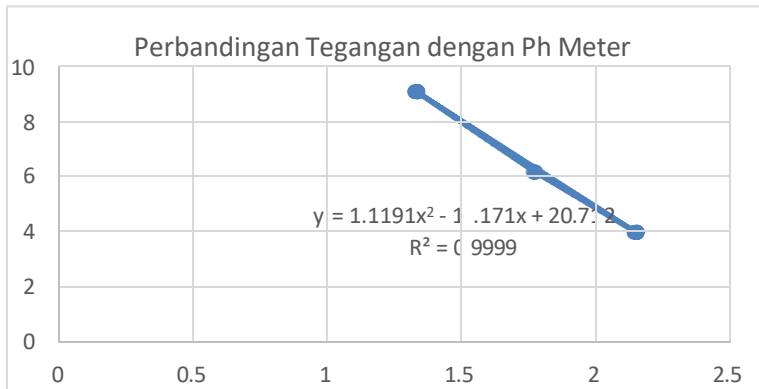
#### 4.1.1. Data Hasil Pengujian Sensor pH 0161

**Tabel 4. 2 Tegangan Sensor Ph Sebelum Linierisasi**

| pH<br>METER       | Tegangan Sensor Ph Serial Monitor |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX   | MIN   |
|-------------------|-----------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|-------|-------|
|                   | Perc 1                            | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |       |       |
| 3.98              | 2.179                             | 2.202  | 2.166  | 2.153  | 2.165  | 2.141  | 2.144  | 2.153  | 2.14   | 2.136   | 2.1579    | 2.202 | 2.136 | 2.136 |
| 6.16              | 1.769                             | 1.801  | 1.769  | 1.776  | 1.775  | 1.779  | 1.791  | 1.814  | 1.782  | 1.769   | 1.7825    | 1.814 | 1.769 | 1.769 |
| 9.08              | 1.349                             | 1.33   | 1.331  | 1.337  | 1.345  | 1.341  | 1.339  | 1.354  | 1.347  | 1.354   | 1.3427    | 1.354 | 1.33  | 1.33  |
| 3.98              | 2.193                             | 2.186  | 2.173  | 2.153  | 2.181  | 2.146  | 2.153  | 2.152  | 2.138  | 2.136   | 2.1611    | 2.193 | 2.136 | 2.136 |
| 6.16              | 1.77                              | 1.741  | 1.744  | 1.778  | 1.8    | 1.776  | 1.788  | 1.77   | 1.779  | 1.8     | 1.7746    | 1.8   | 1.741 | 1.741 |
| 9.08              | 1.335                             | 1.216  | 1.353  | 1.337  | 1.355  | 1.339  | 1.355  | 1.354  | 1.354  | 1.336   | 1.3334    | 1.355 | 1.216 | 1.216 |
| 3.98              | 2.175                             | 2.183  | 2.177  | 2.162  | 2.175  | 2.145  | 2.142  | 2.153  | 2.145  | 2.135   | 2.1592    | 2.183 | 2.135 | 2.135 |
| 6.16              | 1.771                             | 1.795  | 1.763  | 1.782  | 1.784  | 1.766  | 1.787  | 1.782  | 1.785  | 1.777   | 1.7792    | 1.795 | 1.763 | 1.763 |
| 9.08              | 1.341                             | 1.328  | 1.339  | 1.354  | 1.354  | 1.342  | 1.354  | 1.333  | 1.337  | 1.352   | 1.3434    | 1.354 | 1.328 | 1.328 |
| 3.98              | 2.18                              | 2.167  | 2.17   | 2.171  | 2.157  | 2.152  | 2.148  | 2.144  | 2.138  | 2.148   | 2.1575    | 2.18  | 2.138 | 2.138 |
| 6.16              | 1.794                             | 1.794  | 1.768  | 1.772  | 1.775  | 1.782  | 1.775  | 1.771  | 1.795  | 1.769   | 1.7795    | 1.795 | 1.768 | 1.768 |
| 9.08              | 1.326                             | 1.354  | 1.341  | 1.343  | 1.341  | 1.351  | 1.354  | 1.332  | 1.351  | 1.367   | 1.346     | 1.367 | 1.326 | 1.326 |
| 3.98              | 2.172                             | 2.151  | 2.149  | 2.153  | 2.153  | 2.145  | 2.141  | 2.146  | 2.141  | 2.14    | 2.1491    | 2.172 | 2.14  | 2.14  |
| 6.16              | 1.792                             | 1.766  | 1.755  | 1.782  | 1.772  | 1.792  | 1.784  | 1.771  | 1.79   | 1.783   | 1.7787    | 1.792 | 1.755 | 1.755 |
| 9.08              | 1.333                             | 1.315  | 1.339  | 1.339  | 1.343  | 1.349  | 1.355  | 1.356  | 1.351  | 1.354   | 1.3434    | 1.356 | 1.315 | 1.315 |
| Rata Rata Error : |                                   |        |        |        |        |        |        |        |        |         | 1.78      |       |       |       |

**Tabel 4. 1 Sensor Ph Sebelum Linierisasi**

| pH<br>METER       | Sensor Ph Serial Monitor |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX  | MIN  |
|-------------------|--------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|------|------|
|                   | Perc 1                   | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |      |      |
| 3.98              | 4.85                     | 4.7    | 4.86   | 4.91   | 4.86   | 4.96   | 4.95   | 4.91   | 4.97   | 4.99    | 4.90      | 23.02 | 4.99 | 4.7  |
| 6.16              | 6.45                     | 6.4    | 6.54   | 6.51   | 6.52   | 6.5    | 6.45   | 6.35   | 6.49   | 6.54    | 6.48      | 5.11  | 6.54 | 6.35 |
| 9.08              | 8.32                     | 8.4    | 8.4    | 8.37   | 8.34   | 8.36   | 8.37   | 8.3    | 8.33   | 8.3     | 8.35      | 8.05  | 8.4  | 8.3  |
| 3.98              | 4.74                     | 4.77   | 4.83   | 4.91   | 4.8    | 4.94   | 4.91   | 4.92   | 4.98   | 4.98    | 4.88      | 22.56 | 4.98 | 4.74 |
| 6.16              | 6.53                     | 6.66   | 6.65   | 6.5    | 6.41   | 6.51   | 6.46   | 6.53   | 5.5    | 6.41    | 6.42      | 4.16  | 6.66 | 5.5  |
| 9.08              | 8.35                     | 8.46   | 8.3    | 8.37   | 8.3    | 8.36   | 8.3    | 8.3    | 8.38   | 8.34    | 8.13      | 8.46  | 8.3  |      |
| 3.98              | 4.82                     | 4.79   | 4.81   | 4.87   | 4.82   | 4.95   | 4.96   | 4.91   | 4.95   | 4.99    | 4.89      | 22.79 | 4.99 | 4.79 |
| 6.16              | 6.53                     | 6.43   | 6.57   | 6.49   | 6.48   | 6.55   | 6.47   | 6.49   | 6.47   | 6.51    | 6.50      | 5.50  | 6.57 | 6.43 |
| 9.08              | 8.36                     | 8.41   | 8.36   | 8.3    | 8.3    | 8.35   | 8.3    | 8.39   | 8.37   | 8.31    | 8.35      | 8.09  | 8.41 | 8.3  |
| 3.98              | 4.8                      | 4.85   | 4.84   | 4.84   | 4.89   | 4.92   | 4.93   | 4.95   | 4.98   | 4.93    | 4.89      | 22.94 | 4.98 | 4.8  |
| 6.16              | 6.44                     | 6.44   | 6.54   | 6.53   | 6.51   | 6.49   | 6.51   | 6.53   | 6.43   | 6.54    | 6.50      | 5.45  | 6.54 | 6.43 |
| 9.08              | 8.42                     | 8.3    | 8.36   | 8.53   | 8.36   | 8.31   | 8.3    | 8.39   | 8.31   | 8.25    | 8.35      | 8.01  | 8.53 | 8.25 |
| 3.98              | 4.83                     | 4.92   | 4.93   | 4.91   | 4.91   | 4.95   | 4.96   | 4.94   | 4.96   | 4.97    | 4.93      | 23.82 | 4.97 | 4.83 |
| 6.16              | 6.44                     | 6.55   | 6.6    | 6.49   | 6.53   | 6.44   | 6.48   | 6.53   | 6.45   | 6.48    | 6.50      | 5.50  | 6.6  | 6.44 |
| 9.08              | 8.39                     | 8.46   | 8.37   | 8.37   | 8.35   | 8.32   | 8.3    | 8.29   | 8.31   | 8.3     | 8.35      | 8.08  | 8.46 | 8.29 |
| Rata Rata Error : |                          |        |        |        |        |        |        |        |        |         | 12.08     |       |      |      |



Gambar 4. 1 Grafik sebelum Linierisasi

Tabel 4. 3 Sensor Ph Setelah Linierisasi

| pH<br>METER       | Sensor Ph Serial Monitor Setelah Linierisasi |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX  | MIN  |
|-------------------|----------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|------|------|
|                   | Perc 1                                       | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |      |      |
| 3.89              | 3.86                                         | 3.47   | 3.72   | 3.73   | 3.8    | 3.73   | 3.77   | 3.81   | 3.79   | 3.88    | 3.76      | 3.44  | 3.88 | 3.47 |
| 6.27              | 6.33                                         | 6.15   | 6.14   | 6.1    | 6.24   | 5.33   | 6.15   | 6.21   | 6.16   | 5.61    | 6.04      | 3.64  | 6.33 | 5.33 |
| 9.35              | 9.32                                         | 9      | 9.36   | 8.91   | 8.99   | 9.09   | 9.18   | 9.21   | 8.79   | 8.91    | 9.08      | 2.93  | 9.36 | 8.79 |
| 3.89              | 3.8                                          | 3.83   | 3.93   | 3.73   | 3.66   | 3.74   | 3.83   | 3.94   | 3.93   | 3.96    | 3.84      | 1.41  | 3.96 | 3.66 |
| 6.27              | 6.01                                         | 6.13   | 6.24   | 6      | 6.07   | 6.08   | 6.24   | 6.11   | 6.14   | 6.08    | 6.11      | 2.55  | 6.24 | 6    |
| 9.35              | 9.18                                         | 9.33   | 8.18   | 9.07   | 8.99   | 9.09   | 9.1    | 9.06   | 9      | 8.99    | 9.00      | 3.75  | 9.33 | 8.18 |
| 3.89              | 3.53                                         | 3.75   | 3.79   | 3.79   | 3.78   | 3.88   | 4      | 3.98   | 4      | 3.85    | 3.84      | 1.41  | 4    | 3.53 |
| 6.27              | 6.04                                         | 6.09   | 6.08   | 6.02   | 6.02   | 6.06   | 6.17   | 6.01   | 5.99   | 6.12    | 6.06      | 3.35  | 6.17 | 5.99 |
| 9.35              | 9.04                                         | 9.31   | 9.09   | 9.18   | 9.02   | 9.01   | 9.14   | 9.18   | 9.06   | 8.99    | 9.10      | 2.65  | 9.31 | 8.99 |
| 3.89              | 3.41                                         | 3.74   | 3.7    | 3.73   | 3.63   | 3.79   | 3.89   | 3.88   | 3.86   | 3.87    | 3.75      | 3.60  | 3.89 | 3.41 |
| 6.27              | 6.02                                         | 6.21   | 6.09   | 6.24   | 6.18   | 5.61   | 6.24   | 6.24   | 6.16   | 6.23    | 6.12      | 2.36  | 6.24 | 5.61 |
| 9.35              | 9.03                                         | 9.05   | 8.79   | 9.06   | 9.17   | 9.14   | 9.22   | 9.17   | 9.07   | 9.18    | 9.09      | 2.80  | 9.22 | 8.79 |
| 3.89              | 3.63                                         | 3.74   | 3.74   | 3.68   | 3.86   | 3.76   | 3.73   | 3.85   | 3.86   | 4       | 3.79      | 2.70  | 4    | 3.63 |
| 6.27              | 6.09                                         | 6.19   | 6.2    | 6.08   | 6.18   | 6.17   | 5.45   | 6.15   | 6.17   | 6.16    | 6.08      | 2.97  | 6.2  | 5.45 |
| 9.18              | 9.09                                         | 8.99   | 9.27   | 8.99   | 9.15   | 8.99   | 8.72   | 9.05   | 8.99   | 9.18    | 9.04      | 1.50  | 9.27 | 8.72 |
| Rata Rata Error : |                                              |        |        |        |        |        |        |        |        |         |           | 2.74  |      |      |

Pengujian Gambar 4.1, Tabel 4.1, Tabel 4.2, dan Tabel 4.3 menampilkan hasil pengujian sensor pH SEN0161 sebelum dan sesudah proses linierisasi menggunakan tiga larutan *buffer* standar, yaitu pH 3,89; pH 6,27; dan pH 9,35, dengan masing-masing dilakukan tiga kali percobaan. Tabel 4.1 menunjukkan data tegangan keluaran sensor sebelum linierisasi, sedangkan Tabel 4.2 menampilkan hasil pembacaan pH oleh sensor sebelum linierisasi. Grafik pada Gambar 4.1 memperlihatkan hubungan antara tegangan sensor dengan nilai pH aktual yang diukur menggunakan pH meter, menghasilkan persamaan regresi kuadratik

$y=1.119x^2-10.171x+20.717$  dengan nilai koefisien determinasi  $R^2=0.9999$   $R^2=0.9999$ , menandakan hubungan yang sangat kuat. Setelah persamaan regresi ini diimplementasikan pada mikrokontroler untuk proses linierisasi, hasil

pengukuran yang ditampilkan pada Tabel 4.3 menunjukkan akurasi yang meningkat signifikan, dengan rata-rata *error* berkurang dari 0,49 menjadi 0,12. Hal ini membuktikan bahwa proses linierisasi berhasil meningkatkan ketepatan pembacaan sensor pH, sehingga data yang dihasilkan lebih dapat diandalkan untuk pemantauan nutrisi pada sistem hidroponik.

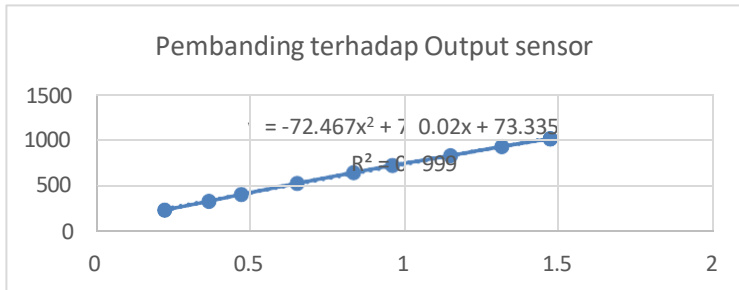
#### 4.1.2. Data Hasil Pengujian Sensor *Gravity*: TDS Analog

**Tabel 4. 4 Data Pengujian Sensor Gravity: TDS Analog sebelum Linierisasi**

| TDS<br>METER      | Sensor TDS (PPM) Serial Monitor Sebelum Linierisasi |         |         |         |         |         |         |         |         |         | Rata Rata | error | MAX     | MIN    |
|-------------------|-----------------------------------------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|-----------|-------|---------|--------|
|                   | Perc 1                                              | Perc 2  | Perc 3  | Perc 4  | Perc 5  | Perc 6  | Perc 7  | Perc 8  | Perc 9  | Perc 10 |           |       |         |        |
| 117               | 38.86                                               | 40.2    | 41.54   | 40.2    | 41.54   | 42.22   | 40.2    | 43.56   | 91.72   | 95.61   | 51.57     | 55.93 | 95.61   | 38.86  |
| 241               | 176.43                                              | 176.43  | 177.05  | 177.05  | 177.05  | 177.05  | 177.05  | 177.05  | 225.67  | 225.67  | 186.65    | 22.55 | 225.67  | 176.43 |
| 336               | 281.3                                               | 282.47  | 284.22  | 282.47  | 278.96  | 283.64  | 281.3   | 282.47  | 328.77  | 327.05  | 291.27    | 13.31 | 328.77  | 278.96 |
| 412               | 357.9                                               | 361.31  | 361.88  | 359.04  | 356.76  | 356.2   | 355.06  | 357.9   | 401.46  | 398.08  | 366.56    | 11.03 | 401.46  | 355.06 |
| 529               | 481.04                                              | 487.75  | 490.55  | 486.07  | 479.92  | 490.55  | 473.21  | 484.95  | 526.93  | 514.61  | 491.56    | 7.08  | 526.93  | 473.21 |
| 646               | 615.68                                              | 617.4   | 617.97  | 607.69  | 607.69  | 618.54  | 602.56  | 613.87  | 648.44  | 647.87  | 619.77    | 4.06  | 648.44  | 602.56 |
| 724               | 709.32                                              | 707.54  | 708.73  | 692.18  | 695.72  | 702.81  | 678.84  | 702.81  | 756.16  | 748.28  | 710.24    | 1.90  | 756.16  | 678.84 |
| 826               | 855.66                                              | 855.02  | 856.31  | 848.58  | 831.96  | 849.22  | 831.32  | 828.79  | 876.46  | 876.46  | 850.98    | 3.02  | 876.46  | 828.79 |
| 926               | 992.81                                              | 908.19  | 999.91  | 994.93  | 987.85  | 970.31  | 966.13  | 990.98  | 1010.63 | 1123.39 | 994.51    | 7.40  | 1123.39 | 908.19 |
| 1009              | 1101.02                                             | 1127.28 | 1118.73 | 1110.23 | 1128.84 | 1106.39 | 1109.46 | 1123.39 | 1166.8  | 287.98  | 1038.01   | 2.88  | 1166.8  | 287.98 |
| Rata Rata Error : |                                                     |         |         |         |         |         |         |         |         |         |           | 12.92 |         |        |

**Tabel 4. 5 Data Pengujian Tegangan Sensor Gravity: TDS Analog sebelum Linierisasi**

| TDS<br>METER      | Tegangan TDS (PPM) Serial Monitor Sebelum Linierisasi |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX  | MIN  |
|-------------------|-------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|------|------|
|                   | Perc 1                                                | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |      |      |
| 117               | 0.05                                                  | 0.05   | 0.05   | 0.05   | 0.05   | 0.05   | 0.05   | 0.05   | 0.11   | 0.12    | 0.06      | 0.12  | 0.05 | 0.05 |
| 241               | 0.22                                                  | 0.22   | 0.22   | 0.22   | 0.2    | 0.22   | 0.22   | 0.22   | 0.28   | 0.28    | 0.23      | 0.28  | 0.2  | 0.2  |
| 336               | 0.36                                                  | 0.36   | 0.36   | 0.36   | 0.36   | 0.36   | 0.36   | 0.36   | 0.43   | 0.42    | 0.37      | 0.43  | 0.36 | 0.36 |
| 412               | 0.47                                                  | 0.47   | 0.47   | 0.47   | 0.46   | 0.46   | 0.46   | 0.47   | 0.53   | 0.52    | 0.48      | 0.53  | 0.46 | 0.46 |
| 529               | 0.64                                                  | 0.65   | 0.66   | 0.65   | 0.64   | 0.66   | 0.63   | 0.65   | 0.71   | 0.7     | 0.66      | 0.71  | 0.63 | 0.63 |
| 646               | 0.84                                                  | 0.84   | 0.84   | 0.82   | 0.82   | 0.84   | 0.82   | 0.83   | 0.88   | 0.88    | 0.84      | 0.88  | 0.82 | 0.82 |
| 724               | 0.97                                                  | 0.96   | 0.96   | 0.94   | 0.95   | 0.96   | 0.92   | 0.96   | 1.03   | 1.02    | 0.97      | 1.03  | 0.92 | 0.92 |
| 826               | 1.16                                                  | 1.16   | 1.16   | 1.15   | 1.13   | 1.15   | 1.13   | 1.12   | 1.21   | 1.18    | 1.16      | 1.21  | 1.12 | 1.12 |
| 926               | 1.32                                                  | 1.22   | 1.33   | 1.32   | 1.32   | 1.29   | 1.29   | 1.32   | 1.34   | 1.46    | 1.32      | 1.46  | 1.22 | 1.22 |
| 1009              | 1.44                                                  | 1.47   | 1.46   | 1.45   | 1.47   | 1.45   | 1.45   | 1.46   | 1.5    | 1.63    | 1.48      | 1.63  | 1.44 | 1.44 |
| Rata Rata Error : |                                                       |        |        |        |        |        |        |        |        |         |           | 0.83  |      |      |



Gambar 4. 2 Grafik TDS sebelum Linierisasi

Tabel 4. 6 Data Hasil Pengujian Sensor Gravity: TDS Analog Setelah Linierisasi

| TDS               | Sensor TDS (PPM) Serial Monitor Setelah Linierisasi |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX    | MIN    |
|-------------------|-----------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|--------|--------|
| METER             | Perc 1                                              | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |        |        |
| 117               | 119.57                                              | 109.57 | 116.1  | 109.57 | 109.57 | 110.76 | 112.54 | 115.5  | 116.1  | 114.32  | 113.36    | 3.11  | 119.57 | 109.57 |
| 241               | 218.4                                               | 220.7  | 217.24 | 220.12 | 215.21 | 212.63 | 221.85 | 218.97 | 223    | 218.97  | 218.71    | 9.25  | 223    | 212.63 |
| 336               | 312.67                                              | 314.91 | 319.94 | 319.38 | 318.82 | 312.67 | 321.61 | 318.26 | 320.5  | 320.5   | 317.93    | 5.38  | 321.61 | 312.67 |
| 412               | 414.56                                              | 409.68 | 402.08 | 403.71 | 400.99 | 402.08 | 401.53 | 399.35 | 400.99 | 400.44  | 403.54    | 2.05  | 414.56 | 399.35 |
| 529               | 528.78                                              | 524.61 | 515.76 | 515.73 | 518.48 | 516.77 | 516.25 | 514.16 | 512.06 | 512.06  | 517.47    | 2.18  | 528.78 | 512.06 |
| 646               | 610.84                                              | 614.37 | 611.35 | 610.84 | 606.29 | 608.31 | 606.29 | 609.83 | 601.73 | 604.27  | 608.41    | 5.82  | 614.37 | 601.73 |
| 724               | 702.61                                              | 700.18 | 697.25 | 691.87 | 688.94 | 690.41 | 687.96 | 689.43 | 686.98 | 684.04  | 691.97    | 4.42  | 702.61 | 684.04 |
| 826               | 804.6                                               | 810.14 | 796.19 | 799.47 | 793.89 | 798.53 | 798.53 | 794.79 | 789.16 | 791.98  | 797.73    | 3.42  | 810.14 | 789.16 |
| 926               | 866.64                                              | 870.71 | 860.29 | 866.12 | 857.56 | 863.01 | 859.83 | 862.1  | 853.92 | 856.65  | 861.68    | 6.95  | 870.71 | 853.92 |
| 1009              | 968.23                                              | 969.25 | 960.46 | 963.06 | 961.19 | 962.19 | 963.49 | 960.9  | 956.14 | 962.62  | 962.75    | 4.58  | 969.25 | 956.14 |
| Rata Rata Error : |                                                     |        |        |        |        |        |        |        |        |         | 4.72      |       |        |        |

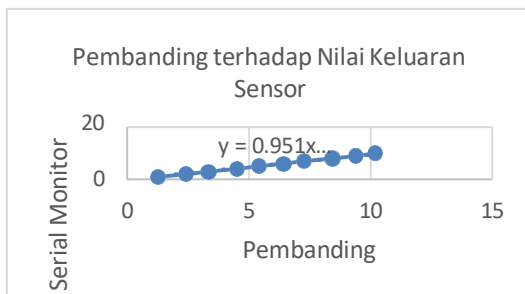
Pengujian Gambar 4.2, Tabel 4.4, Tabel 4.5, dan Tabel 4.6 menampilkan hasil pengujian Sensor Gravity: TDS Analog sebelum dan sesudah proses linierisasi. Tabel 4.5 menunjukkan data tegangan keluaran sensor sebelum linierisasi, sedangkan Tabel 4.4 menampilkan hasil pembacaan TDS (ppm) oleh sensor sebelum linierisasi. Rata-rata *error* pembacaan TDS sebelum linierisasi sebesar 12,392, sementara rata-rata *error* tegangan hanya 0,083. Grafik pada Gambar 4.2 memperlihatkan hubungan antara tegangan keluaran sensor dengan nilai TDS aktual yang diukur menggunakan TDS meter, menghasilkan persamaan regresi linear  $y = -72.467x^2 + 740.02x + 7.33$  dengan koefisien determinasi yang menunjukkan tingkat keterhubungan yang baik.

Setelah persamaan regresi diimplementasikan pada mikrokontroler untuk proses linierisasi, hasil pengukuran yang ditampilkan pada Tabel 4.6 menunjukkan peningkatan akurasi pembacaan sensor secara signifikan, dengan *error* yang lebih kecil dan data yang lebih stabil dibandingkan sebelum linierisasi. Proses ini membuktikan bahwa penerapan metode linierisasi mampu mengoptimalkan kinerja sensor TDS, sehingga hasil pembacaan konsentrasi *Total Dissolved Solid* menjadi lebih akurat dan dapat diandalkan untuk pemantauan kualitas larutan nutrisi pada sistem hidroponik.

### 4.1.3. Data Hasil Pengujian Sensor Ultrasonic HC-SR04

**Tabel 4. 7 Data Hasil Pengujian Sensor Ultrasonic HC-SR04 Sebelum Linierisasi**

| Penggaris<br>(CM) | Sensor Ultrasonic (CM) Serial Monitor |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX   | MIN  |
|-------------------|---------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|-------|------|
|                   | Perc 1                                | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |       |      |
| 1                 | 1.35                                  | 1.35   | 1.35   | 1.26   | 1.35   | 0.84   | 1.52   | 1.11   | 1.09   | 1.43    | 1.27      | 26.50 | 1.52  | 0.84 |
| 2                 | 2.69                                  | 2.61   | 2.2    | 2.61   | 2.1    | 2.69   | 2.11   | 1.94   | 2.52   | 2.61    | 2.41      | 20.40 | 2.69  | 1.94 |
| 3                 | 3.37                                  | 3.12   | 3.29   | 3.69   | 3.12   | 3.29   | 3.52   | 2.86   | 3.95   | 3.12    | 3.33      | 11.10 | 3.95  | 2.86 |
| 4                 | 4.2                                   | 4.61   | 4.72   | 4.55   | 4.63   | 4.7    | 4.46   | 3.88   | 4.87   | 4.37    | 4.50      | 12.48 | 4.87  | 3.88 |
| 5                 | 5.63                                  | 5.55   | 5.8    | 5.21   | 5.46   | 5.53   | 5.46   | 4.72   | 5.38   | 5.38    | 5.41      | 8.24  | 5.8   | 4.72 |
| 6                 | 6.64                                  | 6.55   | 6.55   | 6.38   | 6.64   | 6.47   | 6.47   | 5.72   | 6.3    | 6.38    | 6.41      | 6.83  | 6.64  | 5.72 |
| 7                 | 7.47                                  | 7.47   | 7.06   | 7.38   | 7.47   | 7.38   | 7.13   | 6.64   | 7.72   | 6.96    | 7.27      | 3.83  | 7.72  | 6.64 |
| 8                 | 8.22                                  | 8.47   | 8.9    | 8.56   | 8.3    | 8.47   | 7.98   | 8.47   | 8.73   | 8.3     | 8.44      | 5.50  | 8.9   | 7.98 |
| 9                 | 9.39                                  | 9.31   | 9.48   | 9.48   | 9.31   | 9.05   | 9.73   | 9.32   | 9.65   | 9.22    | 9.39      | 4.38  | 9.73  | 9.05 |
| 10                | 10.56                                 | 10.41  | 9.97   | 9.9    | 9.99   | 9.82   | 10.65  | 10.31  | 10.56  | 9.73    | 10.19     | 1.90  | 10.65 | 9.73 |
| Rata Rata Error : |                                       |        |        |        |        |        |        |        |        |         | 10.12     |       |       |      |



**Gambar 4. 3 Grafik Ultrasonic sebelum Linierisasi**

**Tabel 4. 8 Data Hasil Pengujian Sensor Ultrasonic HC-SR04 Setelah Linierisasi**

| Penggaris<br>(CM) | Sensor Ultrasonic (CM) Serial Monitor Setelah Linierisasi |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX   | MIN  |
|-------------------|-----------------------------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|-------|------|
|                   | Perc 1                                                    | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |       |      |
| 1                 | 1.06                                                      | 1.12   | 1.26   | 1.14   | 1.06   | 1.14   | 0.98   | 1.06   | 0.98   | 0.89    | 1.07      | 6.90  | 1.26  | 0.89 |
| 2                 | 2.01                                                      | 2.17   | 2.17   | 2.17   | 2.09   | 2.01   | 1.77   | 1.93   | 2.16   | 2.09    | 2.06      | 2.85  | 2.17  | 1.77 |
| 3                 | 3.29                                                      | 3.05   | 3.13   | 3.21   | 3.29   | 2.88   | 2.72   | 3.29   | 2.88   | 2.64    | 3.04      | 1.27  | 3.29  | 2.64 |
| 4                 | 4                                                         | 4      | 4.48   | 4.16   | 4.16   | 4.08   | 3.76   | 4.08   | 4.16   | 3.92    | 4.08      | 2.00  | 4.48  | 3.76 |
| 5                 | 5.2                                                       | 5.2    | 5.03   | 5.03   | 5.2    | 4.71   | 4.73   | 5.03   | 5.03   | 4.97    | 5.01      | 0.26  | 5.2   | 4.71 |
| 6                 | 6.15                                                      | 6.23   | 6.31   | 6.07   | 6.07   | 6.15   | 6.07   | 6.15   | 6.07   | 5.99    | 6.13      | 2.10  | 6.31  | 5.99 |
| 7                 | 6.94                                                      | 7.1    | 7.26   | 6.94   | 7.02   | 6.94   | 7.26   | 7.26   | 6.86   | 6.94    | 7.05      | 0.74  | 7.26  | 6.86 |
| 8                 | 7.98                                                      | 8.06   | 8.22   | 7.83   | 7.98   | 7.9    | 7.81   | 8.22   | 7.73   | 7.81    | 7.95      | 0.58  | 8.22  | 7.73 |
| 9                 | 9.25                                                      | 9.01   | 9.01   | 8.93   | 9.01   | 8.69   | 8.77   | 9.09   | 8.61   | 8.87    | 8.92      | 0.84  | 9.25  | 8.61 |
| 10                | 10.13                                                     | 10.29  | 10.05  | 9.82   | 10.13  | 10.05  | 9.8    | 9.5    | 9.82   | 10.13   | 9.97      | 0.28  | 10.29 | 9.5  |
| Rata Rata Error : |                                                           |        |        |        |        |        |        |        |        |         | 1.78      |       |       |      |

Gambar 4.3, Tabel 4.7, dan Tabel 4.8 menampilkan hasil pengujian Sensor Ultrasonic HC-SR04 sebelum dan sesudah proses linierisasi. Tabel 4.7 berisi data pembacaan sensor sebelum linierisasi, yang menunjukkan perbedaan nilai antara hasil pengukuran sensor dengan nilai acuan penggaris. Grafik pada Gambar 4.3 menggambarkan hubungan antara pembacaan sensor dan nilai sebenarnya, menghasilkan persamaan regresi linear  $y=1.05x-2$  yang digunakan untuk mengoreksi hasil pembacaan.

Setelah persamaan regresi ini diimplementasikan pada mikrokontroler, hasil pengukuran yang ditampilkan pada Tabel 4.8 menunjukkan bahwa data yang dihasilkan menjadi lebih stabil dan mendekati nilai ketinggian sebenarnya, dengan rata-rata error yang lebih kecil dibandingkan sebelum linierisasi. Hal ini membuktikan bahwa proses linierisasi mampu meningkatkan akurasi sensor Ultrasonic HC-SR04, sehingga dapat diandalkan untuk mengukur ketinggian larutan pada sistem hidroponik secara presisi. linierisasi.

#### 4.1.4. Data Hasil Pengujian Sensor DHT 11

**Tabel 4. 9 Data Hasil Pengujian Sensor DHT 11**

| Termometer Ruang  | Sensor DHT11 (°C) Serial Monitor |        |        |        |        |        |        |        |        |         | Rata Rata | error | MAX  | MIN  |
|-------------------|----------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|---------|-----------|-------|------|------|
|                   | Perc 1                           | Perc 2 | Perc 3 | Perc 4 | Perc 5 | Perc 6 | Perc 7 | Perc 8 | Perc 9 | Perc 10 |           |       |      |      |
| 21                | 21                               | 21     | 21     | 21     | 21.2   | 22     | 22     | 22     | 22     | 22      | 21.5      | 2.2   | 22   | 21   |
| 23                | 24                               | 25     | 24     | 23     | 23     | 23     | 23     | 23     | 23     | 23      | 23.4      | 1.9   | 25   | 23   |
| 25                | 26                               | 26     | 26     | 26     | 26     | 26     | 26     | 26     | 26     | 26      | 26.0      | 4.0   | 26   | 26   |
| 26.8              | 26                               | 26     | 26     | 26     | 26     | 26     | 26     | 26     | 26     | 26      | 26.0      | 3.0   | 26   | 26   |
| 27.6              | 28                               | 28     | 28     | 28     | 28     | 28     | 28     | 28     | 28     | 28      | 28.0      | 1.4   | 28   | 28   |
| 28.00             | 28.00                            | 28.10  | 28.10  | 28.10  | 28.00  | 28.00  | 28.10  | 28.10  | 28.10  | 28.10   | 28.1      | 0.2   | 28.1 | 28   |
| 29.4              | 29                               | 29     | 29     | 29     | 29     | 29     | 29     | 29     | 29     | 29      | 29.0      | 1.4   | 29   | 29   |
| 30.2              | 30.8                             | 30.8   | 30.8   | 30.8   | 30.8   | 30.8   | 30.8   | 30.8   | 30.8   | 30.8    | 30.8      | 2.0   | 30.8 | 30.8 |
| 31.5              | 31                               | 31     | 31     | 31     | 31     | 31     | 31     | 31     | 31     | 31      | 31.0      | 1.6   | 31   | 31   |
| 32.1              | 32                               | 32     | 32     | 32     | 32     | 32     | 32     | 32     | 32     | 32      | 32.0      | 0.3   | 32   | 32   |
| Rata Rata Error : |                                  |        |        |        |        |        |        |        |        |         | 1.81      |       |      |      |

Tabel 4.9 menampilkan hasil pengujian sensor DHT11 dengan membandingkan pembacaan suhu sensor terhadap termometer ruangan sebagai alat acuan. Pengambilan data dilakukan sebanyak sepuluh kali untuk setiap nilai suhu yang diuji. Hasil pengujian menunjukkan bahwa sensor DHT11 memiliki tingkat akurasi yang cukup baik, dengan *error* maksimum sebesar 2,2 °C dan *error* minimum 0,3 °C. Nilai rata-rata *error* yang relatif kecil menandakan bahwa sensor ini mampu memberikan pembacaan suhu yang konsisten dan dapat diandalkan untuk pemantauan kondisi lingkungan pada sistem hidroponik.

#### 4.1.5. Data Hasil Pengujian PID

Pada pengujian PID ini dilakukan sebanyak 8 kali pengujian untuk mencari nilai parameter kontrol Kp, Ki, dan Kd yang optimal. Setiap pengujian dilakukan dengan variasi parameter yang berbeda untuk melihat respon sistem terhadap nilai *error* dan kestabilan *output*. Hasil dari pengujian ini akan menjadi acuan dalam menentukan konfigurasi akhir sistem kontrol *PID* yang digunakan pada proyek ini, seperti pada percobaan yang dilakukan di bawah ini.

#### 4.1.5.1. Percobaan Pertama



**Gambar 4. 4 Percobaan Pertama PID**

**Tabel 4. 10 Analisis Percobaan Pertama Pid**

| Respon Lup<br>Tertutup | <i>Rise Time</i> | <i>Overshoot</i> | <i>Settling time</i> | Steady-State<br>Error |
|------------------------|------------------|------------------|----------------------|-----------------------|
| KP= 0                  | -                | -                | -                    | - 98%                 |
| KI = 0                 |                  |                  |                      |                       |
| KD = 0                 |                  |                  |                      |                       |

Berdasarkan Grafik *step response* dari gambar 4.4 dan Tabel 4.10, pada kondisi nilai  $K_p$ ,  $K_i$ , dan  $K_d$  sama dengan nol menunjukkan bahwa sistem tidak memberikan respon terhadap *setpoint* yang ditetapkan, yaitu 500 ppm. Hal ini terlihat dari garis *output PPM* yang tetap konstan di sekitar 28,71 ppm tanpa adanya kenaikan menuju *setpoint*. Nilai PWM yang terbaca adalah 0, menandakan bahwa pompa tidak aktif sehingga tidak ada penambahan nutrisi ke dalam sistem. Berdasarkan tabel pengaruh parameter PID, tidak adanya nilai  $K_p$  menyebabkan sistem tidak mampu menurunkan *rise time*, sehingga respon menjadi sangat lambat bahkan tidak terjadi peningkatan sama sekali. Nilai  $K_i$  yang nol membuat *steady-state error* tidak dapat dieliminasi, sehingga selisih antara *output* dan *setpoint* tetap besar. Selain itu,  $K_d$  yang nol menghilangkan efek peredaman maupun percepatan respon sistem. Akibatnya, sistem berada dalam kondisi stagnan, tanpa *overshoot*, tanpa *settling time*, dan dengan *steady-state error* yang sangat tinggi.

#### 4.1.5.2. Percobaan Kedua



Gambar 4. 5 Percobaan Kedua PID

Tabel 4. 11 Analisis Percobaan Kedua Pid

| Respon Lup Tertutup | Rise Time | Overshoot | Settling time | Steady-State Error |
|---------------------|-----------|-----------|---------------|--------------------|
| KP= 1               | 0.5 detik | 50%       | 46 detik      | 10%                |
| KI = 0              |           |           |               |                    |
| KD = 0              |           |           |               |                    |

Berdasarkan Grafik *step response* dari gambar 4.5 dan Tabel 4.11, dengan pengaturan  $K_p = 1$ ,  $K_i = 0$ , dan  $K_d = 0$  menunjukkan bahwa sistem memberikan respon yang cepat namun menghasilkan *overshoot* yang cukup besar. Terlihat dari kurva *PPM output* yang awalnya berada di sekitar 230 ppm kemudian meningkat tajam melewati *setpoint* 500 ppm hingga mendekati 780 ppm. Hal ini sesuai dengan karakteristik aksi proporsional pada tabel panduan, di mana kenaikan nilai  $K_p$  akan menurunkan *rise time* sehingga respon sistem menjadi cepat, namun di sisi lain meningkatkan kemungkinan terjadinya *overshoot*. Karena nilai  $K_i$  diset 0, sistem tidak memiliki kemampuan untuk menghilangkan *steady-state error* sepenuhnya, sehingga nilai PPM akhirnya stabil di sekitar 433,90 ppm yang masih di bawah *setpoint*. Selain itu, tanpa adanya aksi derivatif ( $K_d = 0$ ), tidak ada peredaman terhadap lonjakan, sehingga respon awal menjadi lebih agresif sebelum akhirnya menurun dan cenderung stabil di titik yang tidak tepat.

#### 4.1.5.3. Percobaan Ketiga



Gambar 4. 6 Percobaan Ketiga PID

Tabel 4. 12 Analisis Percobaan Ketiga Pid

| Respon Lup Tertutup | Rise Time | Overshoot | Settling time | Steady-State Error |
|---------------------|-----------|-----------|---------------|--------------------|
| KP= 1               | 4 detik   | 34%       | 36 detik      | 2.09%              |
| KI = 0,02           |           |           |               |                    |
| KD = 0              |           |           |               |                    |

Berdasarkan Grafik *step response* dari gambar 4.6 dan Tabel 4.12, dengan pengaturan  $K_p = 1$ ,  $K_i = 0,02$ , dan  $K_d = 0$  menunjukkan adanya perbaikan pada pencapaian *setpoint* dibandingkan kondisi sebelumnya saat  $K_i = 0$ . Pada awal respon, *PPM output* yang semula sekitar 270 *ppm* meningkat tajam melewati *setpoint* 500 *ppm* hingga mencapai sekitar 680 *ppm*, menunjukkan masih adanya *overshoot* akibat aksi proporsional yang cukup besar. Namun, berbeda dengan kondisi tanpa integral, nilai *PPM* kemudian turun dan bergerak mendekati *setpoint*, stabil di kisaran 489,53 *ppm* yang lebih dekat dengan target. Hal ini sesuai dengan karakteristik aksi integral pada tabel panduan, yaitu membantu mengurangi *steady-state error* dengan menambahkan aksi kontrol berdasarkan akumulasi *error*. Meski begitu, karena nilai  $K_d$  masih 0, tidak ada efek peredaman terhadap lonjakan, sehingga *overshoot* tetap muncul di awal sebelum sistem menstabilkan diri.

#### 4.1.5.4. Percobaan Keempat



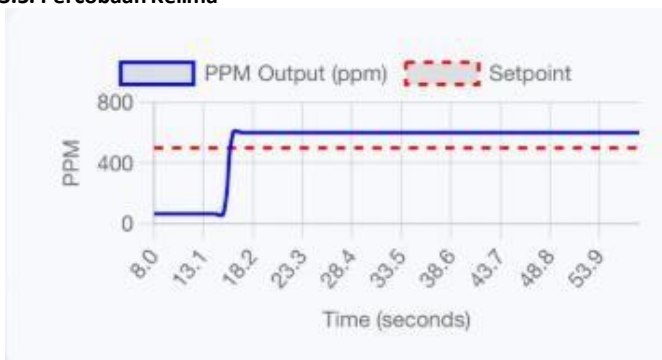
Gambar 4. 7 Percobaan Keempat PID

Tabel 4. 13 Analisis Percobaan Keempat Pid

| Respon Lup Tertutup | Rise Time | Overshoot | Settling time | Steady-State Error |
|---------------------|-----------|-----------|---------------|--------------------|
| KP= 1               | 7 detik   | 90%       | 225 detik     | 6%                 |
| KI = 0,02           |           |           |               |                    |
| KD = 0,1            |           |           |               |                    |

Berdasarkan Grafik *step response* dari gambar 4.7 dan Tabel 4.13, saat  $K_p = 1$ ,  $K_i = 0,02$ , dan  $K_d = 0,1$  menunjukkan kenaikan yang cepat (*rise time* rendah) disertai *overshoot* yang cukup tinggi hingga mendekati  $\pm 900 \text{ ppm}$ . Lonjakan ini muncul karena aksi proporsional yang kuat serta efek “*derivative kick*” saat terjadi perubahan *error* yang sangat cepat. Setelah puncak, komponen derivatif memberi peredaman sehingga kurva segera turun dan menjadi lebih stabil dengan osilasi yang kecil. Komponen integral yang relatif kecil terus mengurangi *error*, namun karena  $K_i$  masih rendah dan PWM dibatasi di 50, nilai akhir berhenti sedikit di bawah *setpoint* dan bertahan di sekitar 450–500 *ppm* (terbaca  $\pm 450,66 \text{ ppm}$ ). Secara ringkas: respon cepat, *overshoot* ada tetapi cepat teredam, *settling time* lebih singkat dibanding tanpa  $K_d$ , dan *steady-state error* mengecil meski belum nol.

#### 4.1.5.5. Percobaan Kelima



Gambar 4. 8 Percobaan Kelima PID

Tabel 4. 14 Analisis Percobaan Kelima Pid

| Respon Lup<br>Tertutup | Rise Time | Overshoot | Settling time | Steady-State<br>Error |
|------------------------|-----------|-----------|---------------|-----------------------|
| KP= 0,5                | 5 detik   | 10%       | 7 detik       | 9 %                   |
| KI = 0,1               |           |           |               |                       |
| KD = 0,05              |           |           |               |                       |

Berdasarkan Grafik *step response* dari gambar 4.8 dan Tabel 4.14, dengan  $K_p = 0,5$ ,  $K_i = 0,1$ , dan  $K_d = 0,05$  menunjukkan respon yang cepat menuju *setpoint* 500 ppm, diikuti *overshoot* yang relatif kecil–sedang (kurva naik sedikit di atas *setpoint* sekitar ~600 ppm) lalu mereda. Nilai  $K_p$  yang sedang mempercepat kenaikan (*rise time* menurun), sedangkan  $K_i$  membantu mengurangi *error* hingga keluaran mendekati *setpoint*. Penambahan  $K_d$  memberi peredaman sehingga lonjakan dan osilasi cepat berkurang dan waktu tunak menjadi lebih singkat. Hasil akhirnya, sistem stabil di sekitar target dengan *steady-state error* kecil.

#### 4.1.6. Pengujian sistem Secara *Real-Time* pada *Greenhouse*.

Tabel 4. 15 Pengujian sistem secara iReal-Time

| No | Waktu | Tampilan Secara <i>Real-time</i> Pada Aplikasi android,<br><i>Website</i> , dan LCD |              |               |              |                   |
|----|-------|-------------------------------------------------------------------------------------|--------------|---------------|--------------|-------------------|
|    |       | pH                                                                                  | TDS<br>(PPM) | Tinggi<br>Air | Suhu<br>(°C) | Kelembapan<br>(%) |

|    |       |     |      |      |    |    |
|----|-------|-----|------|------|----|----|
| 1  | 19.34 | 4.9 | 1328 | 36.1 | 40 | 29 |
| 2  | 19.40 | 3.1 | 1380 | 36.4 | 40 | 27 |
| 3  | 19.45 | 3.8 | 1382 | 36.4 | 40 | 26 |
| 4  | 19.50 | 3.6 | 1383 | 36.4 | 40 | 25 |
| 5  | 19.55 | 3.8 | 1383 | 36.4 | 40 | 24 |
| 6  | 20.00 | 3.9 | 1381 | 36.4 | 40 | 24 |
| 7  | 20.05 | 4   | 1380 | 36.4 | 40 | 24 |
| 8  | 20.12 | 4.2 | 1380 | 47.9 | 40 | 23 |
| 9  | 20.18 | 3.5 | 1381 | 49.6 | 40 | 23 |
| 10 | 20.24 | 3.9 | 1381 | 49.6 | 40 | 24 |

Pada Tabel 4.15 Pengujian sistem dilakukan untuk mengamati performa keseluruhan dari proses *monitoring* yang mencakup sensor pH, TDS (*ppm*), tinggi air, suhu (°C), dan kelembapan (%) dalam periode waktu tertentu. Berdasarkan data yang diperoleh dari pukul 19.34 hingga 20.24, sistem berhasil mencatat data dari seluruh sensor secara *real-time* dan konsisten. Nilai pH berkisar antara 3.1 hingga 4.9, menunjukkan bahwa larutan berada dalam kondisi asam, dengan fluktuasi yang masih wajar untuk sistem hidroponik setelah proses pemberian nutrisi.

Nilai TDS relatif stabil di sekitar 1380 ppm, menunjukkan bahwa kontrol nutrisi oleh sistem PID telah berhasil menjaga konsentrasi larutan mendekati *setpoint* yang diinginkan. Tinggi air sempat menunjukkan perubahan signifikan, dari 36.1 cm ke 49.6 cm, mengindikasikan adanya proses pengisian atau fluktuasi akibat penyesuaian sistem. Sensor suhu dan kelembapan juga mencatat pembacaan yang stabil; suhu berada antara 40°C, sedangkan kelembapan menurun dari 29% menjadi 23%.

Hal ini mencerminkan bahwa suhu ruang *greenhouse* cukup tinggi dan kelembapan relatif menurun seiring waktu. Secara keseluruhan, sistem berhasil melakukan akuisisi data dengan stabil dan responsif. Tidak ditemukan *error* pembacaan atau lonjakan abnormal, yang berarti sensor-sensor berfungsi optimal dan integrasi dengan *Firebase* berjalan dengan baik. Ini membuktikan bahwa sistem *monitoring* dan kontrol berbasis ESP32, logika PID, dan *Firebase* dapat bekerja secara efektif dalam lingkungan *greenhouse* di Politeknik Negeri Batam.

## Bab 5. Kesimpulan dan Saran

### 5.1. Kesimpulan

Berdasarkan Berdasarkan hasil perancangan, pengujian, dan analisis sistem monitoring serta kontrol nutrisi otomatis pada *greenhouse* menggunakan logika PID, maka kesimpulan yang menjawab tujuan penelitian ini adalah sebagai berikut:

1. Sistem berhasil melakukan akuisisi data multi-sensor yang terdiri dari pH, TDS, suhu, kelembapan, dan ketinggian air secara *real-time*, dengan akurasi yang baik berdasarkan hasil pengujian terhadap alat ukur perbandingan.
2. Penerapan metode kontrol PID pada sistem nutrisi mampu menjaga nilai TDS mendekati *setpoint* secara stabil dan responsif, tanpa terjadi lonjakan ekstrem, sehingga parameter nutrisi tetap terjaga sesuai kebutuhan tanaman.
3. Aplikasi mobile Android dan *website* yang dikembangkan dapat menampilkan data *real-time* serta memberikan fitur kontrol jarak jauh, sehingga pengguna dapat memantau dan mengatur sistem dari berbagai perangkat secara mudah.
4. Integrasi antara sistem sensor, kontrol PID, dan aplikasi *mobile* berbasis teknologi IoT melalui *Firebase Realtime Database* berjalan efektif, memastikan data dapat dikirim, disimpan, dan diakses secara sinkron.
5. Hasil pengujian kinerja menunjukkan bahwa sistem mampu mengontrol parameter nutrisi pada *greenhouse* hidroponik dengan baik, didukung oleh rancangan elektrik dan mekanik yang rapi, aman, dan sesuai untuk lingkungan *greenhouse*.

### 5.2. Saran

1. Penambahan sensor oksigen terlarut dan intensitas cahaya dapat dilakukan agar sistem dapat memantau lebih banyak aspek lingkungan tumbuh tanaman.
2. Sistem kontrol dapat dikembangkan lebih lanjut menggunakan metode *fuzzy logic* atau *machine learning* untuk meningkatkan presisi dalam pemberian nutrisi.
3. Diperlukan pengujian jangka panjang terhadap berbagai jenis tanaman untuk menguji keandalan sistem dalam berbagai skenario budidaya.

4. Aplikasi *mobile* dapat dikembangkan lebih lanjut dengan fitur notifikasi otomatis dan grafik riwayat data agar pengguna lebih cepat merespons perubahan kondisi sistem.
5. Perlu dilakukan pengemasan sistem secara lebih tahan terhadap lingkungan luar (debu, air, panas) agar sistem lebih awet untuk digunakan secara komersial.

## Daftar Pustaka

- [1] S. M. L. H. H. S. M. D. A. D. S. M. H. A. S. M. Sheli Mustikasari Dewi, Pertanian Budidaya dan Tanaman, Malang: PT. Literasi Nusantara Abadi Grup, 2024.
- [2] A. W. R. M. O. S. M. A. D. F. F. M. P. Z. A. H. Iqbal Lail Ramadhan, "Pengenalan Budidaya Tanaman Hidroponik Sederhana dengan Sistem Sumbu Menggunakan Barang Bekas," 2024.
- [3] C. N. Harsela, " Sistem Hidroponik Menggunakan Nutrient Film Technique untuk Produksi dan Hasil Tanaman Selada (*Lactuca sativa* L.)," *Jurnal Ilmiah Indonesia*, 2022.
- [4] T. A. H. A. Ariel Ramanda Sipayung, "Perancangan Sistem Monitoring dan Pengendalian Nutrisi pada Tanaman Hidroponik Sistem Nutrient Film Technique (NFT) Menggunakan Kontrol PID.," *Jurnal Ilmiah Teknik Elektro*, 2020.
- [5] U. L. Ririn Solekha, "Sistem Kendali Proportional Integral Derivative (PID) Menggunakan Mikrokontroler Arduino pada Thinkercad," *Jurnal Ilmiah Teknik Elektro*, vol. 5, 2024.
- [6] I. S. Roidah, A. W. Maromi, D. Maulana, M. F. Aghniya and R. P. Fadilah, "Penerapan Sistem Hidroponik NFT (Nutrient Film Technique) dan Irigasi Tetes pada Kelompok Wanita Tani Kirani di Kelurahan Pakunden, Kecamatan Sukorejo, Kota Blitar," *Jurnal Pengabdian Kepada Masyarakat*, 2022.
- [7] S. Nurrahma, T. A. Berbudi, M. R. Firdaus, G. Izzaulhaq and I. Hudati, "Implementasi Kontrol PID pada Kopel Motor DC dengan Menggunakan Filter Kalman," *Jurnal Listrik, Instrumentasi, dan Elektronika Terapan.*, 2023.
- [8] "Pertanian di Balik Layar: Cara IoT, AI, dan Blockchain Membawa Transparansi dalam Produksi Pangan," *bpm bkm*, 8 November 2024.
- [9] D. Setyanto and N. S. Salahuddin, "Prototipe Monitor dan Kontrol Otomatis Iklim Mikro Greenhouse dengan Platform IoT Blynk," *Techno.COM*, vol. 21, 2022.
- [10] H. Nashirudin and T. Pangaribowo, "Prototype Automasi Greenhouse Dengan Monitoring Menggunakan Smartphone dan ESP32 Berbasis Internet of Things," *Jurnal Teknologi Elektro*, vol. 13, 2022.

- [11] A. D. Agustya, R. Handayani and M. I. Sani, "Sistem Kendali dan Monitoring Lingkungan Rumah Kaca dengan Metode IoT," *e-Proceeding of Applied Science*, vol. 5, 2019.

## Biodata



Nama : Jelita Agnesia Purba  
TTL : Batam, 22 September 2004  
Agama : Kristen Protestan  
Alamat : Bida Ayu Blok H No.125  
  
Email : jelitaagnesia2209@gmail.com  
Riwayat Pendidikan : SMA/SMK : SMA Negeri 16 Batam  
SMP : SMP Negeri 16 Batam



Nama : Muhammad Reza  
TTL : Batam, 09 April 2004  
Agama : Islam  
Alamat : Kavling Sagulung Baru Blok O No.115,  
kec. Sagulung, Kel. Sei Binti  
Email : muhammadrezaa9404@gmail.com  
Riwayat Pendidikan : SMA/SMK : SMK Negeri 1 Batam  
SMP : SMP Negeri 50 Batam



Nama : Muhammad Baihaqi Adnan  
TTL : Batam, 10 Januari 2005  
Agama : Islam  
Alamat : Perum Mantang Blok P No.5  
  
Email : haqithax@gmail.com  
Riwayat Pendidikan : SMA/SMK : SMK IT DARUSSALAM  
SMP : SWTQIS

# Lampiran

## Lampiran A. Program Sistem

```
Program :
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <DHT11.h>
#include <WiFi.h>
#include <HttpClient.h>

// ===== PIN DEFINITIONS =====
#define TDS_PIN 34 // Pin sensor TDS (GPIO34 - ADC1)
#define MOTOR_IN1 25 // Pin IN1 motor L298N
#define MOTOR_IN2 26 // Pin IN2 motor L298N
#define ENABLE_PIN 27 // Pin Enable motor L298N
#define TRIG_PIN 5 // Pin Trig sensor ultrasonik
#define ECHO_PIN 18 // Pin Echo sensor ultrasonik
#define PH_PIN 35 // Pin sensor pH (GPIO35 - ADC1)
#define DHT_PIN 4 // Pin sensor DHT11

// ===== CONSTANTS =====
#define JARAK_MAX 50.0 // Tinggi maksimum air (cm)
#define SAMPLES 30 // Jumlah sampel untuk median filter
#define WIFI_SSID "Polibatam" // Nama WiFi
#define WIFI_PASSWORD "" // Password WiFi

// Firebase configuration (HTTP REST API)
#define FIREBASE_HOST "https://hqchulo-default-rtdb.asia-southeast1.firebaseio.com/"
#define FIREBASE_AUTH "AlzaSyAL3l3Fb9_Q1CAkjwxEBvG6Tb1W-HJwUCI" //
Replace with your actual key

// ===== PID VARIABLES =====
double Kp = 0.5, Ki = 0.01, Kd = 0.05; // Default PID values
double setpoint = 500.0; // Target TDS (ppm)
double input, output; // Sensor input & PID output
double error, lastError; // PID error variables
double integral, derivative; // Integral & derivative terms
double outputMin = 50.0; // Minimum motor output (≈20%)
double outputMax = 255.0; // Maximum motor output (100%)
```

```

int tdsValues[SAMPLES]; // Buffer for median filter
String inputString = ""; // Serial input string
bool newData = false; // New serial data flag

// ===== TIMING VARIABLES =====
unsigned long sendDataPrevMillis = 0;
unsigned long lcdUpdateTime = 0;
unsigned long pidDisplayTime = 0;
unsigned long firebaseUpdateTime = 0;

// ===== SENSOR DATA =====
float currentTDS = 0;
float currentPH = 0;
float currentTinggi = 0;
float currentSuhu = 0;
float currentKelembaban = 0;

// ===== SENSOR & LCD INITIALIZATION =====
DHT11 dht11(DHT_PIN);
LiquidCrystal_I2C lcd(0x27, 20, 4); // I2C LCD 20x4 address

// ===== FUNCTION PROTOTYPES =====
float getMedianNum(int bArray[], int iFilterLen);
double computePID(double inp);
void setMotorSpeed(int speed);
void updateLCD(float tds, float ph, float tinggi, float suhu, float kelembaban);
void sendDataToFirebase(float tds, float ph, float tinggi, float suhu, float
kelembaban);
void updateParametersFromFirebase();
void parseSerialInput();
void printHelp();
void displayPIDInfo();
void readAllSensors();
String createJsonData(float tds, float ph, float tinggi, float suhu, float
kelembaban);
void parseFirebaseResponse(String response);
float measureDistance(); // Fungsi baru untuk mengukur jarak

// ===== SETUP FUNCTION =====
void setup() {
  Serial.begin(115200);

```

```

delay(1000);

Serial.println("=====");
Serial.println("  TDS CONTROL SYSTEM WITH PID");
Serial.println("    ESP32 Compatible v2.0");
Serial.println("=====");

// Initialize pins
pinMode(TRIG_PIN, OUTPUT);
pinMode(ECHO_PIN, INPUT);
pinMode(MOTOR_IN1, OUTPUT);
pinMode(MOTOR_IN2, OUTPUT);
pinMode(ENABLE_PIN, OUTPUT);

// Set ADC configuration for ESP32
analogReadResolution(12); // 12-bit ADC (0-4095)
analogSetAttenuation(ADC_11db); // For 0-3.3V input

// Motor off at startup
digitalWrite(MOTOR_IN1, LOW);
digitalWrite(MOTOR_IN2, LOW);
analogWrite(ENABLE_PIN, 0);
Serial.println("Motor initialized: OFF");

// Initialize DHT11 sensor
Serial.println("DHT11 sensor initialized");
delay(2000);

// Initialize LCD
lcd.init();
lcd.backlight();
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("TDS Control System");
lcd.setCursor(0, 1);
lcd.print("ESP32 v2.0");
lcd.setCursor(0, 2);
lcd.print("Booting...");
Serial.println("LCD initialized");

delay(2000);

```

```

// WiFi connection
WiFi.mode(WIFI_STA);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to WiFi: ");
Serial.println(WIFI_SSID);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connecting WiFi...");
lcd.setCursor(0, 1);
lcd.print(WIFI_SSID);

int wifiAttempts = 0;
while (WiFi.status() != WL_CONNECTED && wifiAttempts < 40) {
    delay(500);
    Serial.print(".");
    lcd.setCursor(wifiAttempts % 20, 2);
    lcd.print(".");
    wifiAttempts++;
}

Serial.println();

if (WiFi.status() == WL_CONNECTED) {
    Serial.println("WiFi Connected Successfully!");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP().toString());
    Serial.print("Signal Strength: ");
    Serial.print(WiFi.RSSI());
    Serial.println(" dBm");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("WiFi Connected!");
    lcd.setCursor(0, 1);
    lcd.print("IP: ");
    lcd.print(WiFi.localIP().toString());
    lcd.setCursor(0, 2);
    lcd.print("Signal: ");
    lcd.print(WiFi.RSSI());
    lcd.print(" dBm");
}

```

```

    } else {
        Serial.println("WiFi Connection Failed!");
        Serial.println("Running in offline mode...");

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("WiFi Failed!");
        lcd.setCursor(0, 1);
        lcd.print("Offline Mode");
        lcd.setCursor(0, 2);
        lcd.print("Check credentials");
    }

    delay(3000);
    lcd.clear();

    // Test sensor readings
    Serial.println("Testing sensors...");
    readAllSensors();

    Serial.println("=====");
    Serial.println("System Ready!");
    Serial.println("Type 'HELP' for available commands");
    Serial.println("=====");

    // Show help menu
    printHelp();
}

// ===== MAIN LOOP =====
void loop() {
    // Read all sensors
    readAllSensors();

    // Calculate PID output
    output = computePID(currentTDS);
    setMotorSpeed((int)output);

    // Display data on Serial Monitor
    Serial.print("TDS: "); Serial.print(currentTDS, 2); Serial.print(" ppm");
    Serial.print(" | pH: "); Serial.print(currentPH, 2);
    Serial.print(" | H: "); Serial.print(currentTinggi, 2); Serial.print(" cm");

```

```

    Serial.print(" | T: "); Serial.print(currentSuhu, 1); Serial.print("°C");
    Serial.print(" | RH: "); Serial.print(currentKelembaban, 1); Serial.print("%");
    Serial.print(" | Motor: "); Serial.print(map(output, 0, 255, 0, 100));
    Serial.println("");

    // Update parameters from Firebase every 30 seconds
    if (WiFi.status() == WL_CONNECTED && millis() - firebaseUpdateTime >=
30000) {
        updateParametersFromFirebase();
        firebaseUpdateTime = millis();
    }

    // Send data to Firebase every 15 seconds
    if (WiFi.status() == WL_CONNECTED && millis() - sendDataPrevMillis >=
15000) {
        sendDataToFirebase(currentTDS, currentPH, currentTinggi, currentSuhu,
currentKelembaban);
        sendDataPrevMillis = millis();
    }

    // Update LCD every 2 seconds
    if (millis() - lcdUpdateTime >= 2000) {
        updateLCD(currentTDS, currentPH, currentTinggi, currentSuhu,
currentKelembaban);
        lcdUpdateTime = millis();
    }

    // Check serial input
    if (Serial.available()) {
        inputString = Serial.readStringUntil('\n');
        inputString.trim();
        if (inputString.length() > 0) {
            newData = true;
        }
    }

    if (newData) {
        parseSerialInput();
        newData = false;
    }

    // Display PID info every 10 seconds

```

```

    if (millis() - pidDisplayTime >= 10000) {
        displayPIDInfo();
        pidDisplayTime = millis();
    }

    delay(100);
}

// ===== SENSOR READING FUNCTIONS
=====
void readAllSensors() {
    // Read TDS sensor with median filter
    for (int i = 0; i < SAMPLES; i++) {
        tdsValues[i] = analogRead(TDS_PIN);
        delay(10);
    }

    // Convert to TDS (ppm)
    int medianValue = getMedianNum(tdsValues, SAMPLES);
    float voltage = medianValue * (3.3 / 4095.0);
    currentTDS = (-75.21 * pow(voltage, 2) + 745.09 * voltage + 71.339);
    if (currentTDS < 0) currentTDS = 0;
    input = currentTDS;

    // Read pH sensor
    int analogValue = analogRead(PH_PIN);
    float phvoltage = analogValue * 3.3 / 4095.0;
    currentPH = 1.1191 * phvoltage * phvoltage - 10.171 * phvoltage + 20.712;
    currentPH = constrain(currentPH, 0, 14);

    // Read ultrasonic sensor (water level)
    currentTinggi = JARAK_MAX - measureDistance();
    currentTinggi = constrain(currentTinggi, 0, JARAK_MAX); // Pastikan tidak
negatif

    // Read DHT11 sensor with error handling
    int temperature = 0;
    int humidity = 0;
    int result = dht11.readTemperatureHumidity(temperature, humidity);

    if (result == 0) {
        currentSuhu = temperature;

```

```

        currentKelembaban = humidity;
    } else {
        Serial.print("DHT11 Error: ");
        Serial.println(DHT11::getErrorString(result));
        currentSuhu = 0;
        currentKelembaban = 0;
    }
}

// ===== ULTRASONIC SENSOR FUNCTIONS =====
float measureDistance() {
    // Membersihkan trigger pin
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);

    // Mengirim sinyal trigger selama 10 mikrodetik
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    // Membaca durasi echo pin dengan timeout 30ms (jarak maks ~5m)
    long duration = pulseIn(ECHO_PIN, HIGH, 30000);

    // Jika timeout terjadi, kembalikan jarak maksimum
    if (duration == 0) {
        return JARAK_MAX;
    }

    // Menghitung jarak dalam cm (kecepatan suara 0.0343 cm/μs)
    float distance = duration * 0.0343 / 2;

    // Kalibrasi sensor (sesuaikan dengan kebutuhan)
    distance *= 0.951;

    return distance;
}

// ===== PID FUNCTIONS =====
double computePID(double inp) {
    error = setpoint - inp;

```

```

// Proportional term
double Pout = Kp * error;

// Derivative term
derivative = error - lastError;
double Dout = Kd * derivative;

// Integral term with anti-windup
integral += error;
double maxIntegral = 1000.0;
integral = constrain(integral, -maxIntegral, maxIntegral);
double Iout = Ki * integral;

// Total output
double pidOutput = Pout + Iout + Dout;

// Dead zone to reduce oscillation
if (abs(error) < 5) {
    pidOutput = outputMin;
    integral *= 0.95;
}

// Constrain output
pidOutput = constrain(pidOutput, 0, outputMax);

// Minimum speed check
if (pidOutput > 0 && pidOutput < outputMin) {
    pidOutput = outputMin;
}

lastError = error;
return pidOutput;
}

```

```

// ===== MOTOR CONTROL FUNCTIONS
=====
void setMotorSpeed(int speed) {
    speed = constrain(speed, 0, 255);

    if (speed < 50) {
        // Motor off
        digitalWrite(MOTOR_IN1, LOW);
    }
}

```

```

        digitalWrite(MOTOR_IN2, LOW);
        analogWrite(ENABLE_PIN, 0);
    } else {
        // Motor on (forward direction to add nutrients)
        digitalWrite(MOTOR_IN1, HIGH);
        digitalWrite(MOTOR_IN2, LOW);
        analogWrite(ENABLE_PIN, speed);
    }
}

// ===== MEDIAN FILTER FUNCTION
=====
float getMedianNum(int bArray[], int iFilterLen) {
    int bTab[iFilterLen];
    for (int i = 0; i < iFilterLen; i++) bTab[i] = bArray[i];

    // Bubble sort
    for (int j = 0; j < iFilterLen - 1; j++) {
        for (int i = 0; i < iFilterLen - j - 1; i++) {
            if (bTab[i] > bTab[i + 1]) {
                int temp = bTab[i];
                bTab[i] = bTab[i + 1];
                bTab[i + 1] = temp;
            }
        }
    }

    return (iFilterLen % 2 == 0) ?
        (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2.0 :
        bTab[iFilterLen / 2];
}

// ===== LCD FUNCTIONS =====
void updateLCD(float tds, float ph, float tinggi, float suhu, float kelembaban) {
    lcd.clear();

    // Line 1: TDS and pH
    lcd.setCursor(0, 0);
    lcd.print("TDS:");
    lcd.print(tds, 1);
    lcd.print(" pH:");
    lcd.print(ph, 1);

```

```

// Line 2: Water level and WiFi status
lcd.setCursor(0, 1);
lcd.print("H:");
lcd.print(tinggi, 1);
lcd.print("cm ");
if (WiFi.status() == WL_CONNECTED) {
    lcd.print("WiFi:OK");
} else {
    lcd.print("WiFi:--");
}

// Line 3: Temperature and Humidity
lcd.setCursor(0, 2);
lcd.print("T:");
lcd.print(suhu, 1);
lcd.print("C RH:");
lcd.print(kelembaban, 1);
lcd.print("%");

// Line 4: PID Status
lcd.setCursor(0, 3);
lcd.print("PID:");
lcd.print(map(output, 0, 255, 0, 100));
lcd.print("% SP:");
lcd.print(setpoint, 0);
}

// ===== FIREBASE FUNCTIONS =====
String createJsonData(float tds, float ph, float tinggi, float suhu, float
kelembaban) {
    String json = "{";
    json += "\"tds\":" + String(tds) + ",";
    json += "\"ph\":" + String(ph) + ",";
    json += "\"tinggi_air\":" + String(tinggi) + ",";
    json += "\"suhu\":" + String(suhu) + ",";
    json += "\"kelembaban\":" + String(kelembaban) + ",";
    json += "\"output_pid\":" + String(output) + ",";
    json += "\"setpoint\":" + String(setpoint) + ",";
    json += "\"timestamp\":" + String(millis()) + ",";
    json += "\"wifi_rssi\":" + String(WiFi.RSSI());
    json += "}";

```

```

        return json;
    }

    void sendDataToFirebase(float tds, float ph, float tinggi, float suhu, float
kelembaban) {
        if (WiFi.status() == WL_CONNECTED) {
            HTTPClient http;

            String jsonData = createJsonData(tds, ph, tinggi, suhu, kelembaban);
            String url = String(FIREBASE_HOST) + "/HQ/SENSOR.json?auth=" +
FIREBASE_AUTH;

            http.begin(url);
            http.addHeader("Content-Type", "application/json");

            int httpResponseCode = http.PUT(jsonData);

            if (httpResponseCode > 0) {
                String response = http.getString();
                Serial.println("✓ Data sent to Firebase successfully!");
                Serial.println("Response code: " + String(httpResponseCode));
            } else {
                Serial.println("X Error sending data to Firebase");
                Serial.println("HTTP Error: " + String(httpResponseCode));
            }

            http.end();
        } else {
            Serial.println("X WiFi not connected - cannot send data");
        }
    }

    void parseFirebaseResponse(String response) {
        // Simple parsing for JSON response
        int setpointPos = response.indexOf("\"setpoint\":");
        if (setpointPos != -1) {
            int startPos = setpointPos + 10;
            int endPos = response.indexOf(",", startPos);
            if (endPos == -1) endPos = response.indexOf("}", startPos);
            if (endPos != -1) {
                float newSetpoint = response.substring(startPos, endPos).toFloat();
                if (newSetpoint > 0 && newSetpoint != setpoint) {

```

```

        setpoint = newSetpoint;
        integral = 0;
        Serial.println("✓ Setpoint updated from Firebase: " + String(setpoint));
    }
}
}

```

```

int kpPos = response.indexOf("\\"kp\\":");
if (kpPos != -1) {
    int startPos = kpPos + 5;
    int endPos = response.indexOf(",", startPos);
    if (endPos == -1) endPos = response.indexOf("}", startPos);
    if (endPos != -1) {
        float newKp = response.substring(startPos, endPos).toFloat();
        if (newKp >= 0 && newKp != Kp) {
            Kp = newKp;
            Serial.println("✓ Kp updated from Firebase: " + String(Kp, 4));
        }
    }
}
}

```

```

int kiPos = response.indexOf("\\"ki\\":");
if (kiPos != -1) {
    int startPos = kiPos + 5;
    int endPos = response.indexOf(",", startPos);
    if (endPos == -1) endPos = response.indexOf("}", startPos);
    if (endPos != -1) {
        float newKi = response.substring(startPos, endPos).toFloat();
        if (newKi >= 0 && newKi != Ki) {
            Ki = newKi;
            integral = 0;
            Serial.println("✓ Ki updated from Firebase: " + String(Ki, 4));
        }
    }
}
}

```

```

int kdPos = response.indexOf("\\"kd\\":");
if (kdPos != -1) {
    int startPos = kdPos + 5;
    int endPos = response.indexOf(",", startPos);
    if (endPos == -1) endPos = response.indexOf("}", startPos);
    if (endPos != -1) {

```

```

        float newKd = response.substring(startPos, endPos).toFloat();
        if (newKd >= 0 && newKd != Kd) {
            Kd = newKd;
            Serial.println("✓ Kd updated from Firebase: " + String(Kd, 4));
        }
    }
}

void updateParametersFromFirebase() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        String url = String(FIREBASE_HOST) + "/HQ/CONTROL.json?auth=" +
FIREBASE_AUTH;
        http.begin(url);

        int httpResponseCode = http.GET();

        if (httpResponseCode > 0) {
            String payload = http.getString();
            parseFirebaseResponse(payload);
        } else {
            Serial.println("X Failed to get Firebase parameters: " +
String(httpResponseCode));
        }

        http.end();
    }
}

// ===== SERIAL COMMAND FUNCTIONS
=====

void parseSerialInput() {
    inputString.trim();
    inputString.toUpperCase();

    if (inputString.startsWith("SP ")) {
        float newSetpoint = inputString.substring(3).toFloat();
        if (newSetpoint > 0 && newSetpoint <= 2000) {
            setpoint = newSetpoint;
            integral = 0;
        }
    }
}

```

```

Serial.println("✓ Setpoint changed to: " + String(setpoint) + " ppm");

// Update Firebase
if (WiFi.status() == WL_CONNECTED) {
  HTTPClient http;
  String url = String(FIREBASE_HOST) +
"/HQ/CONTROL/setpoint.json?auth=" + FIREBASE_AUTH;
  http.begin(url);
  http.addHeader("Content-Type", "application/json");
  int code = http.PUT(String(setpoint));
  if (code > 0) {
    Serial.println("✓ Setpoint updated in Firebase");
  }
  http.end();
} else {
  Serial.println("X Invalid setpoint! Use range 1-2000 ppm");
}
}

else if (inputString.startsWith("KP ")) {
  float newKp = inputString.substring(3).toFloat();
  if (newKp >= 0 && newKp <= 10) {
    Kp = newKp;
    Serial.println("✓ Kp changed to: " + String(Kp, 4));

    if (WiFi.status() == WL_CONNECTED) {
      HTTPClient http;
      String url = String(FIREBASE_HOST) + "/HQ/CONTROL/kp.json?auth=" +
FIREBASE_AUTH;
      http.begin(url);
      http.addHeader("Content-Type", "application/json");
      http.PUT(String(Kp));
      http.end();
    }
  } else {
    Serial.println("X Invalid Kp! Use range 0-10");
  }
}

else if (inputString.startsWith("KI ")) {
  float newKi = inputString.substring(3).toFloat();
  if (newKi >= 0 && newKi <= 1) {
    Ki = newKi;

```

```

    integral = 0;
    Serial.println("✓ Ki changed to: " + String(Ki, 4));

    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = String(FIREBASE_HOST) + "/HQ/CONTROL/ki.json?auth=" +
FIREBASE_AUTH;
        http.begin(url);
        http.addHeader("Content-Type", "application/json");
        http.PUT(String(Ki));
        http.end();
    }
    } else {
        Serial.println("X Invalid Ki! Use range 0-1");
    }
}

else if (inputString.startsWith("KD ")) {
    float newKd = inputString.substring(3).toFloat();
    if (newKd >= 0 && newKd <= 1) {
        Kd = newKd;
        Serial.println("✓ Kd changed to: " + String(Kd, 4));

        if (WiFi.status() == WL_CONNECTED) {
            HTTPClient http;
            String url = String(FIREBASE_HOST) + "/HQ/CONTROL/kd.json?auth=" +
FIREBASE_AUTH;
            http.begin(url);
            http.addHeader("Content-Type", "application/json");
            http.PUT(String(Kd));
            http.end();
        }
        } else {
            Serial.println("X Invalid Kd! Use range 0-1");
        }
    }
}

else if (inputString.startsWith("PID")) {
    Serial.println("\n=====
");
    Serial.println("|| PID PARAMETERS ||");

```

```

Serial.println("===== ");
    Serial.println("Setpoint: " + String(setpoint) + " ppm");
    Serial.println("Kp: " + String(Kp, 4));
    Serial.println("Ki: " + String(Ki, 4));
    Serial.println("Kd: " + String(Kd, 4));
    Serial.println("Current TDS: " + String(currentTDS, 2) + " ppm");
    Serial.println("Error: " + String(error, 2));
    Serial.println("Output: " + String(output) + " (" + String(map(output, 0,
255, 0, 100)) + "%)");

Serial.println("===== ");
    }
    else if (inputString.startsWith("SENSOR")) {

Serial.println("\n===== ");
Serial.println("      SENSOR READINGS      ");

Serial.println("===== ");
    Serial.println("TDS: " + String(currentTDS, 2) + " ppm");
    Serial.println("pH: " + String(currentPH, 2));
    Serial.println("Water Level: " + String(currentTinggi, 2) + " cm");
    Serial.println("Temperature: " + String(currentSuhu, 1) + " °C");
    Serial.println("Humidity: " + String(currentKelembaban, 1) + " %");

Serial.println("===== ");
    }
    else if (inputString.startsWith("STATUS")) {

Serial.println("\n===== ");
Serial.println("      SYSTEM STATUS      ");

Serial.println("===== ");
    Serial.println("WiFi: " + String(WiFi.status() == WL_CONNECTED ?
"Connected" : "Disconnected"));

```



```

        Serial.println("X Unknown command! Type 'HELP' for available
commands.");
    }

    inputString = "";
}

void printHelp() {

Serial.println("\n=====
");
    Serial.println("COMMAND LIST");

Serial.println("===== ");
    Serial.println("SP [value] - Set setpoint ");
    Serial.println("    (e.g., SP 500) ");
    Serial.println("KP [value] - Set Kp (e.g., KP 0.5) ");
    Serial.println("KI [value] - Set Ki (e.g., KI 0.01) ");
    Serial.println("KD [value] - Set Kd (e.g., KD 0.05) ");
    Serial.println("PID - Show PID status ");
    Serial.println("SENSOR - Show sensor values ");
    Serial.println("STATUS - Show system status ");
    Serial.println("RESET - Reset system ");
    Serial.println("MOTOR [0-100] - Set motor speed ");
    Serial.println("HELP - Show this menu ");
    Serial.println("TEST - Test all sensors ");

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

void displayPIDInfo() {
    Serial.print("PID Info - ");
    Serial.print("Setpoint: "); Serial.print(setpoint);
    Serial.print(" | Kp: "); Serial.print(Kp, 4);
    Serial.print(" | Ki: "); Serial.print(Ki, 4);
    Serial.print(" | Kd: "); Serial.print(Kd, 4);
    Serial.print(" | Error: "); Serial.print(error, 2);
    Serial.print(" | Output: "); Serial.print(output);
    Serial.print(" ("); Serial.print(map(output, 0, 255, 0, 100));
Serial.println("%)");
}

```

}

## Lampiran B. Dokumentasi Pengujian

