



**Alat *Monitoring* Air dengan Menggunakan Jalur  
Komunikasi Data RS-232**

**Proyek Akhir**

**Oleh:  
Ferdito Nois Firmansyah(3232101060)**

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

### Pernyataan Keaslian Proyek Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir yang berjudul : "Alat *Monitoring* Air dengan Menggunakan Jalur Komunikasi Data *RS-232*" 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.

Batam,



---

Nama : Ferdito Nois Firmansyah  
NIM : 3232101060

## Lembar Pengesahan

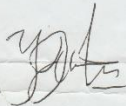
Proyek Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar  
Ahli Madya Teknik (AMd.T.)  
di  
Politeknik Negeri Batam

Oleh:  
Ferdito Nois Firmansyah (3232101060)

Tanggal Sidang: 10,01, 2024

Disetujui oleh :

Penguji I



Ir. Kamarudin, S.T., M.T., IPM  
NIK: 110071

Pembimbing



Ir. Muhammad Syafel Gozali, S.T., M.T.  
NIK: 107050

Penguji II



Afrizal Deri Futra S.Si., M.Si.  
NIK: 115133

# Alat *Monitoring* Air dengan Menggunakan Jalur Komunikasi Data *RS-232*

## Abstrak

Terpenuhinya kebutuhan akan air yang memiliki kualitas yang baik, menyebabkan dibutuhkan alat yang dapat memantau kualitas air. Pada penelitian ini kami mencoba membuat alat yang dapat memantau kualitas air dengan empat komunikasi data yang berbeda. Parameter yang digunakan pada penelitian ini adalah tingkat keasaman, kebasaan, dan kekeruhan suatu air. Pengolahan data dilakukan pada Arduino UNO, dengan menggunakan sensor *pH* seri E-201, dan sensor *turbidity* SEN0189. Range dari sensor *pH* mampu membaca tingkat keasaman, dan kebasaan mulai dari *pH* 0 – 14. Range pembacaan sensor *turbidity* mampu membaca mulai dari 0 – 1000 *NTU*. Komunikasi data yang digunakan pada penelitian ini ada empat yaitu *RS-232*, *RS-485*, *CANBus*, dan *TCP/IP*. Hasil pengujian dari kedua sensor yang digunakan, sensor *pH* mampu memiliki tingkat persentase *error* yang cukup rendah jika dibandingkan dengan hasil pengujian dari nilai sensor *turbidity*. Penelitian ini memiliki hasil akhir yaitu pembacaan data sensor menggunakan multi komunikasi data mampu tampil pada aplikasi yang telah dibuat.

Kata kunci : Sensor *pH*, sensor *turbidity*, Arduino UNO, Komunikasi data

## ***Water Monitoring Device with Data Communication Lines RS-232***

### ***Abstract***

*The need for good quality water is met, resulting in the need for tools that can monitor water quality. In this research we try to create a tool that can monitor water quality with four different data communications. The parameters used in this research are the level of acidity, alkalinity and turbidity of water. Data processing was carried out on Arduino UNO, using the E-201 series pH sensor and the SEN0189 turbidity sensor. The range of the pH sensor is capable of reading acidity and alkalinity levels ranging from pH 0 – 14. The reading range of the turbidity sensor is capable of reading from 0 – 1000 NTU. There are four data communications used in this research, namely Serial 232, RS-485, CANBus, and TCP/IP. The test results of the two sensors used showed that the pH sensor was able to have a fairly low percentage error rate when compared with the test results of the turbidity sensor value. This research has the final result, namely reading sensor data using multiple data communications which can appear in the application that has been created.*

*Keywords : pH sensor, turbidity sensor, Arduino UNO, data communication*

## Kata Pengantar

Puji dan syukur yang tidak terhingga penulis hanturkan kepada Tuhan Yang Maha Esa atas rahmat dan karuniannya sehingga penulis dapat menyelesaikan laporan Proyek Akhir yang berjudul “Alat *Monitoring* Air dengan Menggunakan Jalur Komunikasi Data RS-232”. Penulis menyelesaikan laporan Tugas Akhir untuk memenuhi salah satu persyaratan guna memperoleh gelar Ahli Madya Teknik (Amd.T.) kelulusan di Politeknik Negeri Batam Jurusan Teknik Elektro Program Studi Teknik Instrumentasi. Penulisan Laporan Proyek Akhir ini dapat disusun dan diselesaikan dengan baik berkat dukungan dan bantuan dosen, rekan mahasiswa dan dukungan dari banyak pihak yang ikut dalam membimbing penulis untuk menyelesaikan Laporan Proyek Akhir ini. Pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Tuhan Yang Maha Esa, yang selalu memberkati dalam keadaan terbaik hingga keadaan terburuk penulis dalam mengerjakan Tugas akhir ini.
2. Kepada kedua orang tua, yang selalu memberikan dukungan, kasih sayang, dan kebutuhan hidup kepada penulis.
3. Bapak Ir. Kamarudin, S.T.M.T.,IPM selaku Kepala Program Studi Jurusan Teknik Instrumentasi.
4. Bapak Ir. Muhammad Syafei Gozali S.T. M.T, selaku dosen pembimbing Tugas Akhir.
5. Seluruh Bapak/Ibu Dosen Teknik Elektro dan Seluruh teman-teman Instrumentasi Politeknik Negeri Batam yang telah membantu.

Batam,05-08-2024



**Ferdito Nois Firmansyah**

## Daftar Isi

Pernyataan Keaslian Proyek Akhir .....	<b>Error! Bookmark not defined.</b>
Abstrak .....	iii
Abstract .....	iv
Kata Pengantar .....	v
Daftar Isi .....	vi
Daftar Tabel .....	vii
Daftar Gambar .....	viii
Daftar Lampiran .....	ix
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 .....	2
1.6 Struktur Pembagian Kerja .....	3
Bab 2. Tinjauan Pustaka .....	4
Bab 3. Metode Penelitian .....	14
3.1 Perancangan Elektrikal .....	15
3.2 Perancangan Mekanikal .....	16
3.3 Perancangan <i>Software</i> .....	18
3.4 Alat dan Bahan .....	24
3.5 Pengujian .....	25
Bab 4. Hasil dan Pembahasan .....	27
Bab 5. Kesimpulan dan Saran .....	34
Daftar Pustaka .....	35
Biodata .....	36
Lampiran .....	37

## Daftar Tabel

Tabel 1 Struktur Pembagian Kerja.....	3
Table 2 Spesifikasi sensor pH.....	5
Table 3 Spesifikasi Sensor Turbidity.....	6
Table 4 Spesifikasi Arduino uno.....	8
Table 5 Spesifikasi RS-232 shield.....	9
Table 6 Spesifikasi RS-485 shield.....	9
Tabel 7 Spesifikasi CANBus shield.....	11
Tabel 8 Spesifikasi ethernet shield.....	12
Tabel 9 Spesifikasi LCD I2C 20x4.....	13
Tabel 10 Wiring Schedule komunikasi.....	15
Tabel 11 Wiring Schedule sensor dan komponen pelengkap.....	16
Tabel 12 Alat dan Bahan.....	24
Tabel 13 Pengujian Sensor pH.....	27
Tabel 14 Pengujian Sensor Turbidity sebelum menggunakan persamaan linear.....	28
Table 15 Pengambilan data turbidity setelah kalibrasi.....	30

## Daftar Gambar

Gambar 1 Sensor pH .....	5
Gambar 2 Sensor Turbidity .....	6
Gambar 3 Arduino uno .....	7
Gambar 4 RS-232 shield .....	8
Gambar 5 RS-485 shield .....	9
Gambar 6 CANBus shield .....	11
Gambar 7 Ethernet shield .....	11
Gambar 8 LCD I2C 20x4 .....	12
Gambar 9 Blok Diagram Sistem .....	14
Gambar 10 Perancangan Elektrikal .....	15
Gambar 11 Perancangan Mekanikal tampak luar .....	16
Gambar 12 Perancangan Mekanikal tampak luar .....	17
Gambar 13 Perancangan Mekanikal tampak dalam .....	17
Gambar 14 Perancangan Software .....	18
Gambar 15 Perancangan Software untuk komunikasi CANBus .....	19
Gambar 16 Perancangan Software untuk Komunikasi TCP/IP .....	20
Gambar 17 Flowchart program turbidity .....	21
Gambar 18 Flowchart program pH .....	22
Gambar 19 Flowchart program sistem .....	23
Gambar 20 Grafik Akuisisi Data Sensor pH .....	28
Gambar 21 Grafik Persamaan linearitas turbidity .....	29
Gambar 22 Tampilan pada C# menggunakan komunikasi CANBus .....	31
Gambar 23 Tampilan Web Menggunakan Komunikasi TCP/IP .....	32
Gambar 24 Tampilan Komunikasi Serial 232 .....	32
Gambar 25 Tampilan Komunikasi RS-485 .....	33

## Daftar Lampiran

Lampiran 1 Tampilan bagian dalam .....	37
Lampiran 2 Tampilan Box yang sudah dirakit.....	37
Lampiran 3 Data pengujian sensor pH (sample pH 1) .....	48
Lampiran 4 Data pengujian sensor pH (sample pH 2) .....	48
Lampiran 5 Data pengujian sensor pH (sample pH 3) .....	49
Lampiran 6 Data pengujian sensor pH (sample pH 4) .....	50
Lampiran 7 Data pengujian sensor pH (sample pH 5) .....	51
Lampiran 8 Data pengujian sensor pH (sample pH 6) .....	52
Lampiran 9 Data pengujian sensor pH (sample pH 7) .....	52
Lampiran 10 Data pengujian sensor pH (sample pH 8) .....	53
Lampiran 11 Data pengujian sensor pH (sample pH 9) .....	54
Lampiran 12 Data pengujian sensor pH (sample pH 10) .....	55
Lampiran 13 Data pengujian sensor pH (sample pH 11) .....	56
Lampiran 14 Data pengujian sensor pH (sample pH 12) .....	56
Lampiran 15 Data pengujian sensor pH (sample pH 13) .....	57
Lampiran 16 Data pengujian sensor pH (sample pH 14) .....	58
Lampiran 17 Data pengujian sensor Turbidity (Percobaan 1) .....	59
Lampiran 18 Data pengujian sensor Turbidity (Percobaan 2) .....	59
Lampiran 19 Data pengujian sensor Turbidity (Percobaan 3) .....	60
Lampiran 20 Data pengujian sensor Turbidity (Percobaan 4) .....	61
Lampiran 21 Data pengujian sensor Turbidity (Percobaan 5) .....	61
Lampiran 22 Data pengujian sensor Turbidity (Percobaan 6) .....	62
Lampiran 23 Data pengujian sensor Turbidity (Percobaan 7) .....	63
Lampiran 24 Data pengujian sensor Turbidity (Percobaan 8) .....	63
Lampiran 25 Data pengujian sensor Turbidity (Percobaan 9) .....	64
Lampiran 26 Data pengujian sensor Turbidity (Percobaan 10) .....	65
Lampiran 27 Data pengujian sensor Turbidity (Percobaan 11) .....	65
Lampiran 28 Data pengujian sensor Turbidity (Percobaan 12) .....	66
Lampiran 29 Data pengujian sensor Turbidity (Percobaan 13) .....	67
Lampiran 30 Data pengujian sensor Turbidity (Percobaan 14) .....	67
Lampiran 31 Data pengujian sensor Turbidity (Percobaan 15) .....	68
Lampiran 32 Data pengujian sensor Turbidity (Percobaan 16) .....	68
Lampiran 33 Data pengujian sensor Turbidity (Percobaan 17) .....	69
Lampiran 34 Data pengujian sensor Turbidity (Percobaan 18) .....	70
Lampiran 35 Tampilan pengirim menggunakan komunikasi serial 232 .....	71
Lampiran 36 Tampilan penerima menggunakan komunikasi serial 232 .....	71

# Bab 1. Pendahuluan

## 1.1 Latar Belakang

Kualitas air layak guna diperhatikan dari beberapa parameter diantaranya tingkat keasaman (pH) dan tingkat kejernihan air. Maka diperlukanlah sistem pemantau kualitas air yang mampu menampilkan data tingkat keasaman dan kejernihan air. Seiring dengan kebutuhan data kualitas air tersebut maka dibutuhkan juga sistem monitoring yang mampu menyediakan jalur pengiriman data yang beragam seperti serial RS-232, RS-485, canbus dan TCP-IP.

Penelitian dan pengembangan mengenai sistem monitoring kualitas air telah dilakukan oleh beberapa peneliti, diantaranya (Yunior, Y. T. K., & Kusriani, K. (2021)) dengan penelitian yang berjudul “ Sistem Monitoring Kualitas Air pada Budidaya Perikanan Berbasis *IoT* dan Manajemen Data “. Pada penelitian ini dijelaskan bahwasanya penggunaan metode *Internet of Things* (IoT) dapat membantu mempermudah sistem monitoring dan control kualitas air sehingga berdampak efisiensi sistem budidaya. Pembahasan juga meliputi penggunaan parameter pH, turbidity, dilakukan sebagai identifikasi kadar kualitas air sehingga dapat dijadikan sebagai acuan dan Analisa pada pemberian rekomendasi manajemen *control* kualitas air kolam.[1]

Penelitian dengan judul “ Sistem Monitoring *pH* dan Kekeruhan Kolam Ikan Koi Berbasis *Internet of Things*” oleh (Taufik, A., & Fadhil, A.)Menggunkan Aplikasi Blynk “. Menjelaskan tentang fokus untuk merancang sebuah alat yang digunakan untuk memamntau pH dan kekeruhan air kolam ikan koi menggunakan media terpal, system ini memberi informasi yang sedang berlangsung melalui aplikasi Blynk. Hasil akhir pada system ini bahwa alat dapat memantau pH dan kekeruhan secara otomatis dan tepat. Persentase error yang diperoleh sensor pH adalah 0,6853%, dan persentase error pada sensor kekeruhan sebesar 3,704%.[2]

Penelitian dengan judul “ Pengukur Tingkat Kekeruhan Keasaman Dan Suhu Air Menggunakan Mikrokontroler Atmega328p Berbasis Android “. (Rufiyanto, A., Abdilah, G. S., & Purwaningrum, S. D.)membuat sistem alat pendeteksi air yang berfokus pada parameter kekeruhan air yang dideteksi oleh sensor, keasaman dan kebasaaan air diukur menggunakan sensor pH. Parameter yang diolah akan diproses melalui mikrokontroler, dengan berbasis program arduino ide, untuk menghubungkan mikro dengan smartphome android, maka modul bluetooth dipasang pada mikro. Aplikasi andorid yang digunakan untuk menampilkan data dari alat ukur ke smartphome dibuat di App inventor. Penulis menyimpulkan bahwa alat ukur yang dibuat, membuat pengukuran lebih efisien.[3]

Berdasarkan latar belakang melalui penelitian diatas penulis bermaksud merancang sebuah alat yang dapat memonitoring air dengan software interface, penelitian ini berjudul “ Alat Monitoring Air dengan Menggunakan Multi Jalur Komunikasi Data (Serial 232, RS-485, CANBus, dan TCP/IP). Dengan tujuan alat ini dapat memonitoring kualitas air dengan jarak, dengan parameter keasaman,

kebiasaan, dan kekeruhan.

### 1.2 Rumusan Masalah

Adapun rumusan masalah yang dapat diangkat dari latar belakang yang telah diketahui yaitu sebagai berikut:

1. Bagaimana akuisisi data sensor dilakukan ?
2. Bagaimana mengirimkan hasil dari data sensor menggunakan komunikasi?
3. Bagaimana menampilkan hasil data sensor pada *LCD*, dan Aplikasi ?
4. Bagaimana cara mendapatkan nilai sensor yang lebih akurat?

### 1.3 Tujuan

Adapun tujuan untuk menjawab rumusan masalah dari penelitian alat monitoring air jalur multi komunikasi:

1. Akuisisi data sensor dilakukan dengan menggunakan mikrokontroler Arduino Uno, dan perangkat lunak yaitu Arduino IDE.
2. Hasil data sensor dapat dikirim melalui beberapa komunikasi data yaitu Serial *RS-232*, *RS-485*, *CANBus*, dan *TCP/IP*. Tiap komunikasi data memiliki *COM*, dan *IP Adress* yang harus disesuaikan dengan perangkat penerima data.
3. Menampilkan hasil data sensor, dengan membuat tampilan *LCD*, dan aplikasi yang akan digunakan yaitu *C#*, dan web.
4. Membuat sistem monitoring yang dilengkapi dengan menu kalibrasi sensor *pH*, dan sensor turbidity.

### 1.4 Manfaat

Adapun dibawah ini merupakan manfaat yang diperoleh dari pembuatan alat monitoring air dengan multi jalur komunikasi sebagai berikut :

1. Dapat mengetahui kualitas air, dengan parameter *pH*, kekeruhan, dan tegangan masing - masing sensor melalui tampilan *LCD* yang tersedia.
2. Menyediakan jalur multi komunikasi data yaitu Serial *232*, *RS-485*, *CANBus*, dan *TCP/IP*.
3. Dapat mengetahui kualitas air melalui aplikasi yang sudah dibuat yaitu *C#*, dan *Websserver*.

### 1.5 Batasan

Adapun dibawah ini merupakan batasan masalah pada penelitian ini, dimana batasan merupakan kapasitas yang dimiliki oleh alat :

1. Sensor *pH* *E201* memiliki rata - rata error sebesar 4,39% %, dan resolusi pembacaan sensor mulai dari *pH* 0 – 14.
2. Sensor *turbidity* *SEN0189* memiliki rata - rata error sebesar 18,15 % dan resolusi pembacaan sensor mulai dari 0 -1000 *NTU*.
3. Kalibrasi pada sensor *pH*, dan sensor turbidity masih dilakukan dengan metode pendekatan linear

### 1.6 Struktur Pembagian Kerja

Struktur rincian kerja biasanya memuat pekerjaan yang akan dilakukan dalam mengerjakan suatu proyek. Dibawah ini merupakan struktur yang telah ditetapkan bersama oleh seluruh anggota tim, dan telah disetujui oleh manajer proyek.

**Tabel 1 Struktur Pembagian Kerja**

No	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Cesilia Margaritha Boru Harianja	Ketua – Alat monitoring air pada bagian pH air menggunakan sensor <i>pH</i> seri E-201, dan kekeruhan air menggunakan Sensor <i>turbidity</i> dengan komunikasi RS-485.
2	Alfaiz Ferdiansyah	Anggota - Alat monitoring air pada bagian pH air menggunakan sensor <i>pH</i> seri E-201, dan kekeruhan air menggunakan Sensor <i>turbidity</i> dengan komunikasi TCP/IP.
3	Ferdito Nois Fismansyah	Anggota - Alat monitoring air pada bagian pH air menggunakan sensor <i>pH</i> seri E-201, dan kekeruhan air menggunakan sensor <i>turbidity</i> dengan komunikasi Serial 232.
4	Andhika Maulana	Anggota - Alat monitoring air pada bagian pH air menggunakan sensor <i>pH</i> seri E-201, dan kekeruhan air menggunakan sensor <i>turbidity</i> dengan komunikasi CANBus.

## Bab 2. Tinjauan Pustaka

Pada jurnal yang berjudul “ Rancang Bangun Sistem Pengukuran Kualitas Air untuk Keperluan Higiene Sanitasi Berbasis Arduino UNO “ (Muhaamad Fauzan Zarkashie, 2021). Penulis menyampaikan bahwasanya air merupakan salah satu kebutuhan pokok yang berperan penting menyokong kehidupan di bumi. Air digunakan untuk pemeliharaan kebersihan, mencuci bahan pangan, peralatan makanan, dan pakaian. Selain itu air untuk hygiene sanitasi yang dapat digunakan sebagai air baku air minum. [4] Ada beberapa *indicator* dalam memonitoring air, diantara lain seperti *pH*, kadar oksigen, padatan terlarut, suhu, dan kekeruhan. Penelitian yang akan dilakukan dengan mengukur tingkat keasaman, kebasaaan, dan kekeruhan air. Tingkat keasaman, dan kebasaaan diukur menggunakan sensor *pH*, dan kekeruhan menggunakan sensor *turbidity*, hasil nilai sensor yang akan dikirimkan menggunakan komunikasi yaitu serial 232, RS-485, CANBus, dan TCP/IP.

*pH* mengukur interaksi antara *ion hydrogen* dan hidroksida dalam larutan air. Larutan dengan ion *hydrogen* dalam jumlah besar adalah asam, dan larutan dengan aktivitas ion hidroksida dalam jumlah besar adalah basa. Skala pH berkisar dari 0 hingga 14. *pH* memerankan peran penting dalam sifat sifat cairan, ketika *pH* tinggi akan membuat logam berat berkurang, hal tersebut menunjukkan bahwa suatu larutan itu beracun. Dan perlu diketahui bahwa tidak semua cairan memiliki *pH* (Anugerah Ayu Sendari, 2022).

Keadaan dimana transparansi suatu zat cair berkurang akibat kehadiran zat – zat tak terlarut, disebut dengan kekeruhan. Air yang bersih yaitu yang tidak berwarna, tidak berbau, dan tidak beracun. Akan tetapi adakalanya bahwa air dapat berwarna keruh dan berbau. Menurut Peraturan Menteri Kesehatan No. 416/MENKES/PER/IX/1990 air dapat memenuhi syarat kualitas pemakaian untuk air bersih dengan kekeruhan maksimum diperbolehkan yaitu 25 NTU, dan untuk air minum tidak lebih dari 5 NTU (Mochmad Firzha A). *Nephelometric Turbidity Unit (NTU)* adalah satuan standar untuk mengukur kekeruhan air.

Dalam system komunikasi data, data dikumpulkan dari berbagai melalui transmisi data dan kemudian menghasilkan keluaran, dalam system komunikasi data, ada informasi yang akan dikomunikasikan, bisa dalam bentuk informasi, data, dan sebagainya. Terdapat perangkat pengirim data – data seperti *computer*. Ada perangkat penerima data seperti *computer*, dan sekeumpulan aturan yang memberikan perintah komunikasi data yang menjelaskan kesepakatan antara perangkat komunikasi. Tanpa adanya protocol, maka kedua perangkat tidak dapat berkomunikasi.

Penelitian ini memiliki perangkat keras, dan perangkat lunak yang akan digunakan, dan akan dijelaskan sebagai berikut ini :

## 1. Sensor pH

Sensor yang sudah dilengkapi dengan *probe* dan modul *converter* dengan keluaran antarmuka data *analog*, sehingga bisa dihubungkan dengan board Arduino UNO. Elektroda kaca berfungsi untuk merubah jumlah ion yang ada dalam larutan dan elektroda referensi berfungsi untuk merubah jumlah ion yang terbaca oleh elektroda kaca menjadi nilai tegangan analog. Dengan prinsip kerja yaitu semakin banyak elektron yang terdeteksi pada sampel maka semakin bernilai pula cairan tersebut, dan apabila semakin sedikit elektron yang terdeteksi maka sampel cairan tersebut bersifat basa [4]. Sensor *pH* memiliki rentang pengukuran 0,00-14,00 *PH*. Modul *pH* sensor memiliki spesifikasi tegangan 0V – 5V, dengan *output* pin analog. Gambar 1 merupakan kelengkapan modul sensor *pH*, berikut adalah penjelasan komponen dari sensor *pH* tersebut:



**Gambar 1 Sensor pH**

1. Probe Sensor *pH*, berfungsi untuk mengukur kadar pH suatu air.
2. Module Sensor *pH* untuk mengkonversi *signal* dari *pH* sensor agar dapat dibaca oleh ADC pada mikrokontroler dan juga dimanfaatkan untuk mengkalibrasi bacaan sensor dan mengatur lebar bacaan.

**Table 2 Spesifikasi sensor pH**

Tegangan input	5V
Arus kerja	5-10 mA
Konsentrasi yang dapat terdeteksi	pH 0-14
Deteksi suhu	0-80 degC
Waktu respons	<5 detik
Waktu stabilitas	<60 detik
Output Data	Analog
Power	< 0,5 W
Ukuran	42 mm x 32 mm x 20 mm

## 2. Sensor Turbidity

Sensor kekeruhan mendeteksi air dengan mengukur tingkat kekeruhan, sensor

ini menggunakan cahaya untuk mendeteksi partikel tersuspensi dalam air dengan mengukur transmisi cahaya yang berubah dengan jumlah total padatan tersuspensi (TSS) dalam air.[5] Sensor ini menyediakan mode keluaran sinyal *analog* dan *digital*. Keluaran sinyal analog, nilai keluaran akan menurun saat berada dalam cairan dengan kekeruhan tinggi. Sensor ini memiliki spesifikasi tegangan keluaran 0-4 - 5V, dan memiliki range pembacaan sensor yaitu 0 – 1000 NTU. Gambar 2 dibawah merupakan beberapa komponen dari sensor turbidity dan berikut penjelasan masing-masing komponen tersebut:

1. *Module Probe Turbidity*, berfungsi untuk mendeteksi tingkat kekeruhan suatu air.
2. *Sensor Probe Turbidity*, berfungsi untuk mendeteksi cahaya.
3. *Cable sensor probe*, sebuah kabel yang digunakan untuk menghubungkan sensor probe turbidity dengan module *probe turbidity*.
4. *Cable to microcontroller*, sebuah kabel yang berfungsi untuk menghubungkan sensor kekeruhan dengan mikrokontroler.



**Gambar 2 Sensor Turbidity**

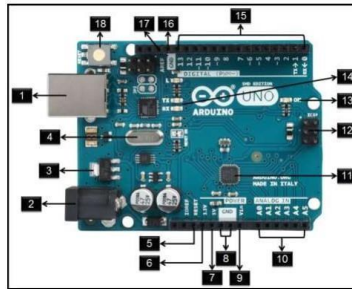
**Table 3 Spesifikasi Sensor Turbidity**

Tegangan Operasional	5V
Tegangan Arus	40mA
Waktu Respon	<500ms
Analog Output	0-4.5V
Temperatur	5°C-90°C
Berat	30g
Range pengukuran	0-3000

### 3. Arduino UNO

Arduino Uno adalah salah satu produk elektroniknya memiliki mikrokontroler ATmega 328. Mikrokontroler dapat digunakan, dengan menghubungkan *board*

arduino Uno ke *computer* dengan menggunakan kabel *USB* untuk menjalankannya. Spesifikasi Arduino Uno memiliki 6 pin analog, dan 14 pin digital. Arduino Uno berfungsi membuat program untuk mengendalikan berbagai komponen elektronika. Terdapat bagian dan fungsi pada Arduino seperti gambar 3 dibawah ini, berikut ini penjelasan bagian dan fungsi pada Arduino:



**Gambar 3 Arduino uno**

1. Power *USB Plug* digunakan untuk menghubungkan papan Arduino dengan komputer atau power bank lewat koneksi *USB* sebagai *supply* listrik. [6]
2. *Power Jack/External Power Supply* sebagai alternatif *supply* sumber listrik untuk Arduino dengan tipe jack (DC 5 - 12V).
3. *Voltage Regulator* digunakan untuk menstabilkan tegangan eksternal
4. *Crystal Oscillator* digunakan sebagai layaknya detak jantung pada Arduino sebagai *timer* atau penghitung.
5. *Reset* digunakan untuk mengulang program Arduino dari awal atau reset. Cara pertama dengan menekan tombol reset (18) secara langsung atau dengan cara kedua yakni menghubungkan pin *reset* dengan *GND* secara singkat.
6. Pin 3.3V sebagai sumber tegangan *output* 3.3 volt.
7. Pin *VCC/ Pin 5V* sebagai sumber tegangan *output* 5 volt.
8. Pin *GND (Ground)* sebagai pin arus listrik negatif dalam sirkuit elektronik. *Ground* merupakan akhir dari setiap jalur arus listrik.
9. Pin *Vin* untuk memasok listrik dari luar ke papan arduino, seperti adapter (sekitar 5V). Jangan memasukkan tegangan dari luar bila *board* sudah mendapatkan *supply* dari *USB*.
10. Pin Analog *In*, papan Arduino UNO memiliki enam pin analog A0 - A5, yang digunakan untuk membaca sinyal atau sensor analog seperti sensor jarak, suhu, dan mengubahnya menjadi nilai digital. Nilai analog berkisar 0 – 1023.
11. *IC* atau *Integrated Circuit* merupakan “otak” dari papan Arduino yang akan diprogram untuk mengatur pin digital dan pin analog.
12. *ICSP (In Circuit Serial Programming)* digunakan untuk memprogram sebuah mikrokontroler menggunakan jalur *USB* serta sebagai jalur untuk transfer data.
13. *LED Power Indicator* sebagai indikator ada atau tidak adanya *supply* listrik

pada papan arduino.

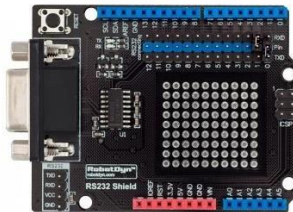
14. LED TX ( Transmit ) dan RX ( Receive ) dua LED tersebut akan berkedip saat pemrograman IC atau papan Arduino berlangsung.
15. Digital Pins I/O memiliki 14 digital pin sebagai I/O yang bekerja dengan cara memberikan nilai logika (0 atau 1). Digital Pin I/O bekerja seperti saklar. Terdapat pin khusus terdiri, yakni :
  - a. Pin 0 yang digunakan sebagai "serial-in" (RX),
  - b. Pin 1 digunakan sebagai "serial-out" (TX).

**Table 4 Spesifikasi Arduino uno**

Tegangan Operasional	5V
Input Volt	7–12V
Digital I/O Pin	16 Pin (4 Sebagai Output PWM)
Analog Input	6 Pin
Clock Speed	16 MHz
Berat	25g
Panjang dan Lebar	68,6 mm dan 53,4 mm
Limit Tegangan Input	20V

#### 4. Komunikasi Data Serial-232

Komunikasi Serial-232 adalah standar komunikasi serial yang didefinisikan sebagai antarmuka antara perangkat terminal data, dan perangkat komunikasi data.[7] Berdasarkan spesifikasi serial-232 memiliki kecepatan 256 kbps atau lebih rendah dengan jarak kurang dari 15 meter, komunikasi RS-485 dapat digunakan untuk komunikasi data antara dua komputer secara langsung.



**Gambar 4 RS-232 shield**

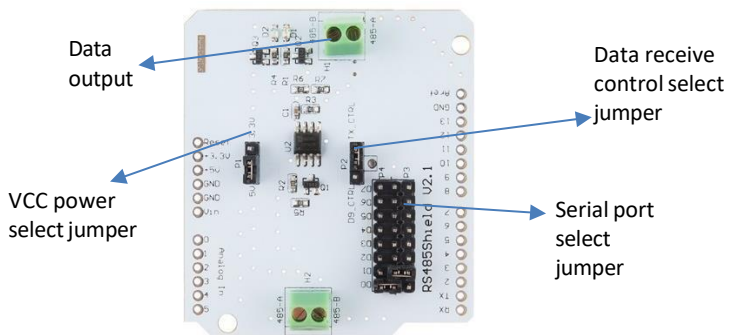
**Table 5 Spesifikasi RS-232 shield**

<i>Voltage</i>	5V
Digital I/O Pin	30 Mbps
Analog Input	1.200m
Modul size	55mmx53mm (2.16"x2.08")
<i>Maximum range / cable length</i>	1000ft

**5. Komunikasi Data RS-485**

RS-485 adalah teknik komunikasi data serial , komunikasi data yang dapat dilakukan pada jarak yang cukup jauh yaitu 1,2 Km[8] RS-485 shield mampu menghubungkan beberapa penerima dalam satu pengirim. Kecepatan transmisi dari RS-485 dapat diukur dalam *baudrate*. Banyak faktor yang mempengaruhi kecepatan maksimum yang dapat dicapai, termasuk kualitas kabel, dan kemampuan perangkat keras yang terhubung dengan komunikasi. Kecepatan *baudrate* yang umumnya digunakan adalah 9600, 19200, 38400,57600, dan 115200. Gambar 5 dibawah merupakan beberapa komponen dari serial komunikasi RS-485 dan berikut penjelasan masing-masing komponen tersebut:

1. VCC power sebagai pin *input* catu daya berupa 5 Volt dan 3.3 Volt.
2. Data receive sebagai jalur pengirim data.
3. Data output sebagai jalur komunikasi data yang akan dikirim.
4. Serial port sebagai jalur *transmitter* dan *receiver*.



**Gambar 5 RS-485 shield**

**Table 6 Spesifikasi RS-485 shield**

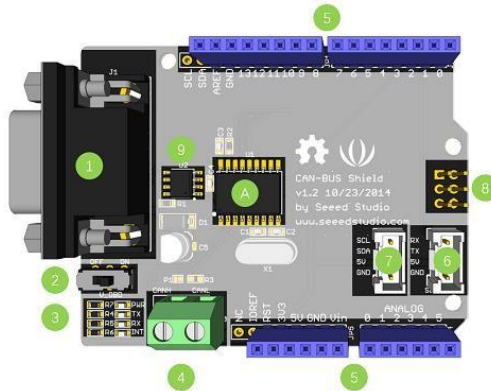
Tegangan Operasional	5V
Kecepatan Transfer Data	30 Mbps
Jarak Dukung Transfer Data	1.200m

<i>IC Chip</i>	MAX485
Ukuran	44 x 14 mm

## 6. Komunikasi Data *CANBus*

*CANBus* dapat digunakan untuk komunikasi data multi point sama seperti EIA-485 [9]. *CANBus* bisa terdapat lebih dari dua perangkat yang dapat bertukar data. Tidak ada master dan slave pada komunikasi *CANBus*, semua perangkat yang terhubung dalam satu jaringan disebut dengan node, dan masing-masing node dapat berkomunikasi sesamenode lainnya selama masih dalam satu jaringan yang sama. *CANBus shield* ini mengadopsi *MCP2551 CANBus controller* dengan antarmuka *SPI* dan *transceiver MCP2551 CAN* untuk memberikan kemampuan *CAN-BUS Arduino*. Gambar 5 dibawah ini merupakan beberapa komponen dari *CAN Bus* berikut penjelasan dan fungsinya:

1. Antarmuka DB9, untuk terhubung ke Antarmuka OBDII melalui kabel OBD.
2. *V\_OBD*, mendapat daya dari Antarmuka OBDII (dari DB9).
3. Lampu Indikator:
  - a. PWR: Kekuatan.
  - b. TX: Berkedip saat data sedang dikirim.
  - c. RX: Berkedip saat ada penerimaan data.
  - d. INT: Interupsi Data
4. Terminal *CAN\_H* dan *CAN\_L*
5. Arduino Uno di pin keluar.
6. Konektor Serial Grove
7. Konektor I2C Grove
8. Pin ICSP.
9. IC *MCP2551*, transceiver CAN berkecepatan tinggi (lembar data).
10. IC *MCP2515*, pengontrol CAN yang berdiri sendiri dengan Antarmuka *SPI*.



**Gambar 6 CANBus shield**

**Tabel 7 Spesifikasi CANBus shield**

Kecepatan komunikasi	1Mb/Detik
Tegangan Kerja	5V (DC)
Bekerja Saat Ini	5mA
Antarmuka	SPI
Ukuran	68 mm x 53 mm
IC Chip	MCP2515

## 7. Komunikasi Data TCP/IP

TCP/IP adalah rangkaian protocol komunikasi yang digunakan untuk menghubungkan perangkat jaringan di internet [10]. TCP mendefinisikan bagaimana aplikasi dapat membuat saluran komunikasi di seluruh jaringan. IP memberi pengertian cara menangani setiap paket untuk memastikan paket mencapai tujuan yang benar.



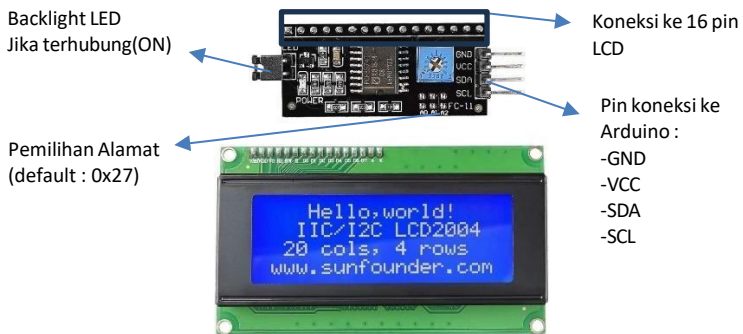
**Gambar 7 Ethernet shield**

**Tabel 8 Spesifikasi ethernet shield**

Connector	RJ45
Controller	W5500
Speed	10/100 Mbps
Communication	SPI
Internal memory	32kb
Maximum sockets	8 individual
Supported protocols	IPv4, ICMP, TCP, UDP, ARP, IGMP, PPPoE, MQTT

## 8. LCD

LCD I2C adalah module LCD yang dikendalikan secara serial sinkron dengan protocol I2C. Arduino sendiri sudah mendukung protocol I2C [11], pada papan Arduino UNO, port I2C terletak pada pin A4 untuk jalur SDA (serial data), dan pin A5 untuk jalur SCL (serial clock). Untuk sisi software, Arduino membantu bekerja dengan protocol ini melalui library 'Wire.h'. Library *LiquidCrystal\_I2C.h* akan dimanfaatkan untuk mengkonversi jalur *parallel LCD* menjadi jalur serial I2C.



**Gambar 8 LCD I2C 20x4**

Dibawah ini merupakan komponen dari I2C berikut penjelasan dan fungsinya:

- A. Pin koneksi ke Arduino:
  - GND: terhubung dengan GND Arduino.
  - VCC: terhubung dengan 5V.
  - SDA: terhubung dengan pin SDA (A4).
  - SCL: terhubung dengan pin SCL (A5).
- B. Jumper backlight berfungsi untuk memilih apakah LED backlight (LED lampu latar LCD) nyala atau padam (opsional). Jika ingin LED nyala maka jumper pada posisi ON (terpasang).
- C. Pemilihan Alamat, A0, A1, A2 untuk pemilihan *address* (alamat) dari I2C. Pada

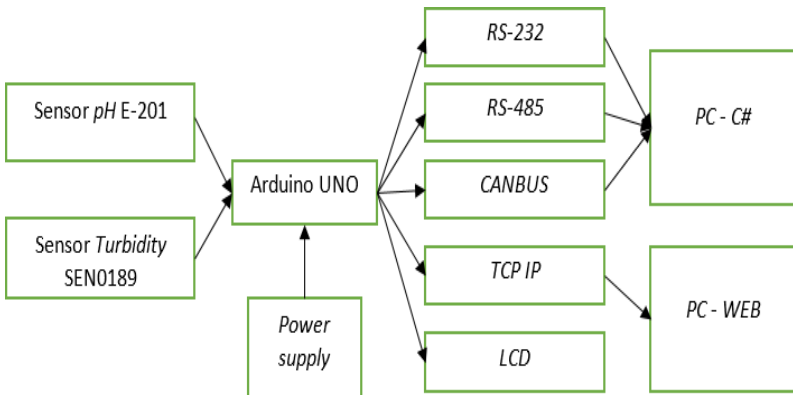
kondisi *default* (tidak terhubung antara A0, A1, A2) maka alamatnya 0x27.  
D. Koneksi ke pin *LCD* (16 pin).

**Tabel 9 Spesifikasi LCD I2C 20x4**

<i>Backlight</i>	Warna Biru
<i>Display</i>	16 karakter x 2 baris
Tegangan	5V DC
Ukuran	80mm x 35mm x 11mm

### Bab 3. Metode Penelitian

Pada pembahasan ini, akan membahas mengenai blok diagram dari sistem alat, perancangan elektrikal, perancangan mekanikal, serta pengujian yang akan dilakukan dalam penelitian ini. Selain itu juga memaparkan kebutuhan akan alat dan bahan yang akan digunakan. Dibawah ini akan dimulai dari blok diagram berkaitan dengan sistem yang dibuat, blok diagram disusun guna mempermudah dalam memahami hubungan antara komponen, memberikan gambaran yang jelas tentang alur data dari sensor hingga pada tampilan. Blok diagram juga membantu dalam perancangan sistem, dan membantu seluruh anggota tim dalam berkomunikasi dan menentukan peran dari tiap – tiap komponen.



**Gambar 9 Blok Diagram Sistem**

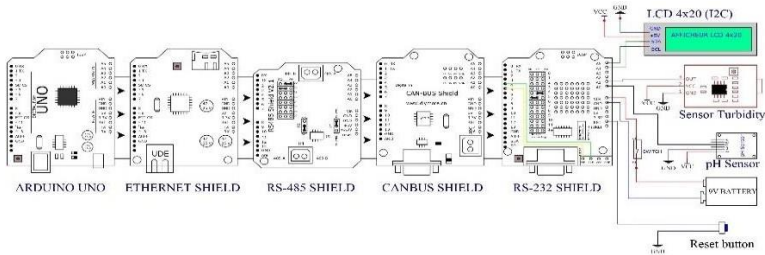
Dalam sistem ini, digunakan dua sensor yaitu pH dan turbidity, yang terhubung kepada mikrokontroler Arduino UNO. Mikro berperan sebagai mengumpulkan data dari kedua sensor dan mengirimkannya melalui beberapa komunikasi yang digunakan. Dalam komunikasi serial 232, data dari sensor dikirimkan melalui koneksi serial 232, demikian pada komunikasi *RS-485*, dan *CANBus*, data sensor dikirimkan melalui *protocol* masing – masing komunikasi.

Selanjutnya data ditampilkan pada *interface*, program yang digunakan adalah program *C#*, program *C#* dapat membaca data dari Arduino UNO melalui koneksi serial 232, *RS-485*, dan *CANBus*. Data juga dapat diakses melalui jaringan *protocol TCP/IP*, tampilan *web* akan memberikan akses ke data sensor dari perangkat yang terhubung kepada jaringan.

Sistem ini juga dilengkapi dengan tombol *reset* guna memungkinkan dalam melakukan *reset* pada *system*, tombol *reset* dapat digunakan untuk memulai ulang mikrokontroler Arduino UNO. Sistem ini diberi sumber oleh baterai 9 volt.

### 3.1 Perancangan Elektrikal

Dalam pembahasan ini, akan dijelaskan elektrikal dari suatu sistem yang akan dibuat. Sistem ini menggunakan jalur komunikasi serial 232, RS-485, CANBus, dan TCP/IP, dan untuk mengukur parameter penting yaitu pH dan turbidity. Sumber daya yang digunakan adalah baterai, berikut dibawah ini merupakan gambar perancangan elektrikal, beserta penjelasannya.



**Gambar 10 Perancangan Elektrikal**

Perancangan elektrikal diatas, dimulai dari mikrokontroler yang dihubungkan dengan keempat komunikasi. Komunikasi yang berada dipaling atas, merupakan komunikasi yang terhubung kepada LCD, sensor, dan tombol reset, komunikasi tersebut merupakan komunikasi serial 232. Elektrikal diatas, menunjukkan terhubungnya komunikasi dengan dua buah sensor, dan satu buah tombol reset, tombol reset bertujuan untuk mengulang kembali suatu proses. Masing – masing dari komunikasi memiliki kabel keluarannya.

**Tabel 10 Wiring Schedule komunikasi**

<b>Board Arduino UNO</b>	<b>Board Ethernet Shield</b>	<b>Board RS-485 Shield</b>	<b>Board CANBus Shield</b>	<b>Board RS-232 Shield</b>
5V	-	VCC	VCC	VCC
GND	GND	GND	GND	GND
3,3	VCC	-	-	-
Digital pin 2	-	TXD	-	-
Digital pin 3	-	RXD	-	-
Digital pin 4	-	-	-	RXD
Digital pin 5	-	-	-	TXD
Digital pin 10	CS	-	CS-	-
Digital pin 11	SI (MOSI)	-	SI (MOSI)	-

Digital pin 12	SO (MISO)	-	SO (MISO)	-
Digital pin 13	SCK	-	SCK	-

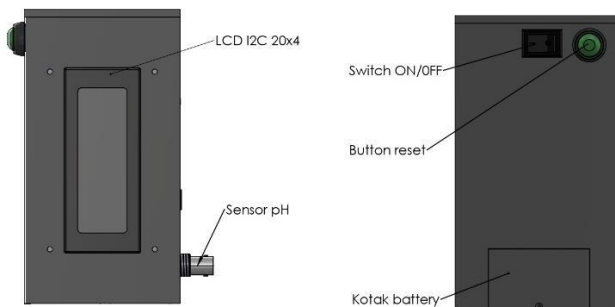
**Tabel 11 Wiring Schedule sensor dan komponen pelengkap**

Arduino UNO	Sensor Turbidity	Sensor pH	Battery 9V	Reset Button	LCD I2C 20x4
5V	VCC	V+	(+)	-	5V
GND	GND	G	(-)	(-)	GND
VIN	-	-	-	-	-
RESET	-	-	-	(+)	-
Analog pin 0	-	PO	-	-	-
Analog pin 1	OUT	-	-	-	-
Analog pin 4	-	-	-	-	SDA
Analog pin 5	-	-	-	-	SCL

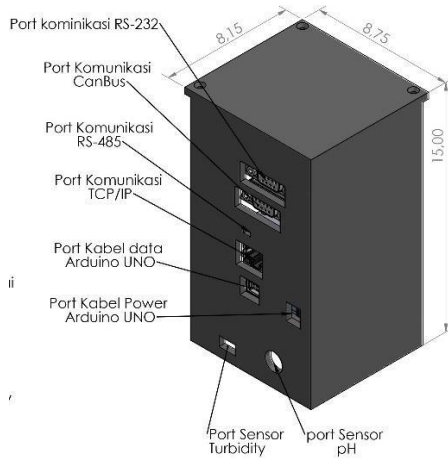
Tabel diatas merupakan tabel *wiring schedule*, dibuat guna untuk menyediakan panduan yang jelas akan pengaturan kabel dan konektor. Hal ini membantu memastikan bahwa setiap kabel terhubung dengan benar, dan dapat meminimalisirkan kesalahan dalam pemasangan rangkaian, serta memungkinkan untuk mempermudah dalam perbaikan.

### 3.2 Perancangan Mekanikal

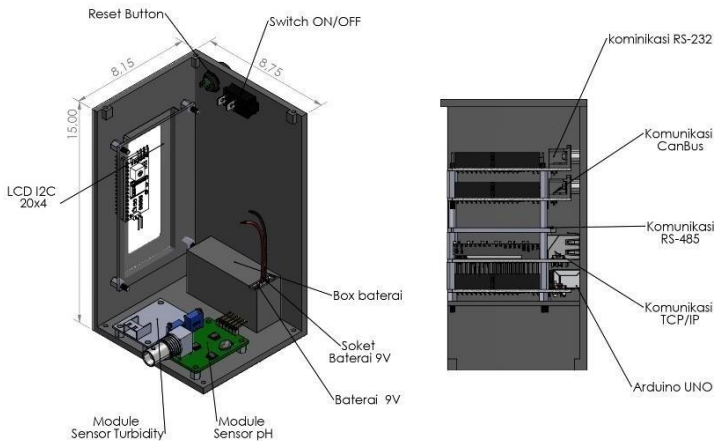
Pembahasan kali ini akan masuk pada mekanikal alat yang merupakan inti dari desain. Pembahasan mencakup perincian rancangan, komponen yang terlibat, serta baut – baut yang diperlukan.



**Gambar 11 Perancangan Mekanikal tampak luar**



**Gambar 12 Perancangan Mekanikal tampak luar**



**Gambar 13 Perancangan Mekanikal tampak dalam**

Pada desain mekanikal diatas, ukuran *cover* alat adalah 15 x 8,15 cm, dan pada *cover* alat ini kami membaginya menjadi 2 bagian agar memudahkan saat

melakukan perakitan alat, pada bagian cover pertama terdapat 4 buah komunikasi, dan pada bagian cover yang kedua terdapat sensor *pH* dan *turbidity*, *battery 9v*, *LCD 20x4*, dan pada tiap komponen dipasangkan spacer 0,5 cm pada bagian bawah, tujuannya adalah agar tiap komponen saat dilakukan perakitan bisa sejajar, pada penutupan alat kami menyatukan 2 bagian dengan baut M3x7.

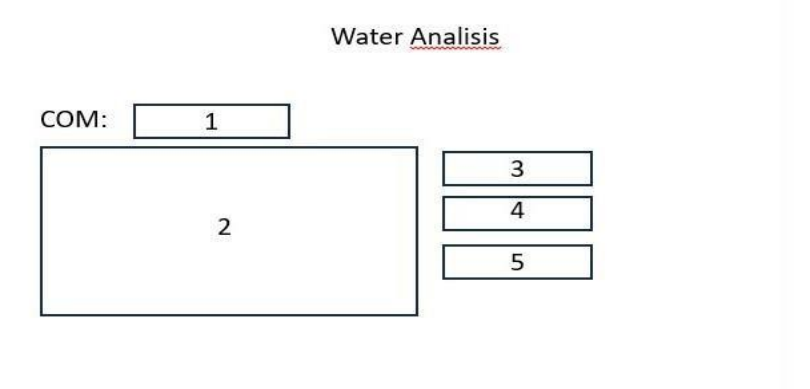
### 3.3 Perancangan Software

Desain tampilan harus dapat memastikan bahwa informasi yang ditampilkan dapat dibaca dan dipahami oleh pengguna. Pada alat monitoring air dengan multi komunikasi, tampilan dirancang agar dapat menampilkan data dari berbagai sumber komunikasi.

#### 1. Perancangan Tampilan Aplikasi C#

Desain tampilan yang akan dibuat pada aplikasi C#, harus mampu membuat pengguna paham akan *item* yang tersedia. Tampilan memuat isi yang berguna dalam kelangsungan penampilan data.

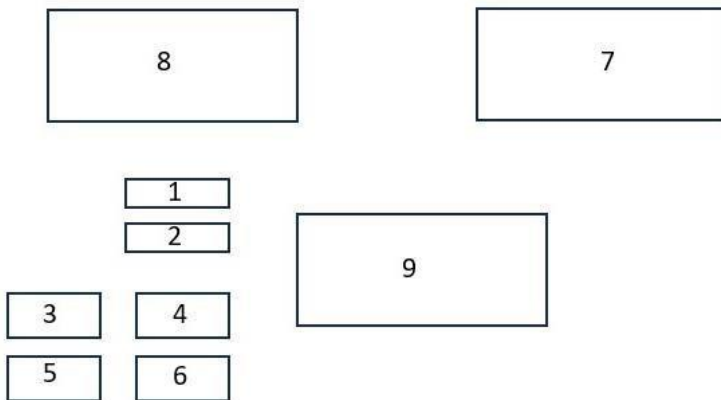
##### a. Perancangan tampilan aplikasi C#, menggunakan komunikasi Serial 232, RS-485, dan CANBus



**Gambar 14 Perancangan Software**

Gambar diatas merupakan perancangan applikasi C# pada alat monitoring air dengan jalur komunikasi Serial 232, dan RS-485, berikut adalah *item* yang digunakan pada tampilan:

1. Kolom *text box* untuk mengetikkan com yang terdeteksi pada PC.
2. Tampilan untuk *Voltage*, *ADC*, juga nilai kadar *pH* dan *turbidity*.
3. *Button* untuk menampilkan hasil pengukuran.
4. *Button* untuk menghentikan hasil pengukuran.
5. *Button* untuk menghapus hasil pengukuran.



**Gambar 15 Perancangan Software untuk komunikasi CANBus**

Gambar diatas merupakan perancangan applikasi C# pada alat monitoring air dengan jalur komunikasi CANBus, berikut adalah *item* yang digunakan pada tampilan:

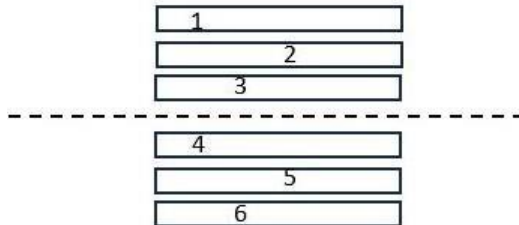
1. Kolom *text box* untuk mengetikkan *COM* yang terdeteksi pada *PC*.
2. Kolom *text box* untuk mengetikkan *Baudrate* agar sesuai pada *Arduino IDE*.
3. *Button* untuk menampilkan hasil pengukuran.
4. *Button* untuk menghentikan hasil pengukuran.
5. *Button* untuk menampilkan hasil pengukuran dari komunikasi *CANBus*.
6. *Button* untuk menyimpan
7. Tampilan untuk komunikasi *CANBus* yang masih berupa bilangan 0 dan 1.
8. Tampilan untuk menampilkan bilangan hexa dan decimal, sebelum dirubah menjadi data *pH* dan *turbidity*.
9. Tampilan untuk *voltage*, juga nilai kadar *pH* dan *turbidity*.

**b. Perancangan Tampilan Webserver**

Gambar dibawah merupakan perancangan applikasi *Web* pada alat monitoring air dengan jalur komunikasi *TCP/IP*, berikut adalah *item* yang digunakan pada tampilan:

# Arduino web server

Nilai pH dan kekeruhan:



**Gambar 16 Perancangan Software untuk Komunikasi TCP/IP**

Dilampirkan item yang akan dihadirkan pada tampilan web, sebagai berikut :

1. Kolom untuk menampilkan nilai *temperature*
2. Kolom untuk menampilkan nilai *pH*
3. Kolom untuk menampilkan nilai *voltage*
4. Kolom untuk menampilkan nilai kekeruhan
5. Kolom untuk menampilkan nilai *voltage*
6. Kolom untuk menampilkan nilai *ADC*

## 2. Flowchart Program

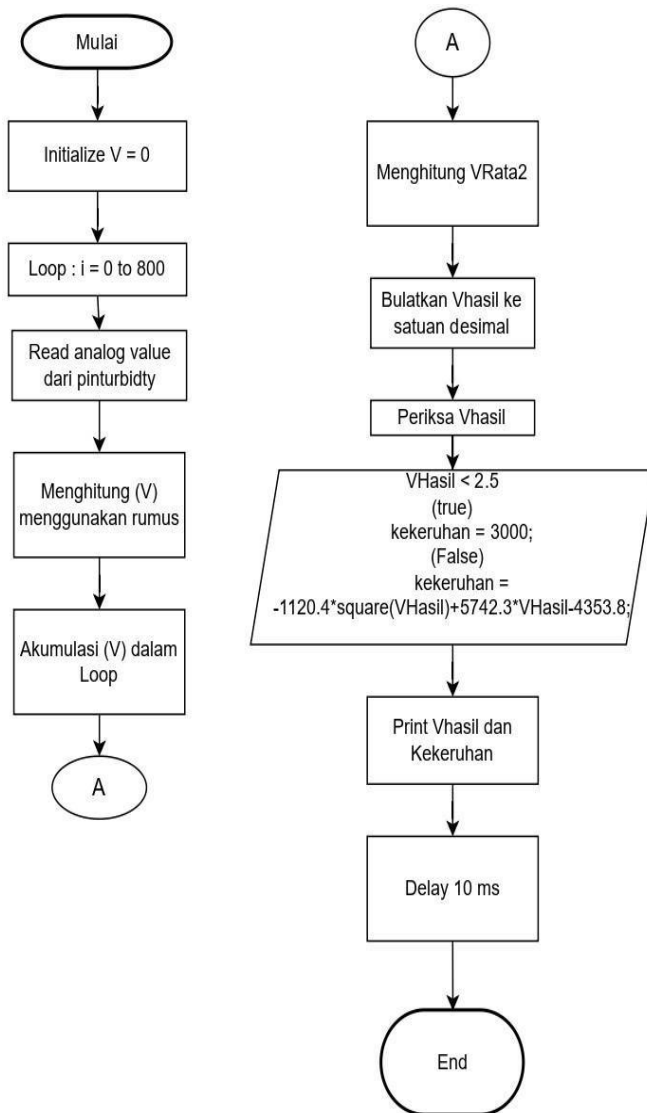
*Flowchart* program untuk pengolahan data sensor digunakan untuk menjelaskan langkah-langkah dalam mengolah data dari sensor. *Flowchart* ini juga bisa menjadi panduan dalam memahami logika program dengan jelas.

### a. Flowchart Program Pengolahan Data Sensor *Turbidity*

*Flowchart* program pada pengolahan sensor *turbidity* dimulai dengan menetapkan nilai *V* menjadi 0, selanjutnya lakukan perulangan dari 0 hingga 799. Selanjutnya, baca nilai analog dari pinTurb, dan hitung nilai *V* menggunakan rumus yang diberikan. Kemudian akumulasi nilai *V* dalam loop, hitung nilai rata – rata (*Vrata-rata*) dari *V*, dan bulatkan nilai *Vhasil* ke satu desimal. Lalu cek kondisi

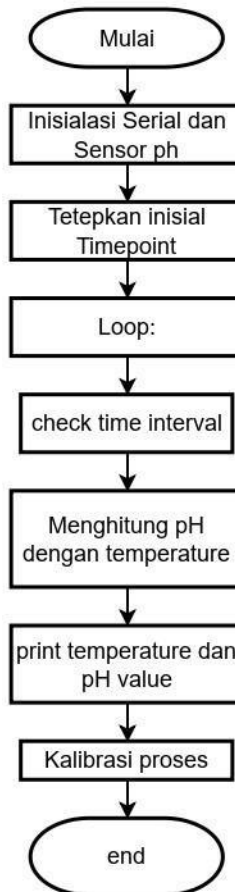
*Vhasil*, jika *Vhasil* kurang dari 2,5 buat nilai kekeruhan menjadi 3000, jika tidak hitung nilai kekeruhan menggunakan rumus yang diberikan. Tampilkan *Vhasil* dan kekeruhan, delay yang dimiliki adalah 10millidetik, dan akhiri eksekusi program.

*Flowchart* program ini menjelaskan langkah – langkah dalam program, dimulai dari inisialisasi, perulangan untuk membaca nilai sensor, hingga pada proses akhir.



**Gambar 17 Flowchart program turbidity**

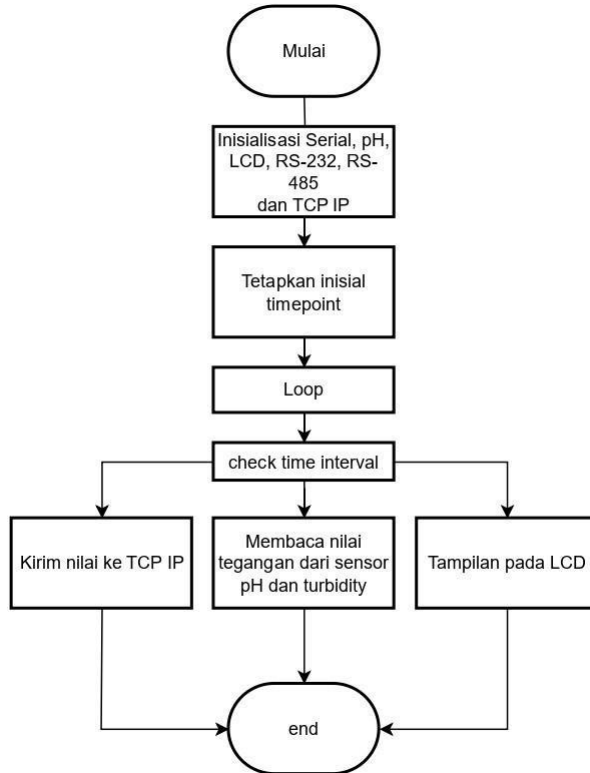
b. **Flowchart Program Pengolahan Data Sensor pH**



**Gambar 18 Flowchart program pH**

Gambar *flowchart* diatas menjelaskan langkah – langkah program dalam mengukur nilai *pH*. Dimulai dari program dimulai, masuk kepada inisialisasi untuk program komunikasi serial, inisialisasi sensor *pH*, tetapkan titik waktu awal untuk mengukur interval waktu. Selanjutnya membaca nilai tegangan dari sensor *pH*, menghitung nilai *pH* dengan kompensasi suhu menggunakan nilai tegangan yang dibaca, menampilkan nilai suhu dan *pH* yang dihitung, terakhir melakukan proses kalibrasi sensor, dan selesai.

c. **Flowchart Program Sistem**



**Gambar 19 Flowchart program sistem**

Gambar *flowchart* diatas, merupakan gambaran bagaimana proses program sistem berjalan, dimulai dari inisialiasi komponen komunikasi Serial 232, RS-485, CANBus, dan TCP/IP. Dilanjutkan dengan membaca nilai tegangan dari sensor pH dan turbidity, menampilkan nilai pH dan kekeruhan pada LCD, untuk komunikasi TCP/IP, data dikirimkan melalui protokol sesuai dengan protokol TCP/IP, kembali ke *loop* utama, dan selesai.

### 3.4 Alat dan Bahan

Dalam pembuatan Proyek akhir berikut memerlukan alat dan bahan seperti yang tertera pada tabel berikut ini :

**Tabel 12 Alat dan Bahan**

No	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Ket
1	Sensor <i>pH</i>	285.000	1	285.000	Dana Kampus
2	Sensor <i>Turbidity</i>	215.000	1	215.000	Dana Kampus
2	Kabel <i>USB</i>	10.000	4	40.000	Dana Kampus
3	Arduino UNO	150.000	4	600.000	Dana Kampus
4	<i>Shield Serial 232</i>	396.000	1	396.000	Dana Kampus
5	<i>Shield RS-485</i>	240.000	1	240.000	Dana Kampus
6	<i>Shield CANBus</i>	380.000	1	380.000	Dana Kampus
7	<i>Shield TCP/IP</i>	389.000	1	389.000	Dana Kampus
8	<i>LCD 20 X 4</i>	120.000	1	120.000	Dana Kampus
9	3D Filamen	300.000	1	300.000	Dana Kampus
10	Baut Aluminium	3.000	30	90.000	Dana Kampus
11	Kabel <i>Jumper</i>	2.000	40	80.000	Dana Kampus
12	Solder	76.000	1	76.000	Dana Kampus
13	Kabel <i>LAN</i>	50.000	1	50.000	Dana Kampus
14	<i>USB RS485</i>	15.000	2	30.000	Dana Kampus
15	<i>Project Board</i>	30.000	1	30.000	Dana Kampus
<b>Total</b>				<b>Rp.3.321.000</b>	

### 3.5 Pengujian

Kalibrasi sangat diperlukan untuk setiap jenis sensor supaya hasil pembacaan sensor memiliki tingkat akurasi yang tinggi. Kalibrasi diperlukan untuk memastikan bahwa hasil pengukuran yang dilakukan akurat dan konsisten. Kalibrasi sensor *pH* dapat dilakukan menggunakan *buffer* kalibrasi *pH* dengan nilai 4 dan 7.

Proses kalibrasi sensor *pH* dilakukan dengan langkah-langkah berikut:

- A. Langkah pertama dimulai ketika sensor telah dihubungkan dengan mikrokontroler, dengan *probe* sensor diisi oleh salah satu *buffer* standar, bisa menggunakan *buffer pH* 4, bisa juga menggunakan *buffer pH* 7.
- B. Setelah itu buka *command box* untuk mengetik perintah “enterpH”.
- C. Jika berhasil maka akan muncul tulisan “Enteh pH calibration Mode”.
- D. Langkah selanjutnya, adalah menulis “calpH” pada *command* Arduino IDE, dan tekan enter.
- E. Jika kita menggunakan *buffer* 4 maka akan muncul tulisan “Buffer solution 4.0, Send ExitpH to save and exit”. Jika kita menggunakan *buffer* 7 maka akan muncul tulisan “Buffer solution 7.0, Send ExitpH to save and exit”.
- F. Langkah terakhir, menulis perintah “exitph” pada *command box* Arduino IDE, dan tekan enter.
- G. Jika berhasil maka akan muncul tulisan “Calibration Succsesful, Exit pH Calibration Mode”.

Proses kalibrasi sensor *Turbidity* dilakukan dengan langkah-langkah berikut:

- A. Siapkan alat dan bahan dari sensor *turbidity* dan *turbidity* meter.
- B. Siapkan larutan Standar yang telah diukur menggunakan *turbidity* meter.
- C. Lakukan percobaan menggunakan sensor *turbidity* dengan larutan yang telah diukur menggunakan alat pembanding, catat hasil dari sensor dan alat pembanding, lakukan hal yang sama hingga air nya jernih.
- D. Kemudian lakukan persamaan garis linear.
- E. Setelah selesai masukkan persamaan linear ke dalam program Arduino.
- F. Upload programnya kemudian lakukan kembali pada pengambilan data dan lihat errornya.

#### 1. Pengujian Akuisisi Data Sensor *pH* dan *Turbidity* Menggunakan Arduino Uno

Pengujian ini dilakukan menggunakan Arduino Uno, ketika menggunakan sensor *pH*, maka menggunakan sampel air *pH* dari 1-14. Ketika menggunakan sensor *turbidity*, maka menggunakan sampel yang disediakan dengan kekeruhan air mulai dari 0 – 1000 *NTU*. Pengujian ini dilakukan pada *software* Arduino IDE, dan dalam setiap pengujian data yang diambil adalah dua puluh data. Data tersebut akan disalin, dan disalin kembali dalam Excel, dari excel, akan didapatkan nilai rata-rata dari setiap pengujian. Pada pengujian ini akan digunakan alat pembanding, yaitu *pH* meter, dan *turbidity* meter sebagai alat pembanding sensor.

Dalam pengambilan data menggunakan sensor *pH* diusahakan untuk selalu mencuci *probe* sensor, dengan menggunakan aquades, jika perpindahan *pH* dari asam ke basa, gunakan larutan NaOH, larutan HCL, dan aquades untuk membersihkan *probe* sensor.

## **2. Pengujian Menggunakan Komunikasi Dengan Hasil Data Sensor Tampil Pada Aplikasi**

Pengujian ini merupakan pengujian akhir, yaitu untuk menampilkan data sensor dari kedua sensor yaitu sensor *pH*, dan sensor *turbidity*, pada *LCD*. Data sensor juga tampil pada software Aplikasi Visual studio C#, dan *Webserver*. Tampilan data sensor pada aplikasi yang dibuat yaitu C#, dapat menggunakan tiga komunikasi yaitu Serial 232, *RS-485*, dan *CANBus*. Pada *TCP/IP*, data sensor tampil pada *webserver* yang tersedia pada *PC*.

## Bab 4. Hasil dan Pembahasan

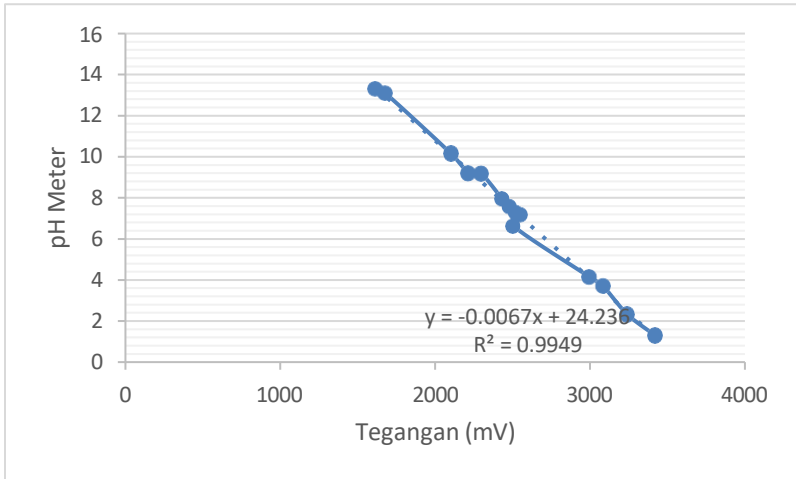
Dalam pengujian akuisisi data, terdapat dua sensor yang akan diakuisisi datanya, yaitu sensor *pH*, dan sensor *turbidity*, akuisisi data sensor menggunakan Arduino UNO sebagai perangkat keras, dan Arduino IDE sebagai perangkat lunak,

### 1. Hasil Pengujian Sensor *pH* Menggunakan Arduino Uno

Akuisisi data sensor *pH*, pada pengujian akuisisi data pada sensor *pH*, penelitian ini menggunakan alat pembanding sensor yaitu *pH* meter, *pH* meter disini berfungsi sebagai alat untuk membandingkan nilai, dengan sensor pada sistem yang dibuat. Selanjutnya data akan tampil pada serial monitor arduino IDE, menggunakan *source code* yang telah disediakan, ketika data tampil akan menunggu banyaknya data, sesuai dengan kebutuhan, *delay* yang digunakan adalah satu detik, jadi untuk dua puluh data yang dibutuhkan, perlu menunggu kurang lebih dua puluh detik untuk dapat menyalin data pada serial monitor Arduino IDE. Setelah data disalin dari serial monitor, pengolahan data dilakukan pada software Excel. Pada tabel dibawah ini, ditunjukkan hasil pengukuran dari *pH* meter, dan persentase error yang telah didapatkan melalui rumus persentase error.

Tabel 13 Pengujian Sensor *pH*

pH Meter	Sensor pH	Tegangan (mV)	Error %
1,3	1,04	3420	20,00
2,33	2,31	3240	0,86
3,73	3,43	3081	8,04
4,16	4,04	2994	2,88
6,65	7,52	2501	13,08
7,19	7,19	2549	0,00
7,3	7,41	2518	1,51
7,59	7,69	2478	1,32
7,95	8,01	2432	0,75
9,17	8,96	2298	2,29
9,19	9,53	2216	3,70
10,17	10,35	2100	1,77
13,11	13,35	1676	1,83
13,31	13,77	1615	3,46



**Gambar 20 Grafik Akuisisi Data Sensor *pH***

Gambar diatas merupakan grafik data dari *pH* meter terhadap data nilai *pH* dari serial monitor yang telah dirata-ratakan. Dengan sumbu X sebagai *pH* meter (*true value*), dan sumbu Y yaitu rata – rata nilai *pH* pada serial monitor, sebagai *indicator* yang ingin dibandingkan dengan *true value*.

## 2. Hasil Pengujian Sensor *Turbidity*

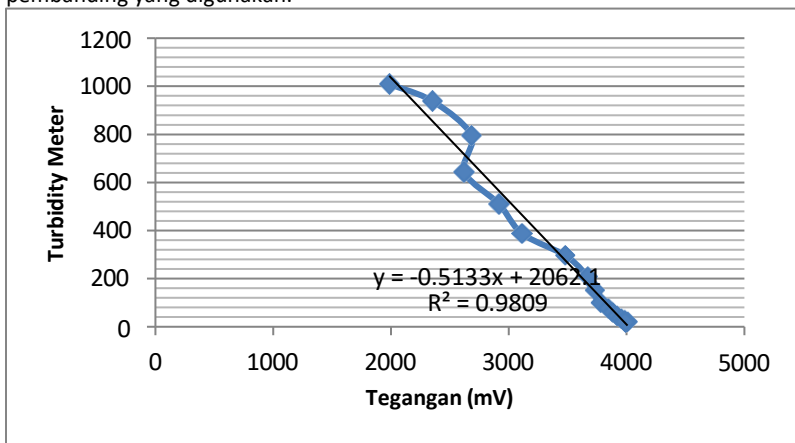
Pengambilan data oleh sensor *turbidity* menggunakan sampel air keruh yaitu teh tarik dilakukan sebanyak 18 kali. Tiap ganti satu kali percobaan, diambil 5ml air keruh, dan sebagai gantinya dimasukkan air bersih ke wadah sebanyak 5ml. Selain menggunakan sensor *turbidity*, digunakan juga *turbidity* meter, sebagai pembanding, untuk membandingkan dengan sensor *turbidity* yang dibuat. Pengambilan data ini, bertujuan untuk mendapatkan persamaan garis linear. Berikut merupakan tabel dan grafik dari pengambilan data yang dilakukan

**Tabel 14 Pengujian Sensor *Turbidity* sebelum menggunakan persamaan linear**

Percobaan	Turbidity Meter	Tegangan (mV)
1	1009	1985
2	938	2352,8
3	796	2682,3
4	644	2621
5	511	2915
6	387	3113,7

7	296	3481
8	208	3670,5
9	152	3731
10	101	3780
11	76	3844
12	60	3881
13	46	3919
14	35,73	3953
15	28,14	3983
16	21,75	4008
17	20,3	3985
18	18,6	3991

Tabel-tabel diatas merupakan pengujian sensor *turbidity* menggunakan Arduino UNO, melalui data diatas, dapat disimpulkan semakin cerah suatu larutan, maka semakin tinggi tegangan yang didapatkan. Begitupun sebaliknya semakin keruh suatu air semakin rendah tegangannya. Dari data diatas menunjukkan bahwa nilai sensor yang dibuat, masih jauh dengan nilai alat pembeding yang digunakan.



**Gambar 21 Grafik Persamaan linearitas *turbidity***

**Table 15 Pengambilan data turbidity setelah kalibrasi**

Percobaan	Turbidity Meter	Rata - rata Turbidity Sensor	Error (%)
1	1009	1042,69	3,34
2	938	854,36	8,92
3	796	685,27	13,91
4	644	716,72	11,29
5	511	565,52	10,67
6	387	463,83	19,85
7	296	275,17	7,04
8	208	177,99	14,43
9	152	146,71	3,48
10	101	121,8	20,59
11	76	88,85	16,91
12	60	69,95	16,58
13	46	50,21	9,15
14	35,73	32,98	7,70
15	28,14	17,43	38,06
16	21,75	4,72	78,30
17	18,6	13,41	27,90
18	20,37	16,59	18,56

Berikut merupakan tabel pengambilan data pada sensor *turbidity* setelah memasukkan persamaan garis linearitas pada program Arduino IDE.

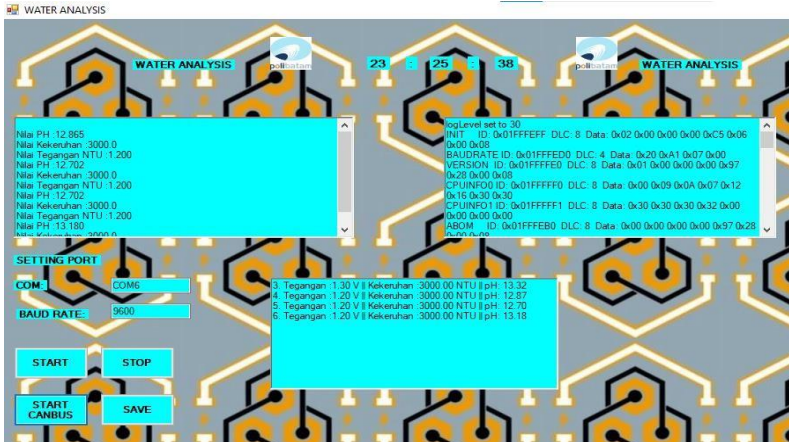
### 3. Pengujian Menggunakan Komunikasi dengan Hasil Data Sensor Tampil Pada Aplikasi

Tahap selanjutnya, masuk pada proses pengujian alat. Bagian ini akan menjelaskan hasil pengukuran dari sensor *pH* dan *turbidity* yang dikomunikasikan melalui berbagai metode seperti Serial 232, RS-485, CANBus, dan TCP/IP, serta bagaimana tampilan data disajikan pada aplikasi C#, dan web. Pembahasan ini memberikan informasi tentang kesiapan alat dalam memonitoring kualitas air menggunakan komunikasi.

#### A. Hasil Pengujian Data Sensor *pH* dan *Turbidity* Menggunakan Komunikasi CANBus

Gambar dibawah ini menunjukkan tampilan aplikasi C#, menggunakan

komunikasi *CANBus*. Terdapat kabel yang kompatibel dengan komunikasi *CANBus*. Data tampil secara *real time* pada aplikasi yang digunakan.



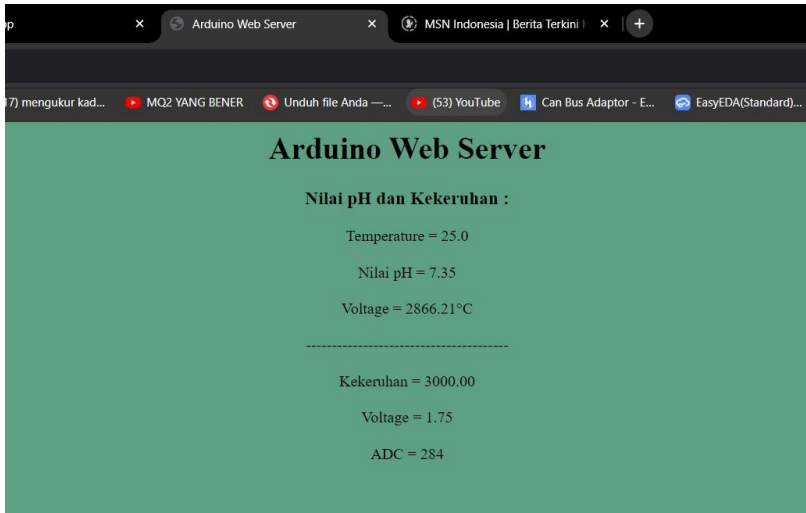
**Gambar 22 Tampilan pada C# menggunakan komunikasi *CANBus***

Ada beberapa bagian dalam tampilan C# diatas, diantara lain :

- Setting COM** : Mengacu pada *port* serial yang digunakan untuk berkomunikasi antara Arduino dan komputer melalui kabel perangkat keras komunikasi serial.
- Setting baudrate** : Mengatur *setting baudrate* pada komunikasi serial antara perangkat keras, seperti Arduino Uno, dan aplikasi C# sangat penting karena *baudrate* menentukan kecepatan mengirim data.
- Display text box** : Menampilkan teks, data sensor, hingga pada tampilan lainnya.
- Function Button** : *Button* digunakan untuk melakukan suatu aksi, atau menghubungkan suatu fungsi kepada suatu proses. Pada gambar diatas terdapat *button* untuk *start* sebagai memulai eksekusi data. *Button stop* untuk mengakhiri proses eksekusi data.

## **B. Hasil Pengujian Data Sensor *pH* dan *Turbidity* Menggunakan Komunikasi *TCP/IP***

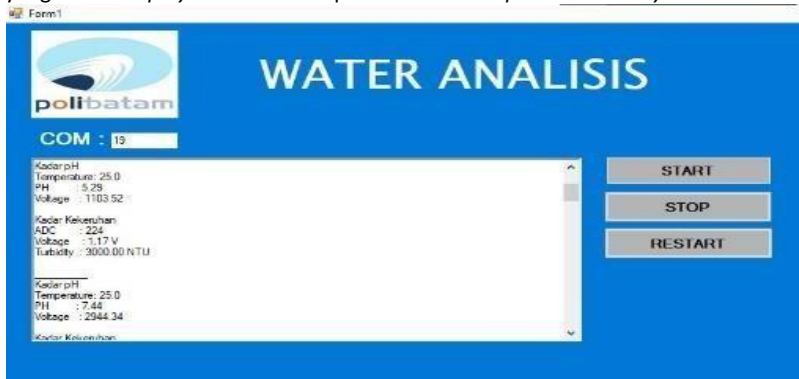
Terdapat kabel *LAN* berwarna kuning yang merupakan kabel yang kompatibel dengan komunikasi shield *TCP/IP*. Data tampil secara *real time*, namun, tidak sejalan dengan data yang tampil pada *LCD* .



Gambar 23 Tampilan *Web* Menggunakan Komunikasi *TCP/IP*

### C. Hasil Pengujian Data Sensor *pH* dan *Turbidity* Menggunakan Komunikasi Serial 232

Pada gambar dibawah, merupakan gambar dari tampilan pada aplikasi *C#*, yang dihubungkan melalui kabel *USB* yang sesuai dengan Serial 232. Tampilan data yang ada di *display C#* adalah tampilan data sensor *pH* dan *turbidity*.



Gambar 24 Tampilan Komunikasi Serial 232

Ada beberapa isi dalam tampilan *C#* seperti gambar diatas, diantara lainnya:

- a. *Setting COM* : *COM* mengacu pada *port* serial yang digunakan untuk berkomunikasi antara *Arduino* dan komputer melalui kabel perangkat keras komunikasi serial.

- b. *Display text box* : Menampilkan teks, data sensor, hingga pada tampilan lainnya.
- c. *Function Button* : *Button* digunakan untuk melakukan suatu aksi, atau menghubungkan suatu fungsi kepada suatu proses. Pada gambar diatas terdapat *button* untuk *start* sebagai memulai eksekusi data. *Button stop* untuk mengakhiri proses eksekusi data, dan *button restart* pengulangan.

**D. Hasil Pengujian Data Sensor pH dan Turbidity Menggunakan Komunikasi RS485**

Pada gambar dibawah, merupakan tampilan pada aplikasi C#, yang dihubungkan melalui kabel USB RS-485.



**Gambar 25 Tampilan Komunikasi RS-485**

Pada gambar diatas, merupakan gambar dari tampilan pada aplikasi C#, yang dihubungkan melalui kabel USB RS-485. Pada tampilan C#, terdapat *Button Start*, yang ditekan ketika ingin memulai keluaran data. *Button Restart* untuk memulai ulang keluaran data, dan *Button Stop* untuk memberhentikan keluaran data.

Ada beberapa isi dalam tampilan C# seperti gambar diatas, diantara lainnya:

- a. *Setting COM* : *COM* mengacu pada *port* serial yang digunakan untuk berkomunikasi antara Arduino dan komputer melalui kabel perangkat keras komunikasi serial.
- b. *Display text box* : Menampilkan data sensor
- c. *Function Button* : Pada gambar diatas terdapat *button* untuk *start* sebagai memulai eksekusi data. *Button stop* untuk mengakhiri proses eksekusi data, dan *button restart* pengulangan.

## Bab 5. Kesimpulan dan Saran

Berdasarkan hasil dari penelitian yang telah dilaksanakan, berikut adalah kesimpulan dan saran dari penelitian yang berjudul “Alat Monitoring Air dengan Menggunakan Multi Jalur Komunikasi Data”.

### A. Kesimpulan

Berdasarkan dengan penelitian yang sudah dilakukan, sub bab kesimpulan akan membawa kita pada titik akhir dari penelitian yang berjudul “Alat Monitoring Air dengan Menggunakan Jalur Multi Komunikasi Data “. Melalui setiap kegiatan, kami berupaya untuk membuat kesimpulan, berikut merupakan kesimpulan dari penelitian ini :

1. Mikrokontroler Arduino Uno digunakan sebagai perangkat keras, dan Arduino IDE sebagai perangkat lunak untuk melakukan akuisisi data. Dengan rata – rata error yang didapatkan dari masing – masing sensor, dengan pendekatan regresi linear. Pada sensor pH didapatkan rata – rata error sebesar 4,39 %, dan sensor turbidity dengan rata – rata error sebesar 18.15 %.
2. Data sensor telah terkirim melalui komunikasi data seperti Serial 232, *RS-485*, *CANBus*, dan *TCP/IP*. Pada komunikasi data Serial 232, *RS-485*, *CANBus* memiliki *COM* yang terdaftar pada perangkat penerima, dan komunikasi data *TCP/IP* pada *IP statis*(192.168.137.177).
3. Tampilan hasil data pengukuran sensor Ph, dan sensor turbidity tersaji pada LCD, aplikasi web, dan C# untuk setiap tipe komunikasi data.
4. Pembacaan sensor telah dilengkapi dengan proses kalibrasi dengan pendekatan regresi linear untuk tipe sensor yang berbeda. Dengan persamaan linear pada sensor *pH* yaitu  $y = -2.7342x + 13,489$ , sedangkan pada sensor turbidity menggunakan persamaan  $y = -0.5133x + 2062.1$

### B. Saran

Berdasarkan pembuatan alat yang telah dilakukan penulis berharap .alat ini dapat, meenggunakan push button sebagai metode kalibrasi pada sensor. Mengembangkan *GUI (graphical user interface)* untuk setiap tampilan komunikasi, dan dapat dikembangkan pada sistem SCADA.

## Daftar Pustaka

- [1] Yunior, Y. T. K., & Kusriani, K. (2021). Sistem Monitoring Kualitas Air Pada Budidaya Perikanan Berbasis IoT dan Manajemen Data. *Creative Information Technology Journal*, 6(2), 153-164. protokol TCP/IP. *Transient: Jurnal Ilmiah Teknik Elektro*, 2(3), pp.643- 650.
- [2] Taufik, A., & Fadhil, A. (2023). Sistem Monitoring pH dan Kekeruhan Kolam Ikan Koi Berbasis Internet of Things Menggunakan Aplikasi Blynk. *Jurnal Teknologi Elektro*, 14(01).
- [3] Zarkashie, M. F. (2021). *Rancang bangun sistem pengukuran kualitas air Untuk keperluan higiene sanitasi berbasis Arduino uno* (Bachelor's thesis Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta).
- [4] Wibi Hiryanto. (2021). Sistem Monitoring Kejernihan dan pH Air Berbasis Arduino
- [5] Mohammad Alfian Ikhsan. (2018). PENDETEKSI KEKERUHAN AIR DI TANDON RUMAH BERBASIS ARDUINO UNO
- [6] Ikram sufi. (2022). Mengenal Bagian - Bagian dan Features pada Papan Arduino UNO beserta Fungsinya Yuk
- [7] Jason Yang. (2019). Bagaimana RS232 Komunikasi Serial Bekerja
- [8] Nurul Septiani. (2024). Sistem Komunikasi Antar Arduino Menggunakan Potokol RS-485
- [9] SIMOR Technology. (2021). apa itu canbus?bisa kah arduino menggunakan protokol ini?
- [10] Kinda Yasar. (2024). Apa itu TCP/IP dan Bagaimana Cara Kerjanya?
- [11] Hery Suryantoro. (2019). PROTOTYPE SISTEM MONITORING LEVEL AIR BERBASIS LABVIEW & ARDUINO SEBAGAI SARANA PENDUKUNG PRAKTIKUM INSTRUMENTASI SISTEM KENDALI

## Biodata



Nama : Cesiia Margaritha BoruHarianja  
TTL : Tanjungpinang, 07 Maret 2003  
Agama : Kristen Protestan  
Alamat : Jalan Bunga Raya No.5a  
Email : [cesiliamargaritha0703@gmail.com](mailto:cesiliamargaritha0703@gmail.com)  
Riwayat Pendidikan SMA/SMK:SMAN2Tanjungpinang  
SMP : SMPN 5 Tanjungpinang



Nama : Alfaiz Ferdiansyah  
TTL : Batam, 31 Agustus 2002  
Agama : Islam  
Alamat : Perumahan Yose Sade Indah,Blok  
D No.2, Jalan Brigjen Katamso,  
Tanjung Uncang, Batu Aji  
Email : [alfaizferdi35@gmail.com](mailto:alfaizferdi35@gmail.com)  
Riwayat Pendidikan SMA/SMK : SMKN 5 Batam  
SMP : SMPN 38 Batam



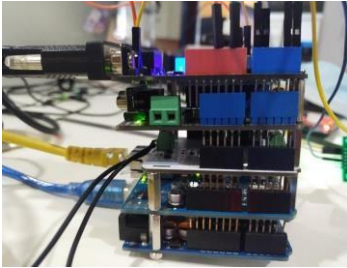
Nama : Ferdito Nois Firmansyah  
TTL : Dabo singkep,18 april 2003  
Agama : Islam  
Alamat : Tlk. Tering, Kec. Batam Kota, Kota  
Batam, Kepulauan Riau 29444  
Email : [ferditofirmansah@gmail.com](mailto:ferditofirmansah@gmail.com)  
Riwayat Pendidikan SMA/SMK : SMKN 1 SINGKEP  
SMP : SMPN 1 SINGKEP



Nama : Andhika Maulana  
TTL : Batam 07 September 2002  
Agama : Islam  
Alamat : Perumahan putra kelana jaya Blok  
A7 No 1  
Email : [andhikamaulanaa1702@gmail.com](mailto:andhikamaulanaa1702@gmail.com)  
Riwayat Pendidikan SMA/SMK : SMA NEGERI 8 BATAM  
SMP: SMP NEGERI 30 BATAM

## Lampiran

Lampiran 1 Tampilan bagian dalam



Lampiran 2 Tampilan Box yang sudah dirakit





## 1. Source Code

### A. Source Code Arduino UNO

```
//LiquidCrystal_I2C mylcd(0x20, 20, 4);//Proteus  
Simulation#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C mylcd(0x27, 20, 4);
```

```
#include "Arduino.h"  
#include  
<SoftwareSerial.h>
```

```

//RS 485 DAN RS232
#define RS485_RX_Pin 3 // Serial Receive pin
#define RS485_TX_Pin 2 // Serial Transmit pin
#define RS232_RX_Pin 5
#define RS232_TX_Pin 4

//TCP/IP

#include

<SPI.h>
#include <Ethernet.h>
// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local
network:byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
0xED}; IPAddress ip(192, 168, 137, 177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for
HTTP):EthernetServer
server(80);

//PH Sensor
#include
"DFRobot_PH.h"
#include <EEPROM.h>
#define PH_PIN A0
float voltage,phValue,temperature =
25;DFRobot_PH ph;

//Turbidity Sensor
#define Turbidity_Sensor_Pin
A1float
Turbidity_Sensor_Voltage; int
samples = 600;
float ntu; // Nephelometric Turbidity Units

SoftwareSerial RS485Serial(RS485_RX_Pin, RS485_TX_Pin); // RX,
TXSoftwareSerial RS232Serial(RS232_RX_Pin, RS232_TX_Pin);
int byteSend;

```

```

void setup() {
// put your setup code here, to run once:

Serial.begin(11520
0);ph.begin();
RS485Serial.begin(9600); // set the data rate
RS232Serial.begin(9600);
mylcd.init();
mylcd.backlight();
mylcd.setCursor(0,
0);delay(100);
mylcd.clear();
delay(500);

//Serial.println("Send PH Data!");
RS485Serial.print("RS485 Ready!"); // Send
messageRS232Serial.print("RS232 Ready!");

//ETHERNET
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}
Serial.println("Ethernet WebServer Example");
//start the Ethernet connection and the
server:Ethernet.begin(mac, ip);

// Check for Ethernet hardware present
if (Ethernet.hardwareStatus() == EthernetNoHardware) {
Serial.println("Ethernet shield was not found. Sorry, can't run without
hardware. :(");
while (true) {
delay(1); // do nothing, no point running without Ethernet hardware
}
}
if (Ethernet.linkStatus() == LinkOFF) {
Serial.println("Ethernet cable is not connected.");
}

// start the server
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP()

```

```

);
}

void loop() {
// put your main code here, to run repeatedly:
static unsigned long timepoint = millis();
if(millis()-timepoint>1000U){ //time interval: 1s
timepoint = millis();
//temperature = readTemperature(); // read your temperature
sensorto execute temperature compensation
voltage = analogRead(PH_PIN)/1024.0*5000; // read the voltage
phValue = ph.readPH(voltage,temperature); // convert voltage to pH with
temperature compensation
Serial.print("temperature:");
Serial.print(temperature,1);
Serial.print("^C pH:");
Serial.println(phValue,2);
}
SensorPH();
SensorTurbidity();
Send_Via_TCP_IP();

//Serial Monitor Print Value
/* Serial.print("Temperature:
");Serial.print(temperature,1);
Serial.print(" ^C PH: ");
Serial.print(phValue,2);
Serial.print(" Voltage: ");
Serial.println(voltage,2);
*/
Serial.print("temperature:");
Serial.print(temperature,1);
Serial.print("^C pH:");
Serial.println(phValue,2);
//LCD Print pH Sensor Value
mylcd.clear();
mylcd.setCursor(0, 0);
mylcd.print("Nilai pH : ");
mylcd.print(phValue,2);

//LCD Print Turbidity Sensor Value

```

```
mylcd.setCursor(0,1);
mylcd.print("Voltage : ");
mylcd.print(voltage);
mylcd.setCursor(0, 2);
mylcd.print("Turbidity : ");
mylcd.print(ntu,2);
```

```
//RS485 Send PH Sensor Value
RS485Serial.println("Kadar pH");
RS485Serial.print("Temperature: ");
RS485Serial.println(temperature,1);
RS485Serial.print("PH          :
"); RS485Serial.println(phValue,2);
RS485Serial.print("Voltage : ");
RS485Serial.println(voltage,2);
RS485Serial.println();
```

```
//RS485 Send Turbidity Sensor Value
RS485Serial.println("Kadar Kekeruhan");
RS485Serial.print("ADC          : ");
RS485Serial.println(analogRead(Turbidity_Sensor_Pin));
RS485Serial.print("Voltage : ");
RS485Serial.print(Turbidity_Sensor_Voltage);
RS485Serial.println(" V");
RS485Serial.print("Turbidity : ");
RS485Serial.print(ntu);
RS485Serial.println(" NTU");
RS485Serial.println();
RS485Serial.println("
_____")
);
```

```
//RS232 Send PH Sensor Value
RS232Serial.println("Kadar pH");
RS232Serial.print("Temperature: ");
RS232Serial.println(temperature,1);
RS232Serial.print("PH          :
");
RS232Serial.println(phValue,2);
RS232Serial.print("Voltage : ");
RS232Serial.println(voltage,2);
```

```

RS232Serial.println();

//RS232 Send Turbidity Sensor Value
RS232Serial.println("Kadar Kekeruhan");
RS232Serial.print("ADC          : ");
RS232Serial.println(analogRead(Turbidity_Sensor_Pin));
RS232Serial.print("Voltage : ");
RS232Serial.print(Turbidity_Sensor_Voltage);
RS232Serial.println(" V");
RS232Serial.print("Turbidity : ");
RS232Serial.print(ntu);
RS232Serial.println(" NTU");
RS232Serial.println();
RS232Serial.println("
_____")
);delay(1000);

ph.calibration(voltage,temperature);
Send_Via_TCP_IP();
}

float readTemperature()
{

}

float round_to_dp( float in_value, int decimal_place )
{
float multiplier = powf( 10.0f, decimal_place );
in_value = roundf( in_value * multiplier ) /
multiplier;return in_value;
}

void SensorPH()
{
//ph.calibration(voltage,temperature);           // calibration process by
SerialCMD

static unsigned long timepoint = millis();
if(millis()-timepoint>1000U){           //time interval: 1s
timepoint = millis();
//temperature = readTemperature();           // read your temperature

```

```

sensorto execute temperature compensation
voltage = analogRead(PH_PIN)/1024.0*5000; // read the voltage
pHValue = ph.readPH(voltage,temperature); // convert voltage to pH with
temperature compensation
Serial.print("temperature:");
Serial.print(temperature,1);
Serial.print("^C pH:");
Serial.println(pHValue,2);
}
}

void SensorTurbidity()
{
Turbidity_Sensor_Voltage = 0;

for(int i=0; i<samples; i++)
{
Turbidity_Sensor_Voltage      +=
(float)analogRead(Turbidity_Sensor_Pin)/1023)*5;
}

Turbidity_Sensor_Voltage = Turbidity_Sensor_Voltage/samples;

Turbidity_Sensor_Voltage = round_to_dp(Turbidity_Sensor_Voltage,2);
if(Turbidity_Sensor_Voltage < 2.5){
ntu = 3000;
}

else
{
ntu          =      -1120.4*sqrt(Turbidity_Sensor_Voltage)+
5742.3*Turbidity_Sensor_Voltage - 4352.9;
}

}

void Send_Via_TCP_IP()
{
//Send PH and Tubidity Value via TCP/IP
// listen for incoming clients
EthernetClient client =
server.available();if (client)

```

```

{
Serial.println("new client");
// an HTTP request ends with a blank line
bool currentLineIsBlank = true;
while (client.connected())
{
if (client.available())
{
char c = client.read();
Serial.write(c);
// if you've gotten to the end of the line (received a newline
// character) and the line is blank, the HTTP request has ended,
// so you can send a reply
if (c == '\n' && currentLineIsBlank)
{
// send a standard HTTP response header
client.println();
client.println();
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html\n");
client.println("<html><head><META                HTTP-EQUIV=""refresh""
CONTENT=""5"">\n");
client.println("<center><title>Arduino Web Server</title></head>");
client.println("<body style=""background-color:#5EA084"">\n");
client.println("<h1>Arduino Web Server</h1>"); client.println("<h3>Nilai
pH dan Kekерuhan : </h3>");

//pH client.print("<h3></h3>Temperature =
");client.print(temperature,1);
client.print("<h3></h3>Nilai pH = ");
client.print(pHValue,2);

//Tegangan
client.print("<h3></h3>Voltage = ");
client.print(voltage);
client.print("&deg;C");
client.print("<h3></h3>                ");
//Turbidity Sensor
client.print("<h3></h3>Kekeruhan = ");
client.println(ntu); client.print("<h3></h3>Voltage =
"); client.println(Turbidity_Sensor_Voltage);
client.print("<h3></h3>ADC = ");

```

```

client.println(analogRead(Turbidity_Sensor_Pin));

// output the value of each analog input pin
static unsigned long timepoint = millis();
if(millis()-timepoint>1000U)
{
    //time interval: 1timepoint = millis();

    //temperature = readTemperature();           // read your temperature
    sensorto execute temperature compensation
    //voltage = analogRead(PH_PIN)/1024.0*5000; // read the voltage
    //phValue = ph.readPH(voltage,temperature); // convert voltage to pHwith
    temperature compensation
}
ph.calibration(voltage,temperature);// calibration process by Serail
CMD
}

client.println("<center></html>"); break;

if (c == '\n') {
// you're starting a new line currentLineIsBlank = true;
}
else if (c != '\r')
{
// you've gotten a character on the current line currentLineIsBlank = false;
}
}
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
Serial.println("client disconnected");
}
}

```

## B. Source Code Tampilan Software C#

```

using System;
using System.Collections.Generic;using System.ComponentModel; using
System.Data;
using System.Drawing;using System.Linq; using System.Text;
using System.Windows.Forms;using System.IO;
namespace water

```

```

{
public partial class Form1 : Form
{
string sDataMasuk;public Form1()
{
InitializeComponent();
}
private void buttonSTART_Click(object sender, EventArgs e)
{
String sNamaPort = textBox1.Text; serialPort1.PortName = "COM" + sNamaPort;
try
{
serialPort1.Open();
}
catch
{
MessageBox.Show("Com tidak ada atau dipakai aplikasi lain");
}
}
private void serialPort1_DataReceived(object sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
sDataMasuk = serialPort1.ReadLine(); sDataMasuk = sDataMasuk.Trim();
this.Invoke(new EventHandler(TampilkanData));
}
private void TampilkanData(object sender, EventArgs e)
{
richTextBox1.Text += sDataMasuk + "\n";
}
private void buttonSTOP_Click(object sender, EventArgs e)
{
serialPort1.Close();
}
private void buttonRestart_Click(object sender, EventArgs e)
{
richTextBox1.Clear();
}
}
}
}

```

### Lampiran 3 Data pengujian sensor pH (sample pH 1)

pH 1	Tegangan (mV)	pH Meter
0,9	3440,86	1,3
0,9	3440,86	1,3
0,93	3435,97	1,3
0,93	3435,97	1,3
0,97	3431,08	1,3
0,97	3431,08	1,3
0,97	3431,08	1,3
1	3426,2	1,3
1	3426,2	1,3
1,03	3421,31	1,3
1,03	3421,31	1,3
1,07	3416,42	1,3
1,1	3411,53	1,3
1,14	3406,65	1,3
1,14	3406,65	1,3
1,14	3406,65	1,3
1,14	3406,65	1,3
1,17	3401,76	1,3
1,17	3401,76	1,3
1,17	3401,76	1,3
1,04	3420,09	1,30

### Lampiran 4 Data pengujian sensor pH (sample pH 2)

pH 2	Tegangan (mV)	pH Meter
2,17	3260,02	2,33
2,21	3255,13	2,33
2,21	3255,13	2,33

2,24	3250,24	2,33
2,24	3250,24	2,33
2,28	3245,36	2,33
2,28	3245,36	2,33
2,28	3245,36	2,33
2,28	3245,36	2,33
2,28	3245,36	2,33
2,28	3245,36	2,33
2,31	3240,47	2,33
2,31	3240,47	2,33
2,34	3235,58	2,33
2,34	3235,58	2,33
2,38	3230,69	2,33
2,38	3230,69	2,33
2,41	3225,81	2,33
2,41	3225,81	2,33
2,45	3220,92	2,33
2,45	3220,92	2,33
2,31	3240,23	2,33

**Lampiran 5 Data pengujian sensor pH (sample pH 3)**

pH3	Tegangan (mV)	pH Meter
3,28	3103,62	3,73
3,28	3103,62	3,73
3,31	3098,73	3,73
3,34	3093,84	3,73
3,34	3093,84	3,73
3,34	3093,84	3,73
3,38	3088,95	3,73
3,38	3088,95	3,73
3,38	3088,95	3,73

3,41	3084,07	3,73
3,48	3074,29	3,73
3,48	3074,29	3,73
3,48	3074,29	3,73
3,48	3074,29	3,73
3,48	3074,29	3,73
3,52	3069,4	3,73
3,52	3069,4	3,73
3,55	3064,52	3,73
3,59	3059,63	3,73
3,66	3049,85	3,73
3,43	3081,13	3,73

**Lampiran 6 Data pengujian sensor pH (sample pH 4)**

pH4	Tegangan (mV)	pH Meter
3,9	3015,64	4,16
3,93	3010,75	4,16
3,93	3010,75	4,16
3,97	3005,87	4,16
4	3000,98	4,16
4	3000,98	4,16
4	3000,98	4,16
4	3000,98	4,16
4	3000,98	4,16
4	3000,98	4,16
4	3000,98	4,16
4	3000,98	4,16
4,07	2991,2	4,16
4,07	2991,2	4,16
4,07	2991,2	4,16
4,1	2986,31	4,16

4,1	2986,31	4,16
4,14	2981,43	4,16
4,17	2976,54	4,16
4,21	2971,65	4,16
4,21	2971,65	4,16
4,04	2994,87	4,16

**Lampiran 7 Data pengujian sensor pH (sample pH 5)**

pH 5	Tegangan (mV)	pH Meter
7,28	2536,66	6,65
7,31	2531,77	6,65
7,38	2521,99	6,65
7,41	2517,11	6,65
7,41	2517,11	6,65
7,45	2512,22	6,65
7,48	2507,33	6,65
7,48	2507,33	6,65
7,48	2507,33	6,65
7,52	2502,44	6,65
7,52	2502,44	6,65
7,59	2492,67	6,65
7,59	2492,67	6,65
7,59	2492,67	6,65
7,59	2492,67	6,65
7,66	2482,89	6,65
7,66	2482,89	6,65
7,66	2482,89	6,65
7,69	2478,01	6,65
7,72	2473,12	6,65
7,52	2501,71	6,65

**Lampiran 8 Data pengujian sensor pH (sample pH 6)**

pH 6	Tegangan (mV)	pH Meter
7,24	2541,54	7,3
7,28	2536,66	7,3
7,31	2531,77	7,3
7,31	2531,77	7,3
7,31	2531,77	7,3
7,34	2526,88	7,3
7,34	2526,88	7,3
7,38	2521,99	7,3
7,38	2521,99	7,3
7,38	2521,99	7,3
7,41	2517,11	7,3
7,41	2517,11	7,3
7,45	2512,22	7,3
7,45	2512,22	7,3
7,52	2502,44	7,3
7,52	2502,44	7,3
7,52	2502,44	7,3
7,52	2502,44	7,3
7,52	2502,44	7,3
7,52	2502,44	7,3
7,55	2497,56	7,3
7,41	2518,08	7,30

**Lampiran 9 Data pengujian sensor pH (sample pH 7)**

pH 7	Tegangan (mV)	pH Meter
7,14	2556,21	7,19
7,14	2556,21	7,19

7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,17	2551,32	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,21	2546,43	7,19
7,24	2541,54	7,19
7,19	2549,364	7,19

**Lampiran 10 Data pengujian sensor pH (sample pH 8)**

pH 8	Tegangan (mV)	pH Meter
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59
7,69	2478,01	7,59



8,03	2429,13	7,95
8,03	2429,13	7,95
8,07	2424,24	7,95
8,07	2424,24	7,95
8,07	2424,24	7,95
8,1	2419,3	7,95
8,01	2432,30	7,95

**Lampiran 12 Data pengujian sensor pH (sample pH 10)**

pH 10	Tegangan (mV)	pH Meter
9,48	2223,85	9,19
9,48	2223,85	9,19
9,52	2218,96	9,19
9,52	2218,96	9,19
9,52	2218,96	9,19
9,59	2209,19	9,19
9,52	2218,96	9,19
9,52	2218,96	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19
9,52	2218,96	9,19
9,52	2218,96	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19
9,55	2214,08	9,19

9,53	2216,52	9,19
------	---------	------

**Lampiran 13 Data pengujian sensor pH (sample pH 11)**

pH 11	Tegangan (mV)	pH Meter
8,55	2355,82	9,17
8,62	2346,04	9,17
8,69	2336,27	9,17
8,76	2326,49	9,17
8,76	2326,49	9,17
8,86	2311,83	9,17
8,86	2311,83	9,17
8,93	2302,05	9,17
8,97	2297,17	9,17
8,97	2297,17	9,17
9,03	2287,39	9,17
9,03	2287,39	9,17
9,07	2282,5	9,17
9,07	2282,5	9,17
9,14	2272,73	9,17
9,14	2272,73	9,17
9,14	2272,73	9,17
9,17	2267,84	9,17
9,17	2267,84	9,17
9,17	2267,84	9,17
8,96	2298,63	9,17

**Lampiran 14 Data pengujian sensor pH (sample pH 12)**

pH 12	Tegangan (mV)	pH Meter
10,21	2121,21	10,17

10,31	2106,55	10,17
10,31	2106,55	10,17
10,31	2106,55	10,17
10,31	2106,55	10,17
10,34	2101,66	10,17
10,34	2101,66	10,17
10,34	2101,66	10,17
10,34	2101,66	10,17
10,34	2101,66	10,17
10,38	2096,77	10,17
10,38	2096,77	10,17
10,38	2096,77	10,17
10,38	2096,77	10,17
10,38	2096,77	10,17
10,38	2096,77	10,17
10,38	2096,77	10,17
10,41	2091,89	10,17
10,41	2091,89	10,17
10,41	2091,89	10,17
10,41	2091,89	10,17
10,35	2100,195	10,17

**Lampiran 15 Data pengujian sensor pH (sample pH 13)**

pH 13	Tegangan (mV)	pH Meter
12,1	1852,39	13,11
12,34	1818,18	13,11
12,41	1808,41	13,11
12,97	1730,21	13,11
13,17	1700,88	13,11
13,38	1671,55	13,11
13,41	1666,67	13,11

13,55	1647,12	13,11
13,55	1647,12	13,11
13,55	1647,12	13,11
13,55	1647,12	13,11
13,55	1647,12	13,11
13,59	1642,23	13,11
13,66	1632,45	13,11
13,69	1627,57	13,11
13,69	1627,57	13,11
13,69	1627,57	13,11
13,69	1627,57	13,11
13,69	1627,57	13,11
13,69	1627,57	13,11
13,69	1627,57	13,11
13,35	1676,20	13,11

**Lampiran 16 Data pengujian sensor pH (sample pH 14)**

pH 14	Tegangan (mV)	pH Meter
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,76	1617,79	13,31
13,79	1612,9	13,31

13,79	1612,9	13,31
13,79	1612,9	13,31
13,79	1612,9	13,31
13,79	1612,9	13,31
13,79	1612,9	13,31
13,79	1612,9	13,31
13,79	1612,9	13,31
13,79	1612,9	13,31
13,77	1615,83	13,31

**Lampiran 17 Data pengujian sensor Turbidity (Percobaan 1)**

PERCOBAAN 1		
Turbidity meter	Tegangan Sensor	NTU Sensor
1009,00	1984,36	1043,53
1009,00	1984,36	1043,53
1009,00	1979,47	1046,04
1009,00	1984,36	1043,53
1009,00	1984,36	1043,53
1009,00	1984,36	1043,53
1009,00	1984,36	1043,53
1009,00	1989,25	1041,02
1009,00	1989,25	1041,02
1009,00	1989,25	1041,02
1009,00	1989,25	1041,02
1009,00	1989,25	1041,02
1009,00	1984,36	1043,53
1009,00	1989,25	1041,02
1009,00	1984,36	1043,53
1009,00	1985,99	1042,69

**Lampiran 18 Data pengujian sensor Turbidity (Percobaan 2)**

PERCOBAAN 2
-------------

Turbidity meter	Tegangan Sensor	NTU Sensor
938,00	2346,04	857,88
938,00	2350,93	855,37
938,00	2346,04	857,88
938,00	2350,93	855,37
938,00	2350,93	855,37
938,00	2350,93	855,37
938,00	2350,93	855,37
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2355,82	852,86
938,00	2352,89	854,36

**Lampiran 19 Data pengujian sensor Turbidity (Percobaan 3)**

PERCOBAAN 3		
Turbidity meter	Tegangan Sensor	NTU Sensor
796,00	2683,28	684,77
796,00	2683,28	684,77
796,00	2683,28	684,77
796,00	2688,17	682,26
796,00	2683,28	684,77
796,00	2688,17	682,26
796,00	2688,17	682,26
796,00	2678,40	687,28
796,00	2683,28	684,77
796,00	2683,28	684,77
796,00	2678,40	687,28

796,00	2678,40	687,28
796,00	2678,40	687,28
796,00	2678,40	687,28
796,00	2678,40	687,28
796,00	2682,31	685,27

**Lampiran 20 Data pengujian sensor Turbidity (Percobaan 4)**

PERCOBAAN 4		
Turbidity meter	Tegangan Sensor	NTU Sensor
644,00	2609,97	722,40
644,00	2624,63	714,88
644,00	2590,42	732,44
644,00	2609,97	722,40
644,00	2609,97	722,40
644,00	2624,63	714,88
644,00	2619,75	717,38
644,00	2605,08	724,91
644,00	2624,63	714,88
644,00	2619,75	717,38
644,00	2624,63	714,88
644,00	2629,52	712,37
644,00	2639,30	707,35
644,00	2644,18	704,84
644,00	2639,30	707,35
644,00	2621,05	716,72

**Lampiran 21 Data pengujian sensor Turbidity (Percobaan 5)**

PERCOBAAN 5		
Turbidity meter	Tegangan Sensor	NTU Sensor
511,00	2849,46	599,47
511,00	2849,46	599,47
511,00	2849,46	599,47

511,00	2873,90	586,93
511,00	2849,46	599,47
511,00	2913,00	566,86
511,00	2898,34	574,38
511,00	2937,44	554,31
511,00	2913,00	566,86
511,00	2942,33	551,80
511,00	2961,88	541,77
511,00	2976,54	534,24
511,00	2971,65	536,75
511,00	2971,65	536,75
511,00	2976,54	534,24
511,00	2915,61	565,52

Lampiran 22 Data pengujian sensor Turbidity (Percobaan 6)

PERCOBAAN 6		
Turbidity meter	Tegangan Sensor	NTU Sensor
387,00	3108,50	466,51
387,00	3108,50	466,51
387,00	3108,50	466,51
387,00	3113,39	464,00
387,00	3113,39	464,00
387,00	3113,39	464,00
387,00	3113,39	464,00
387,00	3113,39	464,00
387,00	3113,39	464,00
387,00	3113,39	464,00
387,00	3118,28	461,49
387,00	3113,39	464,00
387,00	3118,28	461,49
387,00	3113,39	464,00
387,00	3118,28	461,49
387,00	3118,28	461,49

387,00	3113,72	463,83
--------	---------	--------

**Lampiran 23 Data pengujian sensor Turbidity (Percobaan 7)**

PERCOBAAN 7		
Turbidity meter	Tegangan Sensor	NTU Sensor
296,00	3475,07	278,35
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3479,96	275,84
296,00	3484,85	273,33
296,00	3484,85	273,33
296,00	3484,85	273,33
296,00	3484,85	273,33
296,00	3484,85	273,33
296,00	3481,26	275,17

**Lampiran 24 Data pengujian sensor Turbidity (Percobaan 8)**

PERCOBAAN 8		
Turbidity meter	Tegangan Sensor	NTU Sensor
208,00	3665,69	180,50
208,00	3670,58	177,99
208,00	3670,58	177,99
208,00	3670,58	177,99
208,00	3670,58	177,99

208,00	3670,58	177,99
208,00	3670,58	177,99
208,00	3675,46	175,49
208,00	3675,46	175,49
208,00	3675,46	175,49
208,00	3675,46	175,49
208,00	3670,58	177,99
208,00	3665,69	180,50
208,00	3665,69	180,50
208,00	3665,69	180,50
208,00	3670,58	177,99

**Lampiran 25 Data pengujian sensor Turbidity (Percobaan 9)**

PERCOBAAN 9		
Turbidity meter	Tegangan Sensor	NTU Sensor
152,00	3734,12	145,38
152,00	3729,23	147,89
152,00	3729,23	147,89
152,00	3734,12	145,38
152,00	3734,12	145,38
152,00	3734,12	145,38
152,00	3734,12	145,38
152,00	3734,12	145,38
152,00	3729,23	147,89
152,00	3729,23	147,89
152,00	3729,23	147,89
152,00	3729,23	147,89
152,00	3729,23	147,89
152,00	3729,23	147,89
152,00	3734,12	145,38
152,00	3731,51	146,71

**Lampiran 26 Data pengujian sensor Turbidity (Percobaan 10)**

PERCOBAAN 10		
Turbidity meter	Tegangan Sensor	NTU Sensor
101,00	3782,99	120,29
101,00	3778,10	122,80
101,00	3778,10	122,80
101,00	3778,10	122,80
101,00	3778,10	122,80
101,00	3773,22	125,31
101,00	3778,10	122,80
101,00	3778,10	122,80
101,00	3778,10	122,80
101,00	3782,99	120,29
101,00	3782,99	120,29
101,00	3787,88	117,78
101,00	3782,99	120,29
101,00	3778,10	122,80
101,00	3782,99	120,29
101,00	3780,06	121,80

**Lampiran 27 Data pengujian sensor Turbidity (Percobaan 11)**

PERCOBAAN 11		
Turbidity meter	Tegangan Sensor	NTU Sensor
76,00	3841,64	90,19
76,00	3841,64	90,19
76,00	3846,53	87,68
76,00	3841,64	90,19
76,00	3846,53	87,68

76,00	3846,53	87,68
76,00	3846,53	87,68
76,00	3846,53	87,68
76,00	3841,64	90,19
76,00	3841,64	90,19
76,00	3841,64	90,19
76,00	3841,64	90,19
76,00	3846,53	87,68
76,00	3846,53	87,68
76,00	3846,53	87,68
76,00	3844,25	88,85

**Lampiran 28 Data pengujian sensor Turbidity (Percobaan 12)**

PERCOBAAN 12		
Turbidity meter	Tegangan Sensor	NTU Sensor
60,00	3885,63	67,61
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3880,74	70,12
60,00	3881,07	69,95

**Lampiran 29 Data pengujian sensor Turbidity (Percobaan 13)**

PERCOBAAN 13		
Turbidity meter	Tegangan Sensor	NTU Sensor
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3914,96	52,55
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,84	50,05
46,00	3919,51	50,21

**Lampiran 30 Data pengujian sensor Turbidity (Percobaan 14)**

PERCOBAAN 14		
Turbidity meter	Tegangan Sensor	NTU Sensor
35,73	3954,06	32,48
35,73	3954,06	32,48
35,73	3954,06	32,48
35,73	3954,06	32,48
35,73	3949,17	34,99
35,73	3949,17	34,99
35,73	3949,17	34,99
35,73	3949,17	34,99



Turbidity meter	Tegangan Sensor	NTU Sensor
21,75	4012,71	2,38
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4007,82	4,89
21,75	4008,15	4,72

**Lampiran 33 Data pengujian sensor Turbidity (Percobaan 17)**

PERCOBAAN 17		
Turbidity meter	Tegangan Sensor	NTU Sensor
18,60	3998,05	9,90
18,60	3993,16	12,41
18,60	3988,27	14,92
18,60	3998,05	9,90
18,60	3993,16	12,41
18,60	3993,16	12,41
18,60	3993,16	12,41
18,60	3988,27	14,92
18,60	3988,27	14,92
18,60	3993,16	12,41
18,60	3988,27	14,92

18,60	3988,27	14,92
18,60	3988,27	14,92
18,60	3988,27	14,92
18,60	3988,27	14,92
18,60	3991,20	13,41

**Lampiran 34 Data pengujian sensor Turbidity (Percobaan 18)**

PERCOBAAN 18		
Turbidity meter	Tegangan Sensor	NTU Sensor
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3988,27	14,92
20,37	3988,27	14,92
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3988,27	14,92
20,37	3988,27	14,92
20,37	3988,27	14,92
20,37	3983,38	17,43
20,37	3983,38	17,43
20,37	3985,01	16,59

## Lampiran 35 Tampilan pengirim menggunakan komunikasi serial 232

```
Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5')
New Line 9600 baud

Kadar pH
Temperature: 25.0
PH : 5.56
Voltage : 2793.20

Kadar Kekurangan
ADC : 258
Voltage : 1.72 V
Turbidity : 3000.00 NTU

Kadar pH
Temperature: 25.0
PH : 5.63
Voltage : 2773.44

Kadar Kekurangan
ADC : 233
Voltage : 1.73 V
Turbidity : 3000.00 NTU

Kadar pH
Temperature: 25.0
PH : 5.42
Voltage : 2822.73

Activate Windows
Go to Settings to activate Windows.
```

## Lampiran 36 Tampilan penerima menggunakan komunikasi serial 232

```
Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM5')
New Line 9600 baud

Kadar pH
Temperature: 25.0
PH : 5.56
Voltage : 2793.20

Kadar Kekurangan
ADC : 258
Voltage : 1.72 V
Turbidity : 3000.00 NTU

Kadar pH
Temperature: 25.0
PH : 5.63
Voltage : 2773.44

Kadar Kekurangan
ADC : 283
Voltage : 1.71 V
Turbidity : 3000.00 NTU

Kadar pH
Temperature: 25.0
PH : 5.42
Voltage : 2802.73

Kadar Kekurangan
ADC : 246

In 10, Col 1 Arduino Uno on COM5
```

**FORMULIR LOGBOOK BIMBINGAN DAN PENGAJUAN  
SIDANG TUGAS AKHIR**

Nama : Ferdho Nais Firmansyah  
 NIM : 3232101060  
 Pembimbing : Muhammad Syager Ghozali S.T., M.T.  
 Judul : Water Analysis Berbasis Komunikasi

No	Hari/Tgl	Rincian Kegiatan	TTD Pembimbing
1	Kamis 21-09-23	Melakukan komunikasi menggunakan metode master dan slave pada RS232 dan RS485	
2	Jumat 29-09-23	Membuat tampilan C# untuk komunikasi serial	
3	Senin 9/10/23	Data dari turjidi sangat jauh dari hasil pembimbing dan bertawar apa saja yang ditampilkan ke C# dan led	
4	Kamis 14/10/23	mengembangkan codingan antara komunikasi dan tombol kalibrasi	
5	Kamis 21/10/23	menyempurnakan komunikasi antara bertawar dan gambar selanjutnya	
6	Rabu 1/11/23	bertanya tentang hasil pembalasan sensor turjidi/ masih jauh sama pembimbing	
7	Jumat 6/11/23	Bertanya tentang komunikasi RS232, RS485, ethernet	
8	Kamis 9/11/23	masih berkehadapan keluar data ke led	
9	Selasa 14/11/23	tampilan dari 2 komunikasi yaitu RS232 dan RS485 ke aplikasi C#	
10	Senin 19/11/23	Buku akhir dan pengecekan data hingga pada Paper.	

Berdasarkan hasil bimbingan yang telah dilaksanakan selama 3 bulan dan telah disetujui oleh dosen pembimbing, maka dengan ini saya mengajukan diri sebagai peserta Sidang Tugas Akhir.

Batam, 17 November 2023  
 Peserta

Ferdho Nais Firmansyah  
 NIM: 3232101060