



TESIS - EE185401

**PENGEMBANGAN ALAT PENGUKUR PH BERBASIS  
OPTIKAL FLUORESEN DENGAN PENGUKURAN  
YANG CEPAT DAN AKURAT, MURAH DAN  
NIRKABEL UNTUK APLIKASI PADA CAIRAN**

RAHMAT FAUZI SIREGAR  
07111750042002

DOSEN PEMBIMBING  
Dr.Ir. Totok Mujiono, M.Kom.

PROGRAM MAGISTER  
BIDANG KEAHLIAN ELEKTRONIKA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2020



TESIS - EE185401

**PENGEMBANGAN ALAT PENGUKUR PH BERBASIS  
OPTIKAL FLUORESEN DENGAN PENGUKURAN  
YANG CEPAT DAN AKURAT, MURAH DAN  
NIRKABEL UNTUK APLIKASI PADA CAIRAN**

RAHMAT FAUZI SIREGAR  
07111750042002

DOSEN PEMBIMBING  
Dr.Ir. Totok Mujiono, M.Kom.

PROGRAM MAGISTER  
BIDANG KEAHLIAN ELEKTRONIKA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2020





## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar

**Magister Teknik (MT)**

di

**Institut Teknologi Sepuluh Nopember**

Oleh:

**RAHMAT FAUZI SIREGAR**

**NRP: 07111750042002**

Tanggal Ujian: 29 Juni 2020

Periode Wisuda: September 2020

Disetujui oleh:

**Pembimbing:**

1. Dr. Ir. Totok Mujiono, M.Kom  
NIP: 196504221989031001



**Penguji:**

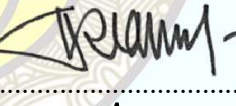
1. Dr. Ir. Hendra Kusuma, M.Eng.Sc.  
NIP: 196409021989031003



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



3. Dr. Tri Arief Sardjono, ST., MT.  
NIP: 197002121995121001



4. Astria Nur Irfansyah, ST., M.Eng, Ph.D.  
NIP: 198103252010121002



Kepala Departemen Teknik Elektro  
Fakultas Teknologi Elektro



Dedet Candra Riawan, S.T., M.Eng., Ph.D.  
NIP. 197311192000031001

*Halaman ini sengaja dikosongkan*

## **PERNYATAAN KEASLIAN TESIS**

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul **“PENGEMBANGAN ALAT PENGUKUR PH BERBASIS OPTIKAL FLUORESEN DENGAN PENGUKURAN YANG CEPAT DAN AKURAT, MURAH DAN NIRKABEL UNTUK APLIKASI PADA CAIRAN”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

Surabaya, 9 Juni 2020



Rahmat Fauzi Siregar

NRP. 07111750042002

*Halaman ini sengaja dikosongkan*



# **PENGEMBANGAN ALAT PENGUKUR PH BERBASIS OPTIKAL FLUORESEN DENGAN PENGUKURAN YANG CEPAT DAN AKURAT, MURAH DAN NIRKABEL UNTUK APLIKASI PADA CAIRAN**

Nama mahasiswa : Rahmat Fauzi Siregar  
NRP : 07111750042002  
Pembimbing : Dr. Ir. Totok Mujiono, M.IKom.

## **ABSTRAK**

Tingkat asam atau basa dalam air berdasarkan larutan (pH) adalah ukuran yang sangat penting bagi makhluk hidup karena sekitar 70% dalam tubuh terdiri dari air. Sebagian besar metabolisme dalam tubuh membutuhkan tingkat pH tertentu. Memiliki pH meter yang cepat dan akurat sangat dibutuhkan, tetapi sebagian besar pH meter yang tersedia membutuhkan waktu beberapa menit untuk mengukur pH suatu cairan. Oleh sebab itu, dilakukan pengembangan alat pengukur pH menggunakan metode fluoresen optik. Pewarna fluoresen berjenis pyranine digunakan dalam pengembangan alat pengukur pH ini. Pyranine diperoleh dengan mengekstraksi stabilo kuning menggunakan isopropil alkohol. Pengukuran dilakukan dengan mencampur sampel cairan dengan cairan fluoresen dan kemudian dieksitasi dengan sumber cahaya violet pada panjang gelombang 405 nm. Intensitas fluoresen hijau yang dipancarkan dari sampel cairan kemudian ditangkap oleh sensor spektral AS7262. Pengembangan alat pengukur pH berbasis fluoresen optik ini memiliki keunggulan dibandingkan dengan metode lain dalam hal waktu pengukuran dan hanya menggunakan sampel cairan yang sangat sedikit. Alat pengukur pH yang telah dikembangkan, diuji dan diverifikasi untuk dapat mengukur pH dari kisaran 4 hingga 11 dengan akurasi 98,13%, nilai kesalahan pembacaan  $\pm 0,13$  dan hanya membutuhkan waktu kurang dari 3 detik untuk melakukan pengukuran. Seluruh pengujian dilakukan pada suhu normal sekitar 25°C hingga 30°C dengan kondisi kedap pencahayaan.

Kata kunci: Alat pengukur pH, pH, Pewarna Fluoresen

*Halaman ini sengaja dikosongkan*

# **DEVELOPMENT OF PH SENSING DEVICES BASED ON OPTICAL FLUORESCENTS WITH RAPID AND ACCURATE MEASUREMENTS, LOW COST AND WIRELESS FOR LIQUID APPLICATIONS**

By : Rahmat Fauzi Siregar  
Student Identity Number : 07111750042002  
Supervisor(s) : Dr. Ir. Totok Mujiono, M.IKom.

## **ABSTRACT**

The level of acid or base in water based on solution (pH) is a very important measure for living things because about 70% of the body consists of water. Most of the metabolism in the body requires a certain pH level. Having a fast and accurate pH meter is needed, but most of the available pH meters take several minutes to measure the pH of a liquid. Therefore, the development of pH sensing devices using optical fluorescent methods was developed. Pyranine type fluorescent dyes are used in the development of this pH sensing device. Pyranine is obtained by extracting yellow highlighter using isopropyl alcohol. Measurements were made by mixing a liquid sample with a fluorescent liquid and then excited with a violet light source at a wavelength of 405 nm. The intensity of the green fluorescent emitted from the liquid sample is then captured by the AS7262 spectral sensor. The development of optical fluorescent-based pH measuring devices has the advantage over other methods in terms of measurement time and only uses very few liquid samples. A pH sensing device that has been developed, tested and verified to be able to measure pH from a range of 4 to 11 with an accuracy of 98.13%, a reading error value of  $\pm 0.13$  and only takes less than 3 seconds to take measurements. All tests are carried out at ambient normal temperature 25°C to 30°C with light-proof conditions.

Key words: pH Sensing Device, pH, Fluorescent Dye

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kepada Allah SWT, karena atas segala nikmat-Nya tesis ini dapat diselesaikan. Tesis berjudul “PENGEMBANGAN ALAT PENGUKUR PH BERBASIS OPTIKAL FLUORESEN DENGAN PENGUKURAN YANG CEPAT DAN AKURAT, MURAH DAN NIRKABEL UNTUK APLIKASI PADA CAIRAN” ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Magister Teknik (M.T.) pada Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Penulis menyadari bahwa dalam penyusunan tesis ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih kepada:

1. Dr. Ir. Totok Mujiono, M.IKom., selaku dosen pembimbing yang telah banyak memberikan saran, bantuan serta bimbingan dalam penelitian penulis.
2. Dosen Penguji.
3. Kepada Koordinator Pascasarjana Teknik Elektronika, Koordinator Pascasarjana Jurusan Teknik Elektro dan Ketua Jurusan Teknik Elektro serta karyawan Pascasarjana Teknik Elektro yang telah membantu penulis dalam segala urusan administrasi selama menempuh kuliah di ITS.
4. Kepada seluruh Civitas Akademisi Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya, atas segala bantuan yang telah diberikan.
5. Kepada Ibu, Kakak, Adik dan kerabat keluarga yang senantiasa memberikan dukungan serta doa untuk penulis dalam menempuh studi pascasarjana.
6. Kepada rekan-rekan Pascasarjana Teknik Elektro yang telah banyak membantu penulis dalam menyelesaikan tugas selama masa kuliah dan tesis ini.

Semoga Tuhan yang pengasih lagi penyayang membalas budi baik mereka semua. Pada akhirnya, penulis menyadari bahwa tesis ini masih belum sempurna.

Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga tesis ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi serta bagi masyarakat demi kemajuan Bangsa.

Surabaya, 9 Juni 2020

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN TESIS .....	iii
PERNYATAAN KEASLIAN TESIS .....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xvii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan.....	3
1.4 Batasan Masalah .....	3
1.5 Kontribusi .....	4
BAB 2 KAJIAN PUSTAKA.....	5
2.1 Kajian Penelitian Terkait.....	5
2.2 Teori Dasar .....	10
2.2.1 Tingkat Asam atau Basa (pH) .....	10
2.2.2 Pewarna Fluoresen Pyranine .....	10
2.2.3 Photoresistor .....	12
2.2.4 Photodiode .....	14
2.2.5 Sensor TSL2561 .....	16
2.2.6 Sensor AS7262 .....	16
2.2.7 Regresi Linier Sederhana .....	17
BAB 3 METODE PENELITIAN.....	19
3.1 Studi Literatur .....	20
3.2 Reagen dan Material .....	21

3.3	Hardware dan Software .....	21
3.4	Mempersiapkan Pewarna Fluoresen .....	22
3.5	Pengembangan Alat Pengukur pH .....	23
3.5.1	Perancangan Instrumen Optik .....	23
3.5.2	Perancangan Pengkondisi Sinyal.....	24
3.5.3	Perancangan Aplikasi .....	26
3.6	Pengujian Alat Pengukur pH .....	27
3.6.1	Mengamati Respon Sensitivitas Fotodetektor.....	28
3.6.2	Mengamati Efek Photobleaching Pada Pewarna Fluoresen.....	28
3.6.3	Mengamati Efek Volume Pewarna Fluoresen.....	29
3.6.4	Kalibrasi Pada Alat.....	29
BAB 4 EKSPERIMEN DAN ANALISA DATA.....		31
4.1	Pengujian Respon Sensitivitas Fotodetektor .....	31
4.2	Pengujian Alat Pengukur pH .....	32
4.2.1	Mengamati Efek Photobleaching .....	33
4.2.2	Mengamati Perubahan Volume Fluoresen.....	33
4.2.3	Kalibrasi.....	35
4.2.4	Menguji Nilai Kesalahan Baca Pada Alat.....	37
4.2.5	Mengukur pH Pada Sampel Acak .....	39
BAB 5 KESIMPULAN.....		41
5.1	Kesimpulan .....	41
5.2	Saran.....	41
DAFTAR PUSTAKA .....		43
LAMPIRAN .....		47
DAFTAR INDEX.....		99



## DAFTAR GAMBAR

Gambar 1.1 Prinsip dasar pengembangan alat pengukur pH .....	2
Gambar 2.1 Deteksi pH ratiometrik oleh sensor Pyranine-APCN yang dipisahkan nanofase.....	5
Gambar 2.2 Spektrum emisi fluoresensi Pyranine-APCN dalam larutan buffer dari eksitasi dan pH yang berbeda pada 460 nm (a) dan 406 nm (b) .....	6
Gambar 2.3 Kurva kalibrasi sensor diukur pada kekuatan ion yang berbeda .....	7
Gambar 2.4 Kurva kalibrasi sensor diukur pada kekuatan ion yang berbeda.....	7
Gambar 2.5 Spektrum emisi fluorescein (450 hingga 650 nm) untuk berbagai larutan pH.....	8
Gambar 2.6 Intensitas fluoresensi relatif pada nilai pH tertentu sebelum diberi organofosfat.....	9
Gambar 2.7 Intensitas fluoresen dengan pestisida OP: (a) titrasi dengan etion; (B) titrasi dengan malathion; (c) titrasi dengan parathion; dan (d) titrasi dengan fenthion. Dalam setiap kasus arah panah menunjukkan konsentrasi 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24 $\mu\text{M}$ .....	9
Gambar 2.8 Struktur kimia dari pyranine .....	11
Gambar 2.9 Panjang gelombang eksitasi dan emisi dari pyranine .....	12
Gambar 2.10 Diagram skematis sel fotoresistif .....	13
Gambar 2.11 Sensitivitas photoresistor.....	13
Gambar 2.12 Pelapisan photoresistor berbentuk serpentine.....	14
Gambar 2.13 Konstruksi photodiode .....	14
Gambar 2.14 Respon sensitivitas photodiode silikon.....	15
Gambar 2.15 Respon sensitivitas sensor TSL2561 .....	16
Gambar 2.16 Sirkuit diagram sensor TSL2561 .....	16
Gambar 2.17 Sirkuit diagram sensor AS7262.....	17
Gambar 3.1 Tahapan penelitian.....	19
Gambar 3.2 Diagram blok studi literatur .....	20
Gambar 3.3 Rancangan instrumen optik.....	24
Gambar 3.4 Alat instrumen optik .....	24

Gambar 3.5 Diagram blok pengkondisi sinyal.....	24
Gambar 3.6 Rangkaian pengkondisi sinyal.....	25
Gambar 3.7 Diagram blok komunikasi antara <i>transmitter</i> dengan <i>receiver</i> .....	26
Gambar 3.8 Antarmuka aplikasi monitoring intensitas fluoresen.....	26
Gambar 3.9 Flowchart aplikasi.....	27
Gambar 4.1 Intensitas fluoresen pada pH air suling.....	32
Gambar 4.2 Efek photobleaching pada pewarna fluoresen.....	33
Gambar 4.3 Intensitas fluoresen dari beberapa volume pewarna fluoresen yang berbeda dengan campuran 2 mL air suling.....	35
Gambar 4.4 Intesitas fluoresen dari keempat pengujian yang telah dilakukan.....	36
Gambar 4.5 Kurva sensitivitas fungsi regresi linier yang diterapkan pada alat yang telah dikembangkan.....	37
Gambar 4.6 Hasil perbandingan pengukuran pH dengan alat yang telah dikembangkan dan pH meter digital pada sampel.....	39

## DAFTAR TABEL

Tabel 2.1 Jenis bahan yang digunakan pada photodiode .....	15
Tabel 4.1 Normalisasi Intesitas Fluoresen Pada pH Air Suling .....	31
Tabel 4.2 Hasil Pengukuran Intensitas Fluoresen Pada pH yang Diamati Menggunakan pH Meter Digital Dengan Volume Air Suling 2 mL .....	34
Tabel 4.3 Hasil Perbandingan Pengukuran Alat Yang Telah Dikembangkan Dengan pH Meter Digital Pada Kisaran pH 4,0 Hingga 10,0 Dengan Interval 0.5 .....	37
Tabel 4.4 Hasil Perbandingan Pengukuran pH Dengan Alat Yang Telah Dikembangkan dan pH Meter Digital Pada Sampel.....	38

*Halaman ini sengaja dikosongkan*

# **BAB 1**

## **PENDAHULUAN**

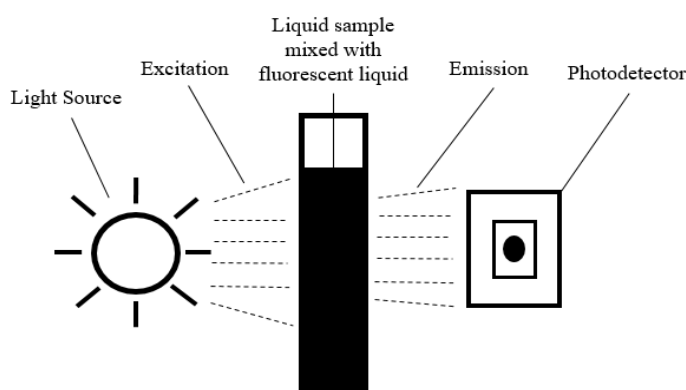
### **1.1 Latar Belakang**

Air adalah kebutuhan yang sangat penting dan mendasar bagi kehidupan. Air adalah zat molekul yang memiliki sifat kimia dan fisik yang unik yang terkait dengan fungsi dalam tubuh manusia [1]. Tubuh manusia cenderung mempertahankan kisaran pH yang dikontrol oleh tubuh dari sekitar 7,35 hingga 7,45 dalam cairan ekstraseluler melalui ekskresi karbon dioksida pernapasan dan ekskresi ginjal dari asam atau basa atau basa asam non-karbonat asam atau basa [2]. Karena air adalah persyaratan paling penting bagi manusia, pengukuran pH diperlukan untuk menguji kualitas air apakah air yang digunakan terkontaminasi yang dapat mempengaruhi perubahan pH dalam air. PH adalah ukuran konsentrasi ion hidrogen atau aktivitas larutan (cairan) yang dapat diukur untuk menentukan nilai pH-nya. Nilai pH berkisar dari pH 0-14 dengan nilai di bawah pH 7 menunjukkan sifat asam dan nilai di atas pH 7 menunjukkan sifat dasar atau alkali. Nilai pada pH 7 adalah pusat skala pengukuran yang tidak menunjukkan sifat asam atau basa [3].

Memiliki pengukuran kadar asam atau basa (pH level) yang cepat sangat penting dalam banyak kasus. Salah satu metode untuk mengukur tingkat pH adalah dengan menggunakan optikal fluoresen. Pewarna fluoresen adalah senyawa kimia yang menentukan perubahan dalam penyerapan cahaya antara reaktan dan proses reaksi atau mengukur cahaya dengan luminesensi [4]. Pewarna fluoresen adalah molekul non-protein yang menyerap cahaya dan memancarkan cahaya pada panjang gelombang yang lebih panjang. Mereka sering digunakan dalam pelabelan biomolekul fluoresen dan bisa lebih kecil dari protein fluoresen tetapi tidak dapat dikodekan secara genetik [5]. Varian pewarna fluoresen yang berbeda memiliki sensitivitas pH yang berbeda [6].

Dalam penelitian ini, pendekatan metode optikal fluoresen digunakan untuk mengukur pH air yang berhubungan langsung dengan manusia atau lingkungan manusia. Metode ini sekarang banyak digunakan karena sensitivitasnya

yang sangat baik, yang memberikan batas deteksi pada tingkat picomolar dan di bawahnya, dan beragamnya metode presentasi sampel yang tersedia [7]. Pewarna fluoresen diekstraksi dan ditambahkan ke sampel air. Jenis pewarna fluoresen yang digunakan dalam penelitian ini adalah pewarna fluoresen berjenis pyranine. Pengembangan alat pengukur pH ini terdiri dari sumber cahaya yang dieksitasikan menuju sampel cairan yang dicampur dengan cairan fluoresen dan pada saat yang sama intensitas emisi fluoresen ditangkap menggunakan fotodetektor. Intensitas emisi fluoresen adalah fungsi dari pH sampel air. Gambar 1 menunjukkan prinsip dasar pengembangan perangkat penginderaan pH.



Gambar 1.1 Prinsip dasar pengembangan alat pengukur pH.

Metode optikal fluoresen memiliki kelebihan seperti memungkinkan perolehan spektrum yang sederhana, cepat, dan tepat. Keuntungan lain dari metode ini adalah pengukuran tidak merusak sampel, dilakukan langsung pada sampel, menghindari langkah-langkah pengolahan bahan kimia yang memakan waktu dan tenaga kerja, menghilangkan konsumsi reagen dan produksi limbah, sehingga memenuhi persyaratan untuk kimia analitik hijau dan ketersediaan sensor fluoresensi untuk aplikasi tertentu yang memungkinkan pengukuran secara real-time [8]. Oleh karena itu, dengan menerapkan metode ini pada alat pengukur pH, hasil pengukuran pH sampel air yang berkaitan erat dengan manusia seperti mengukur pH air yang dikonsumsi dapat dilakukan dengan cepat dan akurat. Pengukuran cepat diperlukan untuk menghemat waktu. Selain itu, pengukuran hanya membutuhkan volume sampel cairan yang sangat sedikit.

## **1.2 Rumusan Masalah**

Berdasarkan dari latar belakang yang telah disampaikan, maka penulis membuat beberapa perumusan masalah dari penelitian ini, yaitu:

1. Menentukan jenis pewarna fluoresen.
2. Merancang instrumen optik berdasarkan spesifikasi dari pewarna fluoresen yang digunakan.
3. Merancang pengkondisi sinyal dari transduser yang digunakan.
4. Merancang dan mendesain antarmuka untuk memonitoring dari perubahan intensitas cahaya yang dipancarkan oleh fluoresen.
5. Melakukan kalibrasi berdasarkan pengamatan intensitas dari fluoresen.
6. Melakukan pengujian dan menganalisa kemampuan alat pengukur pH.

## **1.3 Tujuan**

Berdasarkan rumusan masalah yang telah disebutkan, tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang dan menentukan bahan pewarna fluoresen.
2. Menentukan spesifikasi dan merancang instrumen optik.
3. Mengimplementasikan alat pengukur pH.
4. Mengamati dan menganalisa kemampuan alat pengukur pH dalam melakukan pengukuran dari aspek kecepatan dan akurasi pengukuran.

## **1.4 Batasan Masalah**

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Jenis pewarna fluoresen yang digunakan dalam pengembangan alat pengukur pH ini berjenis pyranine.
2. Pengukuran pH diamati berdasarkan perubahan intensitas cahaya yang diemisikan oleh sampel cairan yang dicampur dengan pewarna fluoresen.
3. Eksitasi yang diberikan berada pada panjang gelombang 405 nm.

## **1.5 Kontribusi**

Kontribusi yang dihasilkan oleh penelitian ini adalah menguji kemampuan alat untuk mengukur pH pada sampel cairan berbasis fluoresen optik dengan cepat, akurat, biaya yang terjangkau dan dapat dimonitoring secara nirkabel. Harapan kedepannya, dari penelitian yang telah dilakukan dapat kembangkan untuk mengukur sampel yang lebih spesifik.

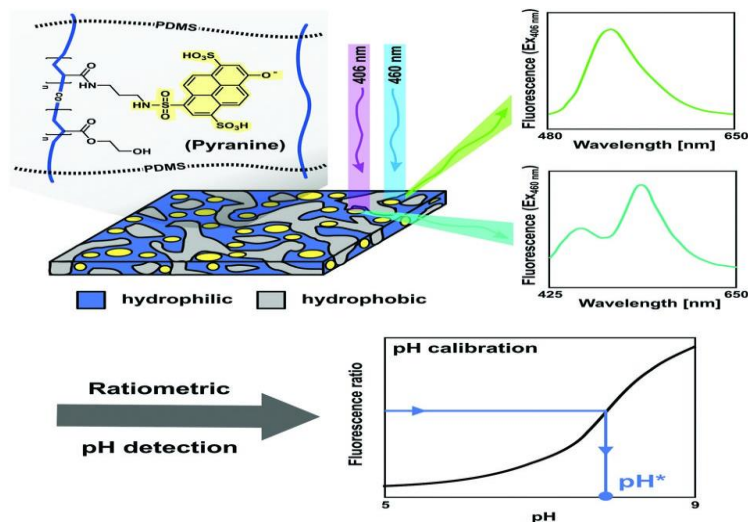


## BAB 2

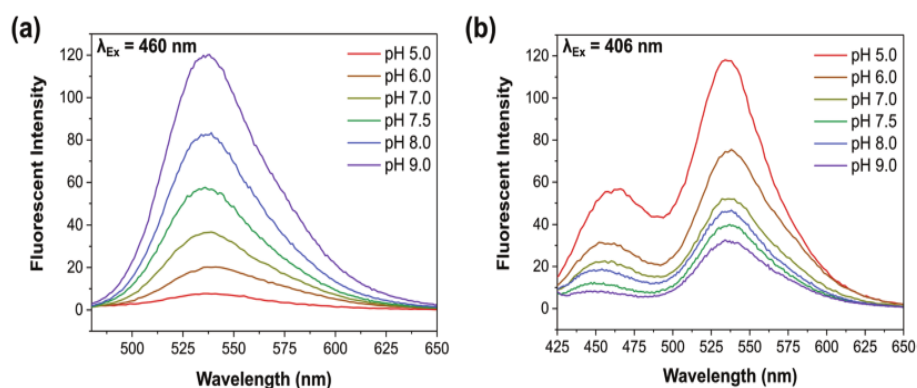
### KAJIAN PUSTAKA

#### 2.1 Kajian Penelitian Terkait

Sebastian, Alina dkk., Menggunakan metode *amphiphilic polymer-linked conjugation* (APCN) yang dimodifikasi oleh nanofase yang dipisahkan oleh pyranine dalam membuat sensor pH ratiometrik fluoresen (Gambar 2.1). Membran APCN yang tipis dan berdiri bebas yang terdiri dari satu polimer hidrofilik dan satu hidrofobik memberikan matriks ideal yang transparan, fleksibel, dan stabil secara optik yang memungkinkan kontak antara pewarna dan lingkungan berair. Pendekatan konjugasi berbasis ester aktif menghasilkan modifikasi pyranine yang sangat homogen dan stabil dari fase hidrofilik APCN. Nanophase dipisahkan ikatan polimer amphiphilic menyediakan sistem matriks yang ideal untuk pyranine. Pendekatan fungsionalisasi berbasis ester aktif memastikan konjugasi stabil ke fase hidrofilik dan kontak dengan lingkungan berair, memungkinkan penggunaannya sebagai sensor pH ratiometrik fluoresen dalam kisaran pH 5-9 dengan eksitasi pada panjang gelombang 460 nm (Gambar 2.2 (a)) dan 406 nm (Gambar 2.2 (b)) [9].

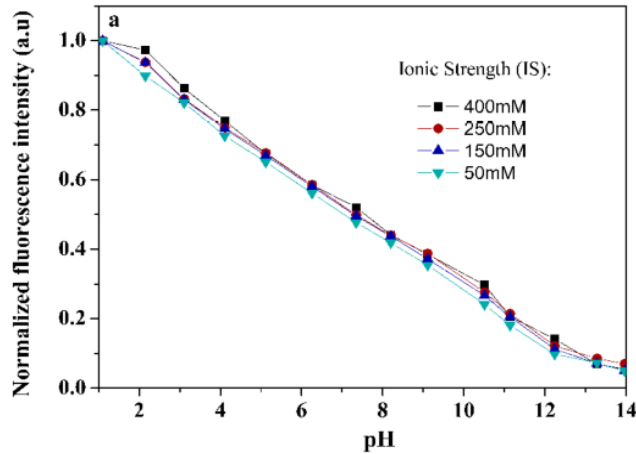


Gambar 2.1 Deteksi pH ratiometrik oleh sensor Pyranine-APCN yang dipisahkan nanofase.



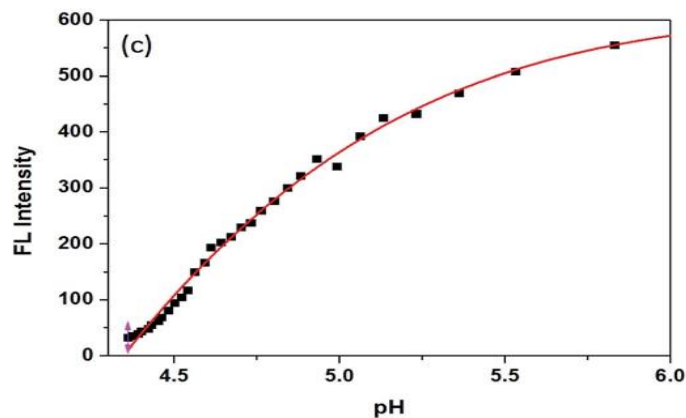
Gambar 2.2 Spektrum emisi fluoresensi Pyranine-APCN dalam larutan buffer dari eksitasi dan pH yang berbeda pada 460 nm (a) dan 406 nm (b).

Serangkaian sensor optik baru untuk mendeteksi pH dengan rentang yang luas berdasarkan fluorofor tunggal dan multi-ionofor dengan nilai pK (a) yang berbeda yang dikembangkan oleh Qi J, Liu D, Liu X, dkk [10]. PK (a) adalah ukuran kemampuan suatu senyawa untuk melepaskan proton ke dalam pelarut, dalam kondisi kesetimbangan atau konstanta kesetimbangan spesifik untuk suatu asam dan basa konjugasinya dalam larutan berair. Sensor optik yang mereka buat menggunakan transfer elektron yang diinduksi foto (PET) sebagai transduksi sinyal dan mengikuti konsep desain "fluorophore-spacer-receptor (ionophore)" yang menggunakan 4-amino-1,8-naphthalimide sebagai fluorofor tunggal, etil sebagian sebagai pengatur jarak, dan serangkaian fenol dan anilin sebagai reseptor. Mereka menggabungkan reseptor dengan enam nilai pKa yang berbeda dengan fluorofor tunggal yang menghasilkan sifat optik yang sensitiv. Desain rasional ini menghasilkan serangkaian sensor pH optik dengan sifat fluoresensi unik dan pengukuran pH akurat mulai dari 1 hingga 14 unit pH. Karena imobilisasi kovalen indikator, sensor ini menunjukkan stabilitas yang sangat baik, reversibilitas yang memadai, dan rentang dinamis yang memuaskan hingga kisaran pH penuh (pH 1-14) (Gambar 2.3) [10].



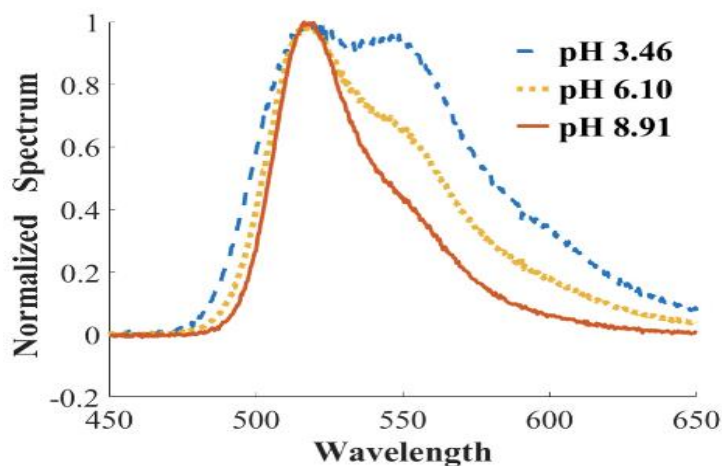
Gambar 2.3 Kurva kalibrasi sensor diukur pada kekuatan ion yang berbeda.

Dalam penelitian mereka, Yuan, Zhang, Xi dan Tao merancang probe baru fluorescent pH,8- (6H, 12H-5,11-methanodibenzo [b, f] diazocineylene) -di (p-ethenyl-pyridine) (TBPP) menggabungkan gugus amina yang mendonasikan elektron dan kelompok piridin yang menerima elektron melalui penghubung dasar Tröger dirancang dan disintesis. Akibatnya, TBPP menunjukkan efek transfer muatan intramolekul yang disebabkan oleh interaksi donor - akseptor antara unit amina dan piridinnya. Emisinya dapat secara terbalik beralih antara kondisi biru dan gelap dengan protonasi dan deprotonasi (Gambar 2.4). Perilaku seperti itu memungkinkannya berfungsi sebagai sensor pH fluoresen dalam larutan. Analisis spektroskopi  $^1\text{H}$  NMR menunjukkan bahwa perubahan afinitas elektron unit piridinil pada protonasi dan deprotonasi bertanggung jawab untuk proses pendeteksian tersebut [11].



Gambar 2.4 Kurva kalibrasi sensor diukur pada kekuatan ion yang berbeda.

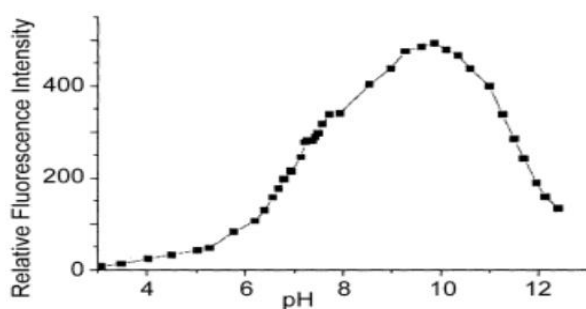
Sharma, Graham, dkk., Telah mengembangkan perangkat penginderaan pH fluoresensi ratiometrik menggunakan pewarna dan LED yang disetujui FDA (Food and Drug Administration) yang bertujuan untuk mengevaluasi kemampuan produksi asam dari endapan plak dalam lubang dan celah di daerah oklusal dan interproksimal pada permukaan enamel gigi. Profil spektral Fluorescein (FITC) dikumpulkan menggunakan spektrometer dan dianalisis dengan algoritme unmixing spektral untuk kalibrasi pada kisaran pH 4,5 hingga 7 (Gambar 2.5). Penelitian pilot in vivo pada subjek manusia dilakukan menggunakan pembilasan sukrosa untuk mempercepat metabolisme bakteri dan untuk mengukur waktu-tergantung penurunan pH. Sistem optik relatif kebal terhadap faktor pembaur seperti photobleaching, konsentrasi pewarna, dan variasi dalam intensitas eksitasi yang terkait dengan teknik pengukuran pH berbasis pewarna fluoresen [12].



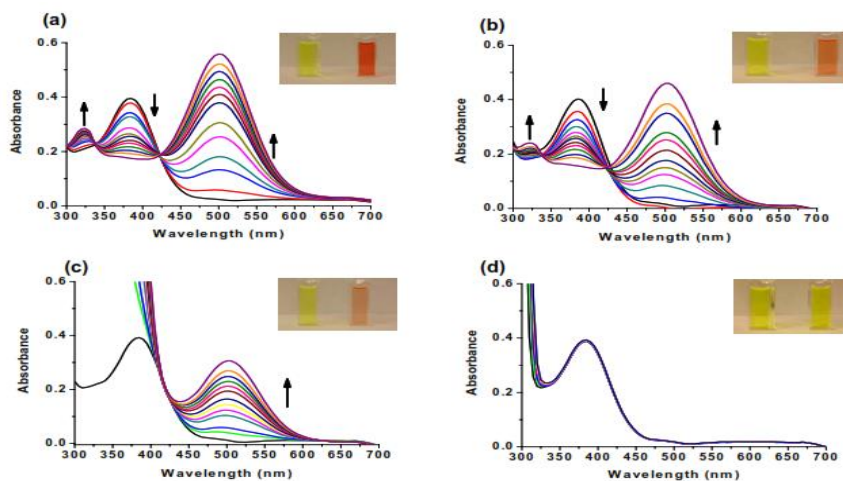
Gambar 2.5 Spektrum emisi fluorescein (450 hingga 650 nm) untuk berbagai larutan pH.

Obare, Chandrima dkk, menggunakan metode kemosensor berbasis fluoresen untuk mendeteksi senyawa organofosfat. Mereka menggunakan pewarna fluoresen yang peka terhadap pH, yang terdiri dari enzim AChE yang terkait dengan senyawa *fluorescein isothiocyanate* (FITC) yang sensitif terhadap pH. Adsorpsi enzim-pewarna diimobilisasikan pada serat kuarsa yang melekat pada *spektrofotometer fluoresensi*. Dengan tidak adanya senyawa organofosfat, enzim AChE mampu menghidrolisis *asetilkolin* yang mengalami penurunan pH sehingga

mengakibatkan pengurangan intensitas fluoresensi FITC karena gangguan konjugasi *fluorophore* pada *protonasi* (Gambar 2.6). Kemudian pengujian dilakukan dengan memberikan senyawa organofosfat dan perubahan warna fluoresen dari kemosensor ini diamati dengan menggunakan lampu UV dibawah temperatur *ambien*. Sensor ini sangat sensitif dan selektif terhadap senyawa organofosfat. Hasilnya, biosensor ini sangat sensitif dimana mampu mendeteksi konsentrasi *paraoxon* dan menunjukkan beberapa selektivitas terhadap senyawa OP yang berbeda (Gambar 2.7) [13].



Gambar 2.6 Intensitas fluoresensi relatif pada nilai pH tertentu sebelum diberi organofosfat.



Gambar 2.7 Intensitas fluoresen dengan pestisida OP: (a) titrasi dengan etion; (B) titrasi dengan malathion; (c) titrasi dengan parathion; dan (d) titrasi dengan fenthion. Dalam setiap kasus arah panah menunjukkan konsentrasi 0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24  $\mu\text{M}$ .

## 2.2 Teori Dasar

### 2.2.1 Tingkat asam atau Basa (pH)

Tingkat asam atau basa (pH) adalah indeks konsentrasi ion hidrogen ( $[H^+]$ ) dalam air. Ion hidrogen mempengaruhi sebagian besar proses kimia dan biologis; dengan demikian, pH merupakan variabel penting dalam menentukan kualitas air. Konsep pH didasarkan pada ionisasi air [14]:



Air murni bersifat netral (tidak bersifat asam atau basa karena  $[H^+]$  sama dengan  $[OH^-]$ ). Ini adalah praktik umum untuk menggunakan skala pH 0-14 dengan nilai pH 7 sebagai titik tengah. Nilai pH di bawah 7 menunjukkan reaksi yang semakin asam, dan nilai pH di atas 7 menunjukkan reaksi yang semakin basa atau alkali. Namun, pH air murni hanya 7,00 pada suhu 25°C. Pada suhu yang lebih rendah, pH air murni di atas 7, pada suhu yang lebih tinggi, pH di bawah 7 [14].

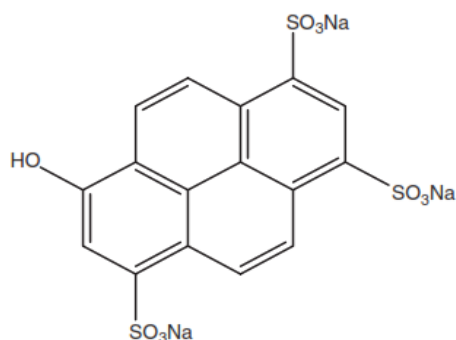
Seiring perubahan pH sampel cairan, dapat terjadi banyak presipitasi, kopresipitasi, dan proses penyerapan yang mengubah komposisi kimia dan laju reaksi sampel. Proses biologis sampel juga dipengaruhi oleh pH-nya. Perubahan kandungan gas terlarut dari sampel dapat mengubah pH. Air umumnya dalam kesetimbangan dengan karbon dioksida ( $CO_2$ ) pada tekanan parsial beberapa kali lipat dari atmosfer. Saat terpapar ke atmosfer,  $CO_2$  ini lolos dan pH naik. Karena itu penting bahwa pH diukur di tempat [15].

### 2.2.2 Pewarna Fluoresen Pyranine

Pewarna fluoresen, atau fluorofor, adalah senyawa yang menyerap cahaya pada panjang gelombang tertentu dan memancarkan cahaya pada panjang gelombang yang lebih tinggi dan menghasilkan fluoresensi dalam warna tertentu. Pewarna ini dapat dikelompokkan ke dalam kategori seperti pewarna organik (Fluorescein, rhodamin, AMCA), fluorofor biologis (Protein fluoresen hijau, phycoerythrin, allophycocyanin) dan titik-titik kuantum. Penggunaan fluorofor dalam pelabelan dan pewarnaan biologis dimulai sejak tahun 1930-an. Jumlah pilihan pewarna fluoresen memungkinkan tidak hanya fleksibilitas, tetapi juga lebih banyak peluang dalam apa yang dapat dideteksi dan diukur dalam aplikasi

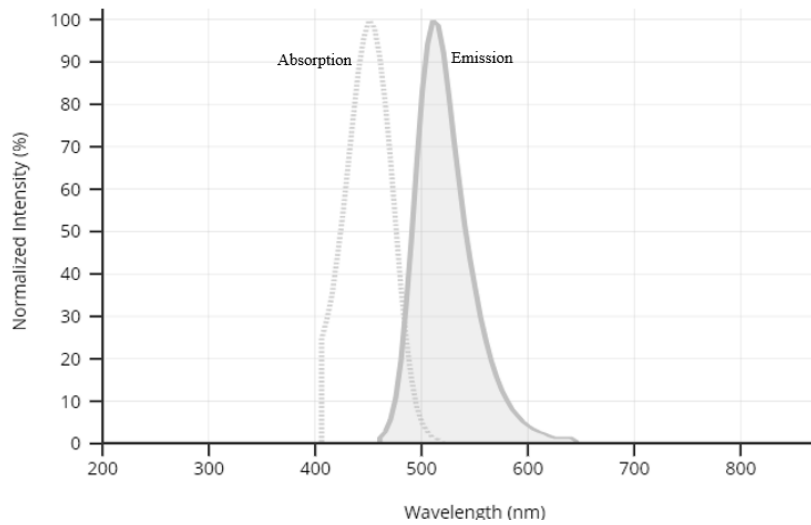
penelitian seperti mikroskop fluoresensi, flow cytometry, dan penyaringan throughput yang tinggi. Beberapa hal yang perlu dipertimbangkan dalam memilih pewarna fluoresens meliputi: eksitasi fluorofor dan spektrum emisi, intensitas relatifnya, kemampuan instrumen pencitraan (seperti laser eksitasi dan filter), dan target yang diukur [16].

Pyranine adalah garam natrium organik. Pyranine memiliki peran sebagai fluorokrom yang mengandung pyranine (3-) yang berasal dari pyrene hydride. Pyranine atau trisodium 8-hydroxypyrene-1,3,6-trisulfonate memiliki berat molekul 524,4 g / mol [17]. Rumus molekul dari pyranine adalah  $C_{16}H_7Na_3O_{10}S_3$ . Pewarna fluoresen ini digunakan dalam berbagai bidang seperti bidang biomedis atau kesehatan dan bidang industri. Pada bidang biomedis atau kesehatan pyranine dapat digunakan untuk mengukur tingkat asam atau basa pada cairan. Pada bidang industri pyranine dapat digunakan sebagai sensor oksigen dan karbon dioksida. Gambar 2.8 menunjukkan struktur kimia dari pyranine [18].



Gambar 2.8 Struktur kimia dari pyranine.

Panjang gelombang puncak absorpsi dari pyranine adalah 405 nm dan puncak panjang gelombang emisi adalah 510 nm [19]. Pyranine menyerap energi cahaya dari panjang gelombang 400 nm hingga 480 nm dan memancarkan cahaya pada panjang gelombang 487 nm hingga 550 nm. Panjang gelombang yang diserap, efisiensi transfer energi, dan waktu sebelum emisi tergantung pada struktur pyranine dan lingkungan kimianya, ketika molekul dalam keadaan tereksitasi berinteraksi dengan molekul di sekitarnya. Gambar. 2.9 menunjukkan panjang gelombang eksitasi dan emisi dari pyranine [20].

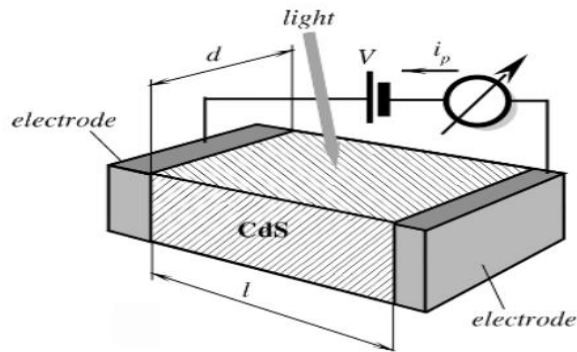


Gambar 2.9 Panjang gelombang eksitasi dan emisi dari pyranine.

### 2.2.3 Photoresistor

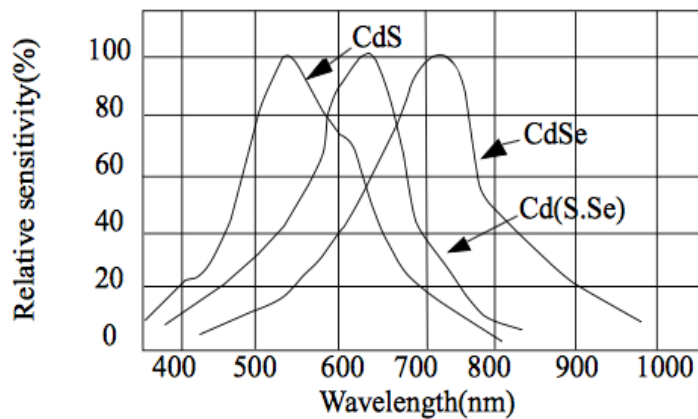
Photoresistor merupakan perangkat fotokonduktif. Bahan yang paling umum digunakan dalam pembuatannya adalah cadmium sulfide (CdS) dan cadmium selenide (CdSe), yang merupakan semikonduktor yang resistansinya berubah ketika cahaya memasuki permukaan. Untuk operasinya, sebuah photoresistor memerlukan sumber daya karena photoresistor tidak menghasilkan arus listrik yang dimana efek cahaya dimanifestasikan dalam perubahan hambatan listrik material. Gambar 2.10 menunjukkan diagram skematis sel fotoresistif. Sebuah elektroda diatur di setiap ujung fotokonduktor. Dalam kegelapan, daya tahan materialnya tinggi. Oleh karena itu, tegangan (V) yang diterapkan menghasilkan arus gelap kecil yang dikaitkan dengan efek suhu. Ketika cahaya muncul di permukaan, arus ( $i_p$ ) mengalir. Alasan arus dapat mengalir dimana tepat di bawah pita konduksi kristal adalah tingkat donor dan ada tingkat akseptor di atas pita valensi. Dalam kegelapan, elektron dan lubang di setiap tingkat hampir dijejalkan di tempatnya di dalam kristal, menghasilkan resistensi semikonduktor yang tinggi [21].





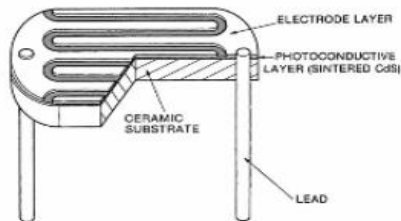
Gambar 2.10 Diagram skematis sel fotoresistif.

Ketika cahaya menerangi kristal fotokonduktif, foton diserap, yang menghasilkan energi tambahan dalam elektron pita valensi. Ini memindahkan mereka ke pita konduksi, menciptakan lubang bebas di pita valensi, meningkatkan konduktivitas material. Karena di dekat pita valensi terdapat tingkat akseptor terpisah yang dapat menangkap elektron bebas tidak semudah lubang bebas, probabilitas rekombinasi elektron dan lubang berkurang dan jumlah elektron bebas dalam pita konduksi tinggi. Karena CdS memiliki celah pita 2,41 eV, panjang gelombang serapan tepi adalah  $\lambda = c / \nu \approx 515 \text{ nm}$ , yang berada dalam rentang spektral yang terlihat. Karenanya, CdS mendeteksi cahaya yang lebih pendek dari panjang gelombang 515-nm. Fotokonduktor lain memiliki panjang gelombang serapan tepi yang berbeda. Misalnya, CdS paling sensitif pada rentang panjang gelombang yang lebih pendek, sedangkan Si dan Ge paling efisien diantara spektrum inframerah. Gambar 2.11 menunjukkan sensitivitas dari photoresistor [21].



Gambar 2.11 Sensitivitas photoresistor.

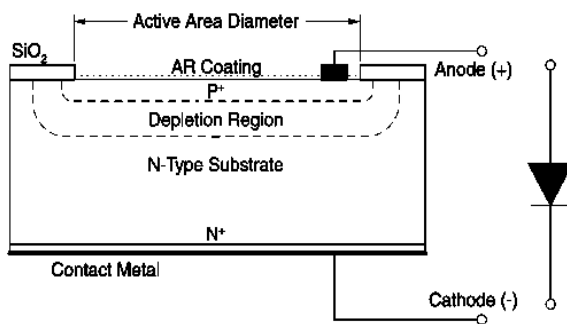
Untuk mendapatkan sensitivitas yang lebih baik dan resistensi sel yang lebih rendah, jarak antara elektroda harus dikurangi, dan lebar dari sensor harus ditingkatkan. Gambar 2.12 menunjukkan pelapisan photoresistor berbentuk serpentine [21].



Gambar 2.12 Pelapisan photoresistor berbentuk serpentine.

## 2.2.4 Photodioda

Photodioda merupakan perangkat semikonduktor yang responsif terhadap partikel energi tinggi dan *foton* yang berfungsi untuk mendeteksi ada atau tidaknya kuantitas cahaya. Bahan yang digunakan untuk membuat photodioda sangat penting untuk mendefinisikan sifat-sifatnya (Tabel 2.1) [21]. Gambar 2.13 menunjukkan konstruksi dari photodioda [21].



Gambar 2.13 Konstruksi photodioda.

Jika P-N *junction* pada pengoperasian *forward biased* (sisi positif dari baterai tehubung ke sisi-P) dan terkena cahaya pada frekuensi yang tepat, maka peningkatan arus akan sangat kecil sehubungan dengan arus gelap (*dark current*). Dengan begitu, arus bias jauh lebih besar dibandingkan arus yang dihasilkan oleh cahaya. Jika P-N *junction* pada pengoperasian *reverse biased* (sisi positif dari baterai tehubung ke sisi-N), ketika cahaya mengenai photodioda, maka akan terjadi

peningkatan arus. Ini disebabkan *foton* yang merupakan partikel terkecil cahaya akan menembus lapisan sisi-N dan memasuki lapisan sisi-P dari photodiode. *Foton-foton* ini akan bertabrakan dengan elektron-elektron yang terikat sehingga elektron tersebut terpisah dari intinya dan menyebabkan terjadinya *hole*. Ini menyebabkan elektron bertambah di sisi-N sedangkan sisi-P akan kelebihan *hole*. Pemisahan muatan positif dan negatif ini menyebabkan beda potensial pada P-N *junction* [21].

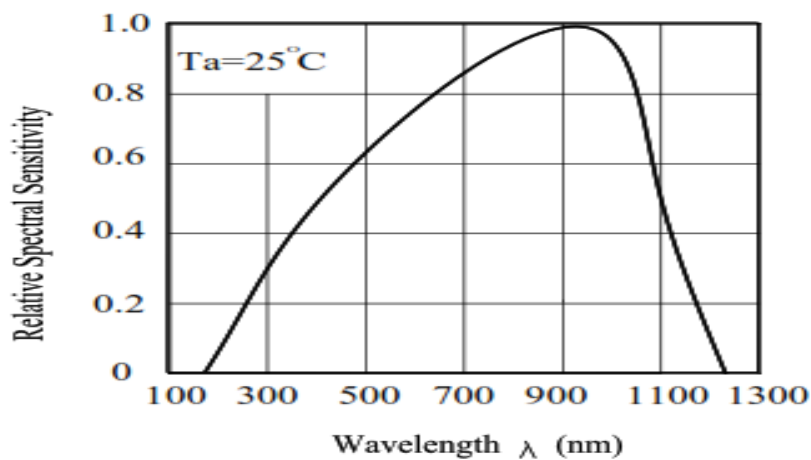
Tabel 2.1 Jenis bahan yang digunakan pada photodiode.

Bahan	Spektrum elektromagnetik Rentang panjang gelombang (nm)
Silicon	190-1100
Germanium	400-1700
Indium gallium arsenide	800-2600
Timah (II) sulfida	<1000-3500
Mercury cadmium telluride	400-14000

Responsivitas photodiode ( $R_\lambda$ ) merupakan ukuran sensitivitas terhadap cahaya, ini didefinisikan sebagai perbandingan dari arus foto ( $I_p$ ) terhadap insiden kekuatan cahaya ( $P$ ) yang terjadi pada panjang gelombang tertentu [22]:

$$R_\lambda = \frac{I_p}{P} \quad (2.2)$$

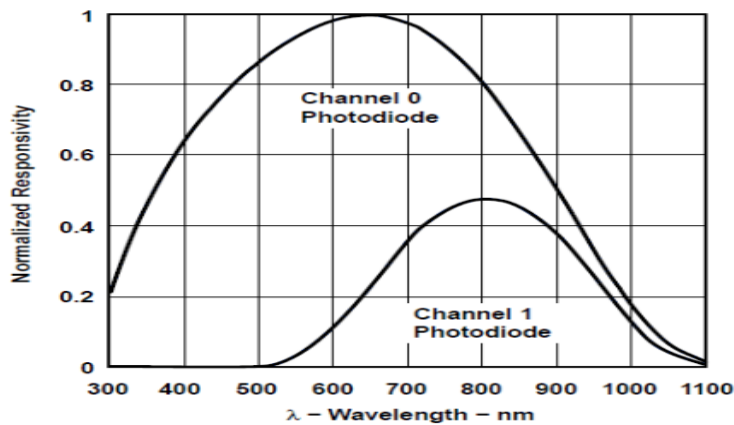
Dari persamaan diatas, ini merupakan ukuran efektivitas konversi daya cahaya menjadi arus listrik. Gambar 2.14 menunjukkan responsivitas dari photodiode silikon [22].



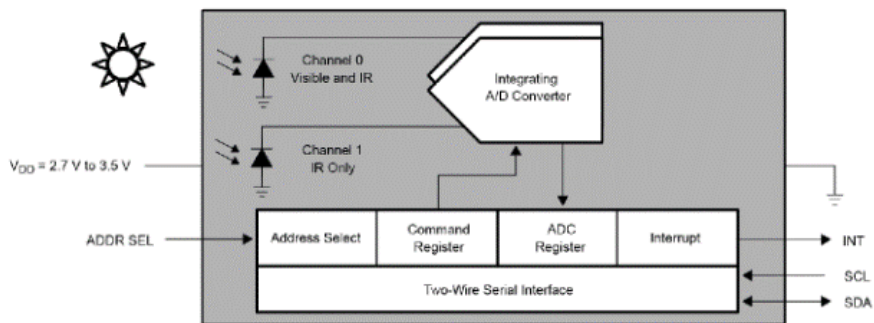
Gambar 2.14 Responsivitas photodiode silikon.

### 2.2.5 Sensor TSL2561

TSL2561 adalah sensor cahaya yang mengubah intensitas cahaya menjadi output sinyal digital yang mampu menghasilkan antarmuka I2C. Didalam modul sensor ini terdapat satu fotodiode yang merespon cahaya tampak dan satu fotodiode yang merespons inframerah pada satu sirkuit yang sudah terintegrasi dalam bentuk CMOS dengan resolusi 16-bit. Sensor ini beroperasi pada tegangan 2.7 – 3.6 Volt dan pada suhu -30 °C sampai 70 °C. Gambar 2.15 menunjukkan respon sensitivitas sensor TSL2561 dan Gambar 2.16 merupakan sirkuit diagram sensor TSL2561 [23].



Gambar 2.15 Respon sensitivitas sensor TSL2561.

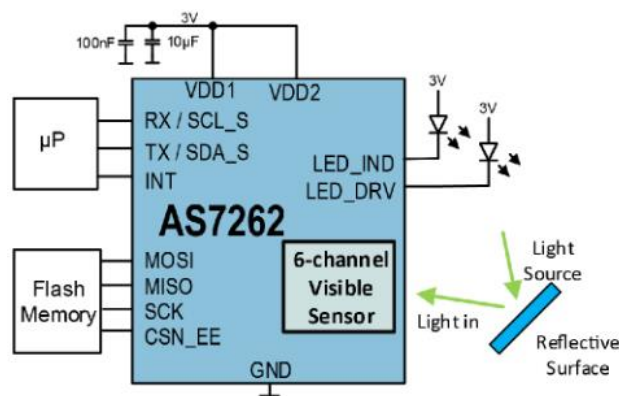


Gambar 2.16 Sirkuit diagram TSL2561.

### 2.2.6 Sensor AS7262

Sensor AS7262 merupakan sensor multispektral yang hemat biaya. Sensor ini memiliki enam kanal dalam panjang gelombang yang terlihat dari sekitar 430 nm hingga 670 nm dengan *Full-Width Half-Max* (FWHM) maksimal 40nm. Sensor

AS7262 mengintegrasikan filter Gaussian ke dalam silikon CMOS standar melalui teknologi filter interferensi deposit nano-optik dan dikemas dalam paket LGA yang menyediakan bukaan internal untuk mengontrol cahaya yang memasuki array sensor. Akses kontrol dan data spektral diimplementasikan melalui set register I<sup>2</sup>C, atau dengan perintah spektral AT tingkat tinggi yang diatur melalui serial UART. Sensor beroperasi pada tegangan 2.7 Volt sampai 3.6 Volt dan bekerja pada temperatur -40 °C sampai 85 °C. Gambar 2.17 menunjukkan sirkuit diagram sensor AS7262 [24].



Gambar 2.17 Sirkuit diagram sensor AS7262.

### 2.2.7 Regresi Linier Sederhana

Regresi Linier Sederhana adalah metode statistik yang berfungsi untuk menguji sejauh mana hubungan sebab akibat antara variabel sebab akibat (X) terhadap variabel pengaruh. Faktor penyebab umumnya dilambangkan dengan X atau disebut juga prediktor sedangkan variabel efek dilambangkan dengan Y atau disebut juga respons. Regresi Linier Sederhana juga merupakan salah satu metode statistik yang digunakan dalam produksi untuk membuat prediksi atau prediksi tentang karakteristik kualitas dan kuantitas [25]. Model Persamaan Regresi Linier Sederhana adalah sebagai berikut:

$$Y = a + bX \quad (2.3)$$

Dimana, Y adalah variabel respon atau variabel dependent (Dependent), X adalah variabel prediktor atau variabel penyebab (Independen), a adalah konstanta,

n adalah jumlah total data dan b adalah koefisien regresi (besarnya respons yang dihasilkan oleh prediktor).

Nilai a dan b dapat dihitung menggunakan rumus di bawah ini:

$$a = \frac{(\Sigma Y)(\Sigma X^2) - (\Sigma X)(\Sigma XY)}{n(\Sigma X^2) - (\Sigma X)^2} \quad (2.4)$$

$$b = \frac{n(\Sigma XY) - (\Sigma X)(\Sigma Y)}{n(\Sigma X^2) - (\Sigma X)^2} \quad (2.5)$$

## BAB 3

### METODE PENELITIAN

Pada bab ini akan diuraikan tentang metodologi penelitian tentang pengembangan alat pengukur pH berbasis fluoresen optik. Tahapan penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan penelitian.

Pada proses berjalannya penelitian, peneliti akan melakukan studi literatur yang lebih mendalam terkait pewarna fluoresen, pH larutan dan instrumen optik. Setelah studi literatur dirasa sudah cukup mendalam, proses penelitian dilanjutkan dengan melakukan pengembangan alat pengukur pH dan peralatan instrumen optik yang akan digunakan, kemudian dilanjutkan dengan mendesain alat dan melakukan

kalibrasi serta prosedur uji coba deteksi. Prosedur uji coba deteksi dilakukan pada kondisi ruangan yang tertutup dengan suhu ruangan 25°C sampai 30°C. Apabila hasil prosedur uji coba pengukuran pH telah memberikan hasil yang cukup memuaskan, maka dilanjutkan dengan melakukan analisa dan membuat kesimpulan. Proses penelitian kemudian dilanjutkan dengan membuat buku dan jurnal sebagai hasil akhir dari penelitian.

### 3.1 Studi Literatur

Pada proses studi literatur akan dilakukan pengumpulan seluruh informasi terkait hal-hal yang diperlukan dalam proses penelitian. Informasi yang didapat melalui studi literatur akan dijadikan sebagai langkah awal untuk melanjutkan ke proses berikutnya. Sumber-sumber studi literatur yang akan dilakukan dapat berupa buku, jurnal ilmiah, dan artikel ilmiah. Dalam proses ini peneliti akan mempelajari literatur mengenai pH suatu larutan, pewarna fluoresen dan karakteristik dan instrumen optik. Kemudian yang terakhir adalah mempelajari literatur mengenai pengkondisi sinyal. Gambar 3.2 menunjukkan diagram blok dari studi literatur.



Gambar 3.2 Diagram blok studi literatur.



Dalam proses studi literatur, pewarna fluoresen dipilih berdasarkan dari beberapa literatur yang terkait dengan penelitian yang sudah pernah dilakukan, sehingga dari pemilihan pewarna fluoresen ini peneliti dapat menentukan spesifikasi dari peralatan instrumen optik yang akan digunakan.

### **3.2 Reagen dan Material**

Adapun reagen yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Pewarna fluoresen berjenis pyranine.
2. Isopropyl alkohol 99%.
3. 100 mL Natrium Hidroksida (NaOH) dengan konsentrasi 1 Molar.
4. 100 mL Asam Klorida (HCL) dengan konsentrasi 1 Molar.
5. 1 Liter air suling.

Adapun material yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Pipet tetes 1 mL dan 3 mL.
2. Gelas ukur 10 mL.
3. Tabung reaksi (10mm x 75mm).

### **3.3 Hardware dan Software**

Adapun hardware yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Mikrokontroler Arduino Nano.
2. Sensor AS7262.
3. Sensor TSL2561.
4. Photodiode dan photoresistor.
5. Led dengan panjang gelombang 405 nm.
6. Modul wireless NRF24L01.
7. Power Supply 5V.
8. LCD 16 x 2.

9. PH meter digital PH-201 dengan elektroda pH PE-03 dari Lutron Electronic Enterprise CO., LTD.
10. Komputer atau laptop.

Adapun software yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Arduino IDE 1.8.
2. NetBeans IDE 8.2.
3. ISM band (Industrial, Scientific and Medical) band.

ISM (Industrial, Scientific and Medical) band adalah spektrum radio yang dapat digunakan untuk tujuan apa pun tanpa lisensi di sebagian besar negara. Band frekuensi 2,4 GHz didefinisikan dalam kisaran 2,4000 GHz hingga 2,4835 GHz dengan lebar pita 83,5 MHz. Penggunaan 2,4 GHz ISM untuk LAN nirkabel telah ditentukan oleh IEEE (Institute of Electrical and Electronics Engineers) dalam standar protokol 802.11-2007. Selain penggunaan peralatan LAN nirkabel, frekuensi ISM 2,4 GHz juga ditujukan untuk oven microwave, telepon rumah nirkabel, monitor, dan kamera video nirkabel. Karena penunjukannya pada banyak perangkat, frekuensi 2,4 GHz ISM meningkatkan potensi interferensi atau gangguan sinyal [26].

Seluruh hardware yang digunakan dalam penelitian diperoleh dengan harga yang terjangkau dan mudah didapatkan.

### **3.4 Mempersiapkan Pewarna Fluoresen**

Pada penelitian ini, pewarna fluoresen pyranine diperoleh dengan cara mengekstraksi sebuah stabilo berwarna kuning dengan larutan 25 ml isopropyl alkohol 99%. Hasil ekstraksi tersebut disimpan di dalam wadah steril dengan suhu ruangan 25°C sampai 30°C.

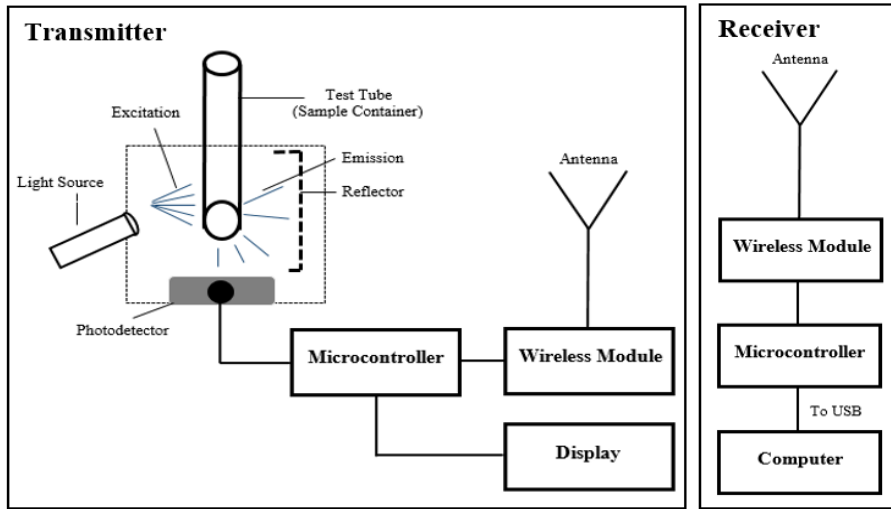
### **3.5 Pengembangan Alat Pengukur pH**

Pada tahap pengembangan alat pengukur pH ini dilakukan perancangan instrumen optik, perancangan pengkondisi sinyal dan perancangan aplikasi untuk memonitoring data.

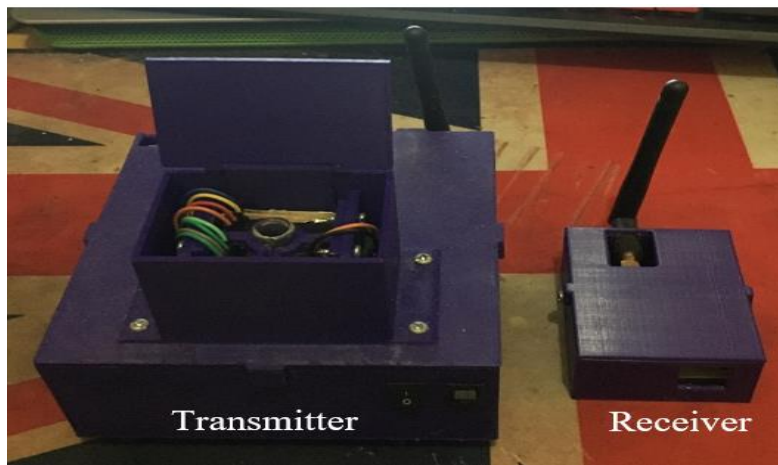
#### **3.5.1 Perancangan Instrumen Optik**

Pada tahap ini, perancangan instrumen dilakukan dengan memilih sumber cahaya berdasarkan pewarna fluoresen yang digunakan. Pewarna fluoresen pyranine memiliki panjang gelombang absorpsi puncak pada 405 nm. Sehingga sumber cahaya yang digunakan berada pada panjang gelombang 405 nm. Setelah sumber cahaya ditentukan kemudian menentukan fotodetektor, fotodetektor ditentukan berdasarkan dari spektrum emisi pewarna fluoresen yang digunakan, panjang gelombang puncak emisi dari pyranine adalah 515 nm. Dari profil spektrum emisi tersebut, maka fotodetektor berupa photoresistor, fotodiode, sensor TSL2561 dan sensor AS7262 diuji coba untuk mengetahui sensitivitas dari masing-masing sensor tersebut terhadap intensitas emisi yang dipancarkan oleh pewarna fluoresen. Fotodetektor diatas diuji karena memiliki respon sensitivitas yang dapat merespon panjang gelombang dari intensitas emisi fluoresen dengan panjang gelombang puncak di 515 nm.

Pada tahap perancangan instrumen optik terdiri dari transmitter dan receiver. Pada tahap perancangan bagian transmitter yang dilakukan pertama kali adalah menentukan sumber cahaya dan fotodetektor yang digunakan. Setelah sumber cahaya dan fotodetektor ditentukan, selanjutnya menentukan posisi dari sumber cahaya, fotodetektor dan wadah sampel. Posisi sumber cahaya, wadah sampel dan fotodetektor dibuat sejajar dan saling berdekatan. Peletakan posisi ini dipilih dengan alasan agar eksitasi cahaya benar-benar diserap oleh cairan pewarna fluoresen yang bercampur dengan cairan sampel seutuhnya dan emisi yang dilepaskan oleh pewarna fluoresen masuk ke fotodetektor seutuhnya. Hasil pembacaan intensitas emisi fluoresen kemudian dikirimkan ke receiver untuk dilakukan monitoring data. Gambar 3.3 menunjukkan rancangan instrumen optik dan gambar 3.4 menunjukkan alat instrumen optik.



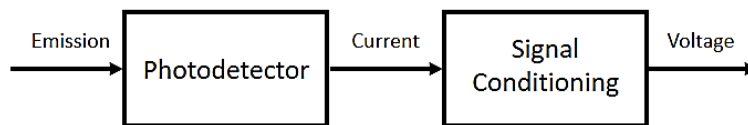
Gambar 3.3 Rancangan alat instrumen optik.



Gambar 3.4 Alat Instrumen optik.

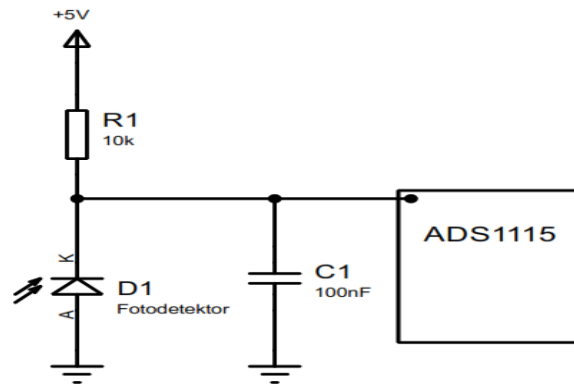
### 3.5.2 Perancangan Pengkondisi Sinyal

Pada tahap ini, sinyal dari fotodetektor LDR dan fotodiode yang dihasilkan masih dalam besaran arus sehingga dibutuhkan pengkondisi sinyal untuk mengubah arus tersebut menjadi tegangan. Pengkondisi sinyal ini menggunakan rangkaian pembagi tegangan. Gambar 3.5 menunjukkan diagram blok dari pengkondisi sinyal.



Gambar 3.5 Diagram blok pengkondisi sinyal.

Pada rangkaian pengkondisi sinyal ini, resistansi dari fotodetektor berubah ketika intensitas emisi dari fluoresen jatuh pada permukaan fotodetektor ( $R_{Fotodetektor}$ ) yang mengakibatkan terjadinya perubahan arus. Gambar 3.6 menunjukkan rangkaian pengkondisi sinyal.

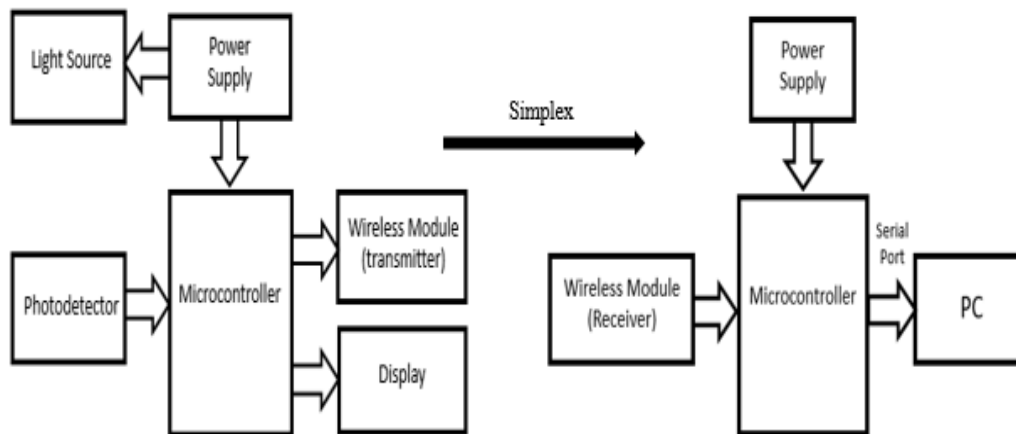


Gambar 3.6 Rangkaian pengkondisi sinyal.

Sehingga tegangan output ( $V_o$ ) yang diperoleh dituliskan dalam persamaan berikut:

$$V_o = \frac{V_s \times R_{Fotodetektor}}{(R_1 + R_{Fotodetektor})} \quad (3.1)$$

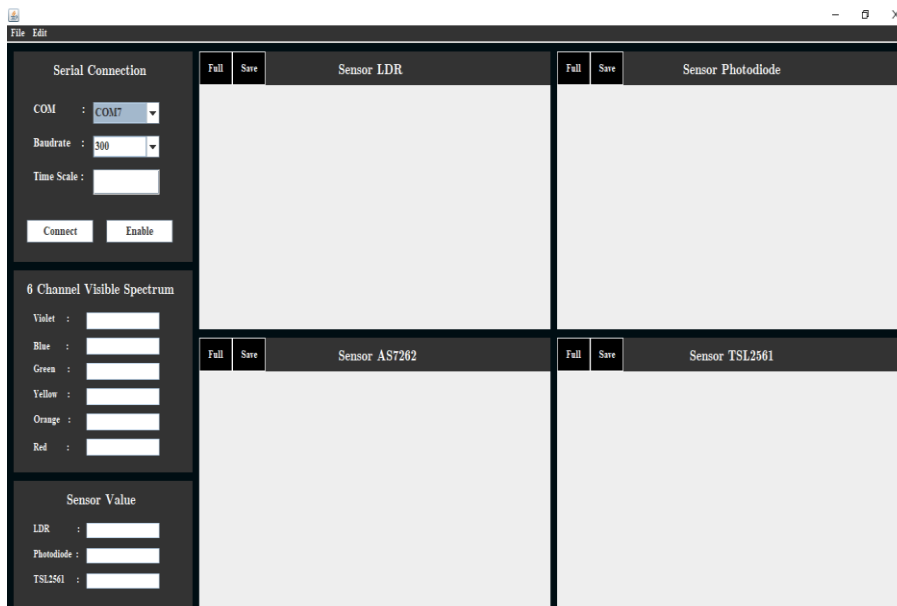
Dimana, tegangan sumber ( $V_s$ ) = 5 Volt,  $R_1$  = 10 K $\Omega$ . Penggunaan kapasitor ( $C_1$ ) 100 nF bertujuan untuk membuang noise dari tegangan sumber. Kemudian, tegangan output dari pengkondisi sinyal diproses dengan mikrokontroler Arduino Nano dengan memperbesar resolusi data tersebut menggunakan modul ADS1115. Modul ini merupakan modul ADC 16 bit. Penggunaan modul ini bertujuan untuk mendapatkan nilai yang lebih presisi. Untuk fotodetektor TSL2561 dan AS7262 tidak memerlukan perancangan pengkondisi sinyal karena di kedua modul ini pengkondisi sinyal sudah terintegrasi. Sinyal output dari fotodetektor yang sudah diolah menggunakan mikrokontroler selanjutnya ditampilkan pada LCD 16  $\times$  2 dan dikirim ke PC melalui modul *wireless* NRF24L01 (*transmitter*). Komunikasi ini dilakukan dalam satu arah (*simplex*). Sinyal dari tiap fotodetektor diamati menggunakan aplikasi yang sudah dirancang. Dimana sinyal dari tiap fotodetektor merupakan representasi dari intensitas emisi fluoresen. Gambar 3.7 menunjukkan diagram blok komunikasi antara *transmitter* dengan *receiver*.



Gambar 3.7 Diagram blok komunikasi antara *transmitter* dengan *receiver*.

### 3.5.3 Perancangan Aplikasi

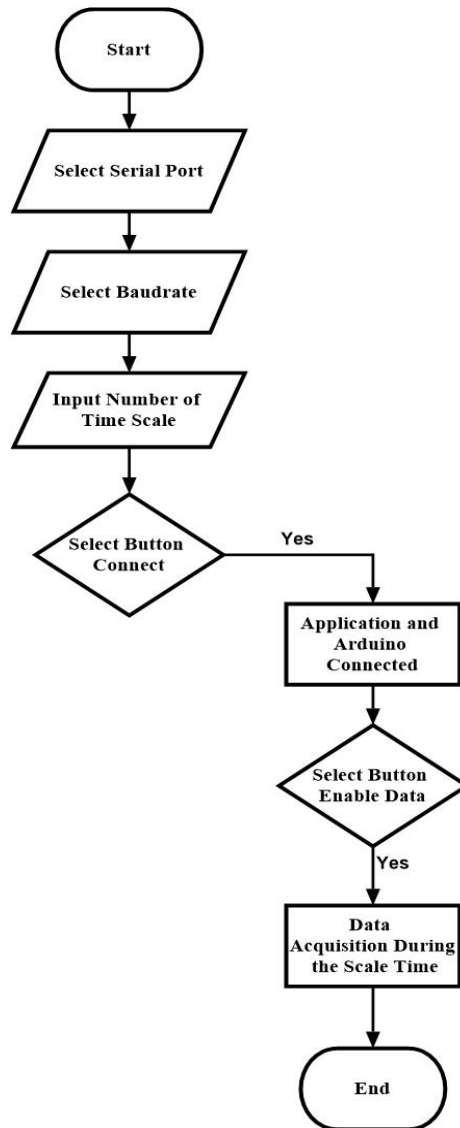
Aplikasi untuk memonitoring intensitas fluoresen dirancang dengan menggunakan bahasa pemrograman Java. Aplikasi ini menggunakan beberapa librari seperti Java Development Kit (JDK 1.8), JfreeChart dan JserialComm 2.4.0. Gambar 3.8 menunjukkan antarmuka aplikasi monitoring intensitas fluoresen.



Gambar 3.8 Antarmuka aplikasi monitoring intensitas fluoresen.

Aplikasi dapat berjalan dengan menyamakan baudrate dan nama komunikasi serial (COM) dengan modul penerima. Waktu pengambil data diset pada time scale. Kemudian dengan menekan tombol “Connect” aplikasi dan modul

receiver akan terhubung dan dengan menekan tombol “Enable” data dan kurva dari masing-masing sensor akan ditampilkan. Monitoring data akan berhenti ketika waktu data yang diset telah terpenuhi. Gambar 3.9 menunjukkan flowchart kerja dari aplikasi.



Gambar 3.9 Flowchart aplikasi.

### 3.6 Pengujian Alat Pengukur pH

Pada tahap pengujian alat pengukur pH ini ada beberapa tahap yang dilakukan diantaranya mengamati sensitivitas fotodetektor, mengamati efek photobleaching, mengamati efek perubahan volume pewarna fluoresen dan

kalibrasi. Seluruh pengujian dilakukan pada temperatur normal sekitar 25°C sampai 30°C tanpa ada interferensi cahaya dari luar.

### **3.6.1 Mengamati Respon Sensitivitas Fotodetektor**

Pada tahap ini, intensitas dari pewarna fluoresen diamati untuk melihat respon sensitivitas dari tiap fotodetektor. Pengamatan ini bertujuan untuk menentukan fotodetektor yang digunakan pada alat pengukur pH. Pengamatan dilakukan dengan mencampur 0.2 mL pyranine dengan 2 mL air suling untuk tiap pH air suling. Larutan dituangkan kedalam tabung reaksi untuk dilakukan pengujian. Tingkat pH pada air suling disesuaikan menggunakan larutan natrium hidroksida untuk menaikkan pH dan larutan asam klorida untuk menurunkan pH pada air suling. Kemudian, pH air suling diukur menggunakan pH meter digital. Tingkat pH air suling yang diukur yaitu 0.17, 2.16, 4.15, 5.61, 7.01, 8.38, 9.32, 10.3 dan 12.11. Tingkat pH air suling yang telah diukur menggunakan pH meter digital kemudian diukur dengan menggunakan alat pengukur pH yang telah dikembangkan. Dari hasil pengamatan fotodetektor yang mempunyai sensitivitas lebih baik akan dijadikan sebagai sensor utama pada pengembangan alat pengukur pH ini.

### **3.6.2 Mengamati Efek Photobleaching Pada Pewarna Fluoresen**

Pada tahap ini, efek photobleaching diamati untuk menentukan waktu yang tepat dan cepat untuk melakukan pengukuran pH pada larutan. Efek photobleaching merupakan fenomena ketika fluoresen kehilangan fluoresensi karena kerusakan yang disebabkan oleh cahaya. Hal ini menyebabkan hilangnya fluoresensi dan sinyal saat pengukuran sampel. Ketika cahaya dengan panjang gelombang yang tepat diarahkan pada fluoresen, fluoresen bertransisi dari kondisi dasar ke tahap singlet dan triplet yang tereksitasi. Dalam keadaan tereksitasi, fluoresen dapat berinteraksi dengan molekul lain dan menjalani modifikasi kovalen permanen [27]. Efek photobleaching diamati dengan menuangkan 0,5 mL larutan pewarna fluoresen yang dicampur dengan 2 mL air ke dalam tabung reaksi. Sumber cahaya 405 nm dieksitasikan menuju tabung reaksi dan intensitas emisi fluoresen diamati dalam 30 detik menggunakan aplikasi yang telah dibuat.



### **3.6.3 Mengamati Efek Pewarna Fluoresen**

Pada tahap ini, pengaruh volume fluoresen diamati menggunakan pH meter digital dan intensitas dari fluoresen diamati menggunakan instrumen optik. Pengukuran dilakukan dengan menetapkan 2 mL air suling dan campuran fluoresen dengan volume yang berbeda. Air suling sebagai sampel digunakan karena memiliki pH sekitar 7.0. Volume fluoresen diamati pada 0.2 mL, 0.4 mL, 0.5 mL, 0.6 mL dan 0.8 mL.

### **3.6.4 Kalibrasi Pada Alat**

Pada tahap ini, kalibrasi bertujuan untuk mendapatkan pengukuran yang tepat. Data intensitas fluoresen yang telah diperoleh dari hasil pengukuran dengan alat yang telah dikembangkan diproses menggunakan fungsi regresi linier. Air suling diukur menggunakan pH meter digital PH-201 dan elektroda pH PE-03 pada skala 2 hingga 12 dengan interval 0,5. Air suling yang telah diukur menggunakan pH meter digital kemudian diukur pada alat, prosedurnya adalah dengan menetapkan 0,5 mL larutan pewarna fluoresen yang dicampur dengan 2 mL air suling yang sebelumnya diukur menggunakan pH meter digital. PH-201 dan pH elektroda PE-03 untuk setiap pengukuran pH.

Pengkalibrasian dimulai dengan mengukur pH air suling dari 2, 2.5, 3, 3.5, 4, 4.5, 5, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11, 11.5 dan 12 menggunakan pH meter digital. pH air suling disesuaikan dengan larutan natrium hidroksida dan asam klorida. Pada setiap pengukuran pH air suling yang telah diukur dengan pH meter digital, hal yang sama dilakukan juga mengukur pH air suling dengan alat yang telah dikembangkan untuk mendapatkan intensitas fluoresen pada masing-masing pH. Mengukur pH air suling menggunakan alat yang telah dikembangkan dilakukan dengan mencampurkan 0,5 mL larutan pewarna fluoresen dengan 2 mL air suling ke dalam tabung reaksi yang sebelumnya telah diukur menggunakan pH meter digital.

Pengujian dengan alat yang telah dikembangkan dilakukan sebanyak empat kali menggunakan metode dan pH yang sama menggunakan aplikasi yang dikembangkan. Kemudian, intensitas fluoresen dari setiap pengukuran yang telah

dilakukan diproses menggunakan fungsi regresi linier. Aplikasi ini bekerja dengan membaca tingkat intensitas emisi yang dipancarkan dari larutan pewarna fluoresen yang telah dicampur dengan sampel air pH suling melalui sensor AS7262. Intesitas fluoresen yang dibaca oleh sensor dikirim ke komputer secara nirkabel menggunakan modul NRF24L01. Intensitas fluoresen yang dibaca oleh alat pengukur pH yang telah dikalibrasi merupakan fungsi dari pH larutan sampel yang diuji.

## BAB 4

### EKSPERIMEN DAN ANALISA DATA

Penelitian ini dilakukan untuk membuktikan bahwa alat pengukur pH yang dikembangkan mampu mengukur tingkat pH larutan dalam rentang yang luas secara cepat dan akurat. Pada penelitian ini, fotodetektor diuji coba untuk mengetahui respon sensitivitas dari masing-masing fotodetektor dalam membaca intensitas fluoresen. Setelah fotodetektor ditentukan kemudian dilakukan pengamatan efek photobleaching dari pewarna fluoresen, efek perubahan volume dari pewarna fluoresen. Setelah pengamatan dilakukan dilanjutkan dengan pengkalibrasian alat pengukur pH. Seluruh pengujian dilakukan pada kondisi temperatur normal sekitar 25°C sampai 30°C tanpa ada interferensi cahaya dari luar.

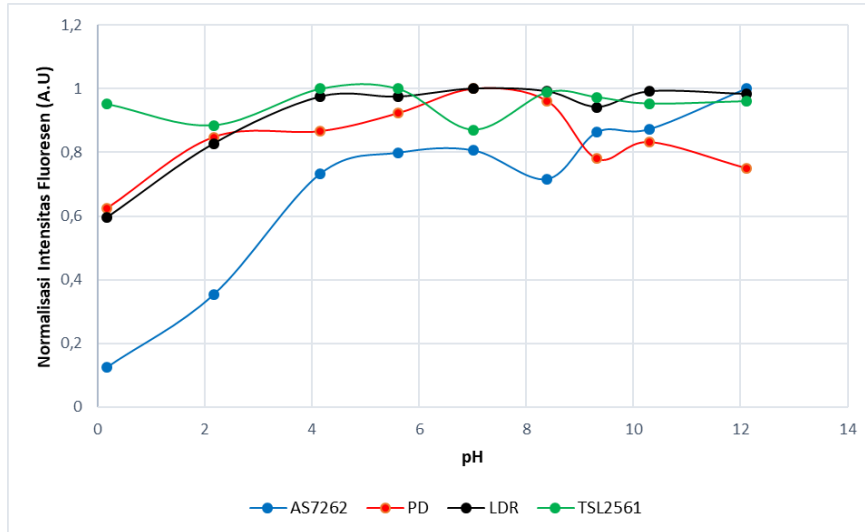
#### 4.1 Pengujian Respon Sensitivitas Fotodetektor

Pada penelitian ini, pengujian dimulai dengan mengamati intensitas fluoresen yang dibaca oleh fotodiode, photoresistor, sensor TSL2561 dan sensor AS7262. Pengamatan dilakukan dengan mencampur 0.2 mL pyranine dengan 2 mL air suling untuk tiap pH air suling. Larutan dituangkan kedalam tabung reaksi untuk dilakukan pengujian. Tingkat pH pada air suling disesuaikan menggunakan larutan natrium hidroksida untuk menaikkan pH dan larutan asam klorida untuk menurunkan pH pada air suling. Kemudian, pH air suling diukur menggunakan pH meter digital. Tingkat pH air suling yang diukur yaitu 0.17, 2.16, 4.15, 5.61, 7.01, 8.38, 9.32, 10.3 dan 12.11. Tingkat pH air suling yang telah diukur menggunakan pH meter digital kemudian diukur dengan menggunakan alat pengukur pH yang telah dikembangkan. Tabel 4.1 menunjukkan normalisasi intensitas fluoresen pada pH air suling. Gambar 4.1 menunjukkan intensitas fluoresen pada pH air suling.

Tabel 4.1 Normalisasi Intesitas Fluoresen Pada pH Air Suling.

pH Air Suling	Normalisasi Intensitas Fluoresen			
	AS7262	PD	LDR	TSL2561
0.17	0.125	0.624	0.596	0.952
2.16	0.352	0.848	0.827	0.885
4.15	0.732	0.867	0.975	0.999

5.61	0.799	0.923	0.975	1
7.01	0.806	1	1	0.871
8.38	0.715	0.962	0.991	0.989
9.32	0.863	0.78	0.943	0.974
10.3	0.872	0.832	0.991	0.954
12.11	1	0.75	0.983	0.961



Gambar 4.1 Intensitas fluoresen pada pH air suling.

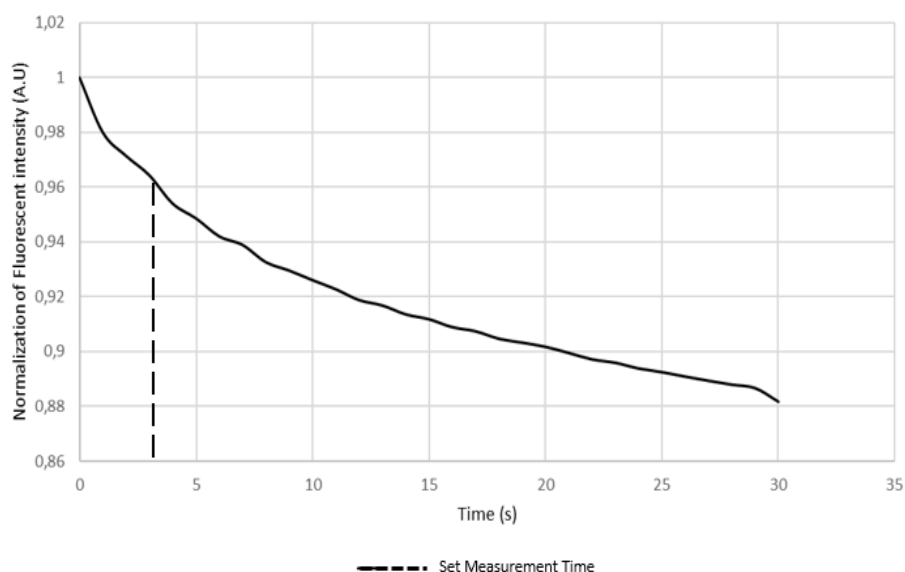
Dari hasil seluruh pengujian, Sensor AS7262 menunjukkan respon sensitivitas paling baik dalam menangkap intensitas emisi dari fluoresen diantara fotodetektor lainnya. Ini disebabkan didalam modul sensor AS7262 terdapat filter gaussian yang terintegrasi dengan fotodiode CMOS. Sehingga, sensor AS7262 dijadikan sebagai sensor utama pada alat untuk mengukur pH pada suatu larutan.

#### 4.2 Pengujian Alat Pengukur pH

Dari hasil seluruh pengujian yang telah dilakukan sebelumnya maka dapat dikembangkan alat untuk mengukur pH dari suatu larutan. Pada pengujian alat pengukur pH ini dilakukan pengamatan efek photobleaching, pengaruh volume dari fluoresen, kalibrasi dan pengujian alat dengan mengukur pH air.

### 4.2.1 Mengamati Efek Photobleaching

Dalam pengujian ini, efek photobleaching diamati untuk menentukan waktu yang tepat dan cepat untuk melakukan pengukuran pH pada air. Efek photobleaching diamati dengan menuangkan 0,5 mL larutan pewarna fluoresen yang dicampur dengan 2 mL air ke dalam tabung reaksi. Sumber cahaya 405 nm dieksitasikan menuju tabung reaksi dan intensitas emisi fluoresen diamati dalam 30 detik menggunakan aplikasi yang telah dibuat. Gambar 4.2 menunjukkan efek photobleaching pada 0.5 mL pewarna fluoresen dengan 2 mL air.



Gambar 4.2 Efek photobleaching pada 0.5 mL pewarna fluoresen dengan 2 mL air.

Dari hasil pengujian, efek photobleaching ditunjukkan pada Gambar 4.2. Karena efek photobleaching ini, waktu pengukuran ditetapkan pada detik ketiga atau pengukuran hanya dilakukan selama tiga detik. Alasan waktu pengukuran hanya dilakukan dalam hanya 3 detik yaitu untuk menghindari terjadinya kesalahan baca akibat tidak stabilnya efek photobleaching. Oleh karena itu waktu pengukuran hanya dilakukan dalam waktu 3 detik untuk memperoleh hasil pengukuran yang cepat dan akurat.

### 4.2.2 Mengamati Perubahan Volume Dari Fluoresen

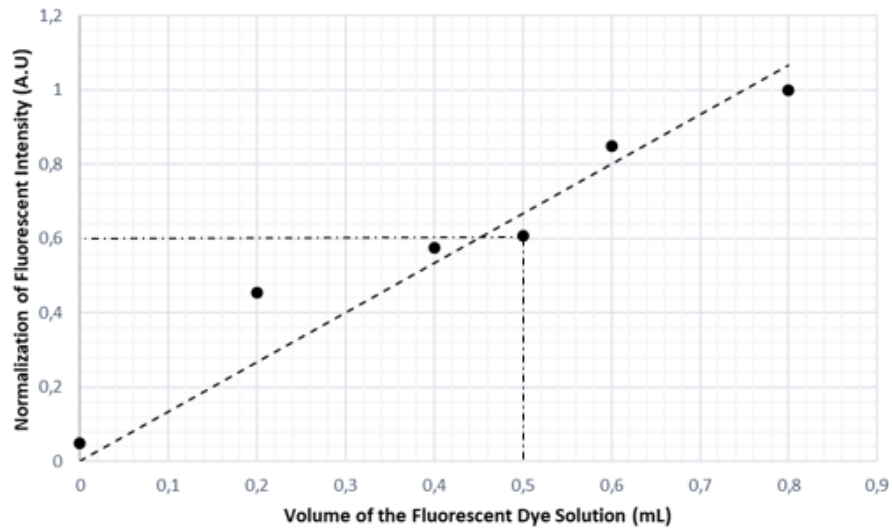
Pada pengujian ini, pengaruh volume fluoresen diamati menggunakan pH meter digital dan intensitas dari fluoresen diamati menggunakan instrumen optik.

Pengukuran dilakukan dengan dengan menetapkan 2 mL air suling dan campuran fluorezen dengan volume yang berbeda. Air suling sebagai sampel digunakan karena memiliki pH sekitar 7.0. Volume fluorezen diamati pada 0.2 mL, 0.4 mL, 0.5 mL, 0.6 mL dan 0.8 mL. Tabel 4.2 menunjukkan hasil pengukuran intensitas fluorezen pada pH yang diamati menggunakan pH meter digital dengan volume air suling 2 mL.

Tabel 4.2 Hasil Pengukuran Intensitas Fluorezen Pada pH yang Diamati Menggunakan pH Meter Digital Dengan Volume Air Suling 2 mL.

Volume Air	Volume Fluorezen (mL)	pH Meter Digital	Intensitas Fluorezen (A.U)
2 mL	0	6.27	659.33
	0.2	6.77	5948.17
	0.4	7.28	7543.38
	0.5	7.47	7960.93
	0.6	7.69	11129.94
	0.8	8.11	13099.87

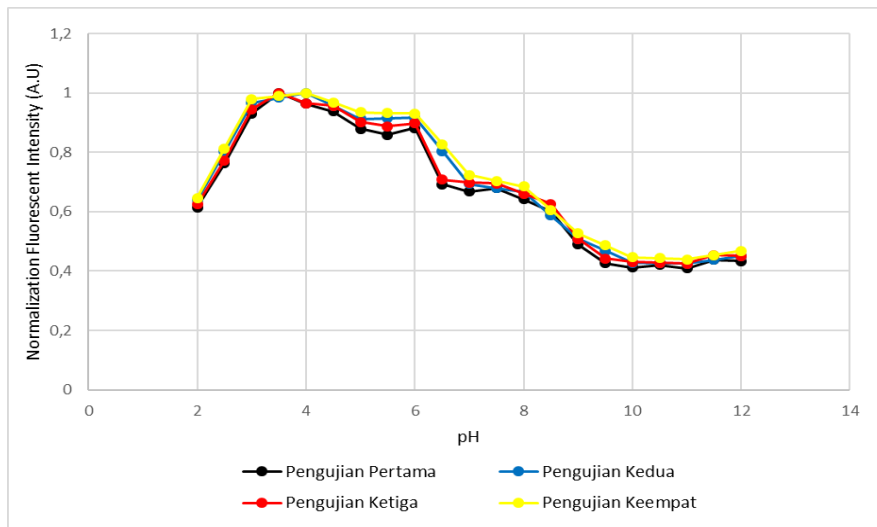
Dari hasil pengukuran, sensor dapat membaca nilai pH yang berbeda yang diukur dengan pH meter digital. Sehingga, diputuskan untuk menggunakan 0.5 mL volume larutan pewarna fluorezen. Pemilihan 0.5 mL volume pewarna fluorezen karena nilai intensitas fluorezen yang dibaca oleh sensor berada di tengah-tengah pengukuran beberapa volume larutan pewarna fluorezen, sehingga diasumsikan dengan volume 0,5 mL pewarna fluorezen, alat yang akan dikalibrasi memiliki rentang pengukuran yang luas. Selain itu, wadah sampel (tabung reaksi) memiliki kapasitas volume hanya 3 mL. Gambar 4.3 menunjukkan intensitas fluorezen dari beberapa volume pewarna fluorezen yang berbeda dengan campuran 2 mL air suling.



Gambar 4.3 Intensitas fluorensen dari beberapa volume pewarna fluorensen yang berbeda dengan campuran 2 mL air suling.

#### 4.2.3 Kalibrasi

Kalibrasi dilakukan dengan mengukur pH air suling dari rentang 2 sampai 12 dengan interval 0.5 menggunakan pH meter digital PH-201 dan elektroda pH PE-03. Air suling yang telah diukur menggunakan pH meter digital kemudian diukur pada alat dengan menetapkan 0,5 mL larutan pewarna fluorensen yang dicampur dengan 2 mL air suling yang sebelumnya diukur menggunakan pH meter digital. Pengujian dengan alat yang telah dikembangkan dilakukan sebanyak empat kali menggunakan metode dan pH yang sama menggunakan aplikasi yang dikembangkan. Kemudian, intensitas fluorensen dari setiap pengukuran yang telah dilakukan diproses menggunakan fungsi regresi linier. Aplikasi ini bekerja dengan membaca tingkat intensitas emisi yang dipancarkan dari larutan pewarna fluorensen yang telah dicampur dengan sampel air pH suling melalui sensor AS7262. Intensitas fluorensen yang dibaca oleh sensor dikirim ke komputer secara nirkabel menggunakan modul NRF24L01. Dari keempat pengujian yang telah dilakukan, hasil masing-masing pengujian menunjukkan hasil yang tidak terlalu berbeda. Gambar 4.4 menunjukkan intensitas fluorensen dari keempat pengujian yang telah dilakukan.



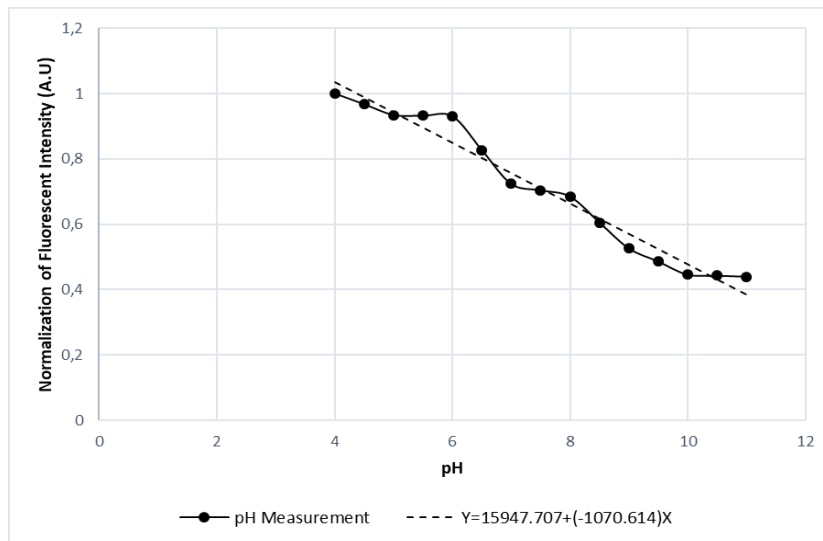
Gambar 4.4 Intesitas fluorens dari keempat pengujian yang telah dilakukan.

Dari hasil pengujian, data intensitas fluorens yang telah diperoleh kemudian diproses dengan menerapkan fungsi regresi linier. Fungsi ini dapat diterapkan karena kurva uji yang telah dilakukan menunjukkan perubahan pH linier terhadap intensitas fluorens. Dari keempat pengujian, dihitung rata-rata dari hasil intensitas fluorens masing-masing. Dari perolehan rata-rata intensitas fluorens yang diperoleh diproses dengan fungsi regresi linier untuk diterapkan pada alat yang telah dikembangkan. Data yang diolah untuk mendapatkan fungsi regresi linier adalah data intensitas fluorens dari pH 4,0 hingga 11,0. Hal ini disebabkan linearitas hanya ditunjukkan pada kisaran pH ini, seperti yang telah ditunjukkan pada Gambar. 4.4. Fungsi regresi linier yang diperoleh ditunjukkan dalam persamaan berikut.

$$Y = 15947.707 + (-1070.614)X \quad (4.1)$$

Dimana, Y adalah hasil pengukuran pH (Output), X adalah intensitas fluorensensi dari nilai pembacaan sensor (Input). Fungsi ini diterapkan dan digunakan untuk kalibrasi 0,5 mL larutan pewarna fluorescent dicampur dengan 2 mL air. Dari fungsi yang telah diperoleh, sensitivitas perangkat pengindra pH dapat diukur dari kisaran pH 4,0 hingga 11,0. Gambar 4.5 menunjukkan kurva sensitivitas fungsi regresi linier yang diterapkan pada alat yang telah dikembangkan.





Gambar 4.5 Kurva sensitivitas fungsi regresi linier yang diterapkan pada alat yang telah dikembangkan.

#### 4.2.4 Menguji Nilai Kesalahan Baca Pada Alat

Dalam pengujian ini, kemampuan pengukuran alat yang telah dikembangkan dibandingkan dengan pH meter digital dari rentang pH 4 hingga 10 dengan interval 0.5. Pengujian dilakukan dengan 0,5 mL larutan pewarna fluorescent yang dicampur dengan 2 mL air suling. pH air suling disesuaikan dengan larutan natrium hidroksida untuk meningkatkan pH dan larutan asam klorida untuk menurunkan pH air suling. Fungsi regresi linier  $Y = 15947.707 + (-1070.614) X$  diterapkan pada alat yang telah dikembangkan. Tabel 4.3 menunjukkan hasil perbandingan pengukuran alat yang telah dikembangkan dengan pH meter digital pada kisaran pH 4.0 hingga 10.0 dengan interval 0,5.

Tabel 4.3 Hasil Perbandingan Pengukuran Alat Yang Telah Dikembangkan Dengan pH Meter Digital Pada Kisaran pH 4,0 Hingga 10,0 Dengan Interval 0.5.

pH Meter Digital	Alat Yang Telah Dikembangkan.	$\Delta$ Error
4.01	4.11	0.1
4.50	4.67	0.17
5.01	5.04	0.03
5.52	5.61	0.09
6.03	5.96	0.07
6.53	6.46	0.07

7.03	6.89	0.14
7.53	7.39	0.14
8.02	7.88	0.14
8.52	8.40	0.12
9.01	9.33	0.32
9.52	9.76	0.24
10.03	9.94	0.09

Dari hasil Pengujian diatas, alat yang telah dikembangkan memiliki akurasi yang cukup baik. Namun, ini tidak sebgus akurasi pH meter digital. Nilai rata-rata kesalahan baca alat adalah  $\pm 0.13$ . Dari rata-rata kesalahan baca alat maka akurasi rata-rata alat adalah 98.13%. Alat yang telah dikembangkan memiliki sensitivitas terbaik pada kisaran pH 5.0 hingga 8.5.

#### 4.2.5 Mengukur pH Pada Sampel Acak

Pada pengujian ini, dilakukan pengukuran pH pada sampel yang belum diketahui menggunakan alat yang telah dikembangkan dan pH meter digital. Pengujian dilakukan dengan mencampur 0.5 mL larutan pewarna fluoresen dengan 2 mL larutan sampel. Sampel diperoleh dari beberapa tempat. Untuk sampel pertama yang diperoleh dari air PDAM, sampel kedua diperoleh dari air limbah masyarakat yang terletak di Jalan Kedung Pengkol, Surabaya, sampel ketiga diperoleh dari air hujan dan sampel keempat diperoleh dari air yang dicampur dengan asam klorida. Tabel 4.4 hasil perbandingan pengukuran pH dengan alat yang telah dikembangkan dan pH meter digital pada sampel. Gambar 4.6 menunjukkan hasil perbandingan pengukuran pH dengan alat yang telah dikembangkan dan pH meter digital pada sampel.

Tabel 4.4 Hasil Perbandingan Pengukuran pH Dengan Alat Yang Telah Dikembangkan dan pH Meter Digital Pada Sampel.

Sampel	pH Meter Digital	Alat Yang Telah Dikembangkan
Air PDAM	7.51	7.50
Air Limbah	9.15	9.35
Air Hujan	5.79	5.71

---

Air Yang  
Mengandung  
Asam Klorida

---

3.69

3.77



Gambar 4.6 Hasil perbandingan pengukuran pH dengan alat yang telah dikembangkan dan pH meter digital pada sampel.

Dimana, pH3 adalah hasil pengukuran pH menggunakan fungsi regresi linier  $Y = 15947.707 + (-1070.614)X$  yang diterapkan pada alat.

Semua pengukuran pH pada tiap pengujian dilakukan dalam kondisi suhu normal yaitu dari 25°C hingga 30°C. Waktu yang dibutuhkan alat untuk mengukur pH untuk setiap pengujian adalah 3 detik.

*Halaman ini sengaja dikosongkan*

## **BAB 5**

### **KESIMPULAN**

#### **5.1 Kesimpulan**

Kesimpulan yang diperoleh dari hasil penelitian yang telah dilakukan adalah pengembangan alat pengukuran pH menggunakan larutan pewarna fluoresen berhasil diterapkan untuk mengukur tingkat pH dalam air. Namun, ada batasan dalam pengukuran, seperti akurasi pengukuran yang tinggi hanya dapat diukur dalam kisaran pH 5.0 hingga 8.5 karena pyranine hanya memiliki sensitivitas tinggi pada pH tersebut. Alat yang telah dikembangkan memiliki akurasi 98.13% dan nilai kesalahan pembacaan  $\pm 0.13$  dari pengujian yang telah dilakukan. Intensitas fluoresen merupakan fungsi dari pengukuran pH. Alat ini hanya membutuhkan waktu kurang dari 3 detik untuk mengukur tingkat pH. Alat ini hanya dikalibrasi untuk 0.5 mL pyranine dengan 2 mL larutan yang akan diuji. Larutan pyranine dibuat dengan mengekstraksi sebuah stabilo kuning dengan 25 mL alkohol isopropil. Dalam wadah sampel terdapat 20% volume pyranine dan 80% volume sampel. Untuk melakukan pengukuran dengan sampel lain, alat ini harus dikalibrasi terlebih dahulu, kalibrasi dilakukan menggunakan metode yang sudah dijelaskan dalam penelitian ini. Alat ini tidak dapat mengukur pH < 4 dan pH > 10 karena terjadi perubahan warna larutan pada pyranine sehingga menghasilkan kesalahan pembacaan pengukuran oleh sensor AS7262 karena perubahan intensitas cahaya yang begitu signifikan. Alat ini dikembangkan dengan biaya yang terjangkau dan bahan yang mudah didapat. Alat pengukur pH ini juga terhubung ke aplikasi yang dikembangkan secara nirkabel yang berjalan pada sistem operasi windows.

#### **5.2 Saran**

Penelitian ini masih jauh dari kesempurnaan, sehingga pengukuran pH menggunakan sampel cairan dan campuran volume pewarna fluoresen yang berbeda diperlukan kalibrasi ulang.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] K. Kanokvalai, N. Piyavadee, and V. Supenya, "The pH of water from various sources: an overview for recommendation for patients with atopic dermatitis," *Asia Pac Allergy*, vol. 3, pp. 155-160, 2013.
- [2] A. L. M. Haider, "Health effects of alkaline diet and water, reduction of digestive-tract bacterial load, and earthing," *Alternative Therapies in Health and Medicine*, April 2016.
- [3] Ben Chioma, A. E. Jack, A. S. Philipokere, G. K., "A comparative study on the measurement of ph of water, using ph metre and water testing kit [testube method] in port harcourt," *IIARD.*, vol. 1, no. 3, 2015.
- [4] Xu Yu-Ling, Li Feng-Ye, Ndikuryayo Ferdinand, Yang Wen-Chao, and Wang Hong-Mei, "Cholinesterases and engineered mutants for the detection of organophosphorus pesticide residues," vol. 18, pp. 4281, 2018.
- [5] "Fluorescent Dyes" 25-Des-2016 [Online]. Available: <https://www.nature.com/subjects/fluorescent-dyes>. [Accessed: 28-Mei-2020].
- [6] Campbell Tessa N. and Choy Francis Y.M., "The effect of pH on Green Fluorescent Protein: a Brief Review," vol. 2, pp. 1-4, 2001.
- [7] J.N. Miller., "FLUORESCENCE | Quantitative Analysis," *Encyclopedia of Analytical Science (Second Edition)*, Elsevier, pp. 163-170, 2005.
- [8] Ewa Sikorska, Igor Khmelinskii, Marek Sikorski, "Fluorescence spectroscopy and imaging instruments for food quality evaluation," *Woodhead Publishing Series in Food Science, Technology and Nutrition*, pp. 491-533, 2019.
- [9] U. Sebastian, O. Alina, P. Guido, R. René, B. Nico and B. Luciano., "Pyranine-Modified Amphiphilic Polymer Conetworks as Fluorescent Ratiometric pH Sensors," *Macromolecular Rapid Communications.*, vol. 40, pp. 1900360, 2019.
- [10] Qi J, Liu D, Liu X, et al., "Fluorescent pH Sensors for Broad-Range pH Measurement Based on a Single Fluorophore," *Anal Chem.*, 2015;87(12):5897-5904.
- [11] C. Yuan, Y. Zhang, H. Xi and X. Tao., "An acidic pH fluorescent probe based on Tröger's base," *RSC Adv.*, 2017, 7, 55577.

- [12] Manuja Sharma, Jasmine Y. Graham, Philip A. Walczak, Ryan M. Nguyen, Lauren K. Lee, Matthew D. Carson, Leonard Y. Nelson, Shwetak N. Patel, Zheng Xu, Eric J. Seibel, "Optical pH measurement system using a single fluorescent dye for assessing susceptibility to dental caries," *J. Biomed. Opt.* 24(1) 017001 (8 January 2019).
- [13] Obare S.O., De Chandrima, Guo Wen, Haywood T.L., Samuels T.A., Adams C.P., Masika N.O., Murray D.H., Anderson G.A., Campbell Keith and Fletcher Kenneth, "Fluorescent Chemosensors for Toxic Organophosphorus Pesticides: A Review," vol. 10, pp. 7018-7043, 2010.
- [14] B. Claude, T.Craig and V. Rawee, "Interpretation of pH, Acidity, and Alkalinity in Aquaculture and Fisheries," *North American Journal of Aquaculture*, vol. 73, 2011.
- [15] Julian K. Trick, Marianne Stuart, Shaun Reeder, "Contaminated Groundwater Sampling and Quality Control of Water Analyses," *Environmental Geochemistry (Second Edition)*, Elsevier, pp. 25-45, 2018.
- [16] "Fluorophores, Dyes & Probes " 25-Des-2016 [Online]. Available: <https://www.biocompare.com/7670-Fluorophores-Dyes-Probes>. [Accessed: 28-Mei-2020].
- [17] National Center for Biotechnology Information. PubChem Database. Pyranine, CID=61388, <https://pubchem.ncbi.nlm.nih.gov/compound/Pyranine> (accessed on May 28, 2020).
- [18] R. W. Sabnis, "Handbook of Fluorescent Dyes and Probes," Wiley, 2015.
- [19] Manuja Sharma, Jasmine Y. Graham, Philip A. Walczak, Ryan M. Nguyen, Lauren K. Lee, Matthew D. Carson, Leonard Y. Nelson, Shwetak N. Patel, Zheng Xu, Eric J. Seibel, "Optical pH measurement system using a single fluorescent dye for assessing susceptibility to dental caries," *J. Biomed. Opt.* 24(1) 017001 (8 January 2019).
- [20] "Fluorescence Spectrum Viewer" [Online]. Available: <https://www.aatbio.com/fluorescence-excitation-emission-spectrum-graph-viewer>. [Accessed: 09-June-2020].
- [21] Fraden, Jacob, "Handbook of Modern Sensors," Physics, Design, and Applications, Fourth Editon, Springer, 2014.
- [22] OSI Optoelectronics, "Photodiode Characteristics and Applications," [Online]. Available: [www.osioptoelectronics.com](http://www.osioptoelectronics.com).



- [23] "TSL2561 Ambient Light Sensor". Available: <https://learn.adafruit.com/tsl2561>. [Accessed: 28-Mei-2020].
- [24] "AS7262 Consumer Grade Smart 6-Channel VIS Sensor". Available: <https://ams.com/as7262>. [Accessed: 28-Mei-2020].
- [25] Chatterjee, S. and Hadi, A.S. (2006). Simple Linear Regression. In Regression Analysis by Example (eds S. Chatterjee and A.S. Hadi).
- [26] Minhas, Neha and Kumar, Dr., "Performance Analysis of ISM Antennas Band: A Survey," International Journal of Advanced Computer Research., 2018.
- [27] P, Letter. "Photobleaching in Fluorescence Microscopy". AZoLifeSciences.<https://www.azolifesciences.com/article/Photobleaching-in-Fluorescence-Microscopy.aspx>. (accessed June 06, 2020).

*Halaman ini sengaja dikosongkan*

## LAMPIRAN

### 1. Program Alat Instrumen Optik

#### a. Transmitter (kalibrasi)

```
#include "Adafruit_AS726x.h"
#include <SparkFunTSL2561.h>
#include <Adafruit_ADS1015.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

//Setup LCD
LiquidCrystal_I2C lcd(0x3F, 16, 2);

//Setup NRF24L01
RF24 radio(9, 10);
byte addresses[][6] = {"1Node", "2Node"};

//Setup ADS1115
Adafruit_ADS1115 ads;

//Setup TSL2561
SFE_TSL2561 light;
boolean gain; // Gain setting, 0 = X1, 1 = X16;
unsigned int ms; // Integration ("shutter") time in milliseconds

//Setup AS7262
Adafruit_AS726x ams;
uint16_t sensorValues[AS726x_NUM_CHANNELS];
```

```

//Declare Structure Program untuk dikirim
struct Kirim {
    uint16_t adcLDR, adcPD;
    unsigned int TSLvis, TSLir;
    uint16_t V, B, G, Y, O, R;
    uint8_t temp;
};
typedef struct Kirim Package;
Package kirim;

//Declare variable and pin
const int buttonPin = 2; // the pin that the pushbutton is attached to
const int ledUngu = 3; // the pin that the LED is attached to

int buttonState;
int count = 0;
long durasi = 0;
long jeda = 1000;
long durasiCapture = 0;
long jedaCapture = 10000;
int sensor1 = 0;
int mundur = 3;
void(* reset) (void) = 0;

//Function Regression Linear
double Ga = 15947.71;
double Gb = -1070.61;

void setup() {
    pinMode(buttonPin, INPUT_PULLUP);
    pinMode(ledUngu, OUTPUT);
}

```

```

//digitalWrite(ledUngu, LOW);
Serial.begin(9600);

//Start LCD
lcd.begin();
lcd.setCursor(0,0);
lcd.print(" Biosensor pH ");
lcd.setCursor(0,1);
lcd.print(" 0711175004200 ");
delay(1000);

//Start AS7262
ams.begin();

//Start TSL2561
light.begin();
gain = 0;
unsigned char time = 2;
light.setTiming(gain,time,ms);
light.setPowerUp();

//Start ADS1115
ads.begin();

//Start NRF24L01
radio.begin();
ConfigNRF();

lcd.clear();
}

void loop() {
//Stop receive data
radio.stopListening();

/*-----AS7262-----*/
 kirim.temp = ams.readTemperature();
 ams.startMeasurement(); //begin a measurement

//wait till data is available

```

```

bool rdy = false;
while(!rdy){
    delay(5);
    rdy = ams.dataReady();
}
//read the values!
ams.readRawValues(sensorValues);
//ams.readCalibratedValues(calibratedValues);
kirim.V = sensorValues[AS726x_VIOLET];
kirim.B = sensorValues[AS726x_BLUE];
kirim.G = sensorValues[AS726x_GREEN];
kirim.Y = sensorValues[AS726x_YELLOW];
kirim.O = sensorValues[AS726x_ORANGE];
kirim.R = sensorValues[AS726x_RED];
Serial.print("V:["); Serial.print(kirim.V);
Serial.print("], B:["); Serial.print(kirim.B);
Serial.print("], G:["); Serial.print(kirim.G);
Serial.print("], Y:["); Serial.print(kirim.Y);
Serial.print("], O:["); Serial.print(kirim.O);
Serial.print("], R:["); Serial.print(kirim.R);
Serial.print("], Temp:"); Serial.println(kirim.temp);

/*-----TSL2561-----*/
if (light.getData(kirim.TSLvis,kirim.TSLir))
{
    Serial.print("TSL Visible:");
    Serial.print(kirim.TSLvis);
    Serial.print("], TSL IR1:");
    Serial.println(kirim.TSLir);
}

/*-----ADS1115-----*/

```

```

    kirim.adcPD = ads.readADC_SingleEnded(0);
    kirim.adcLDR = ads.readADC_SingleEnded(1);
    Serial.print("LDR:");
    Serial.print(kirim.adcLDR);
    Serial.print(", Photodiode:");
    Serial.println(kirim.adcPD);

    buttonState = digitalRead(buttonPin);
    if (buttonState == LOW) {
        digitalWrite(ledUngu, HIGH);
    }
    else{
        digitalWrite(ledUngu, LOW);
    }

    /*-----Send data-----*/
    radio.write(&kirim, sizeof(kirim));
}

void ConfigNRF() {
    radio.setPALevel(RF24_PA_MIN);
    radio.setDataRate(RF24_2MBPS);
    radio.setChannel(124);
    radio.openWritingPipe(addresses[1]);
    radio.openReadingPipe(1, addresses[0]);
}

```

#### b. Receiver (Kalibrasi)

```

#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#include <SPI.h>

RF24 radio(9,10); //ce,cs pin

```

```

byte addresses[][6] = {"1Node","2Node"};

struct Terima{
  int16_t adcLDR, adcPD;
  unsigned int TSLvis, TSLir;
  uint16_t V, B, G, Y, O, R;
  uint8_t temp;
};
typedef struct Terima PackageT;
PackageT terima;

int dataWaktu;
String komen;
char karakter;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  radio.begin();
  radio.setPALevel(RF24_PA_MIN);
  radio.setDataRate(RF24_2MBPS);
  radio.setChannel(124);
  radio.openWritingPipe(addresses[0]);
  radio.openReadingPipe(1, addresses[1]);
  radio.startListening();
}

void loop() {
  // put your main code here, to run repeatedly:
  radio.startListening();
  if(radio.available()){
    while(radio.available()){
      terima.V;
      terima.B;
      terima.G;
      terima.Y;
      terima.O;
      terima.R;
      terima.TSLvis;
      terima.TSLir;
      terima.adcLDR;
      terima.adcPD;
      terima.temp;

      radio.read(&terima, sizeof(terima));
    }
  }
  while (Serial.available(>0){
    karakter= Serial.read();
    komen.concat(karakter);
    delay(10);
  }
}

```



```

    }
    if (komen.equals("F") == true){
        Serial.println(dataWaktu++);
        Serial.println(terima.temp);
        Serial.println(terima.V);
        Serial.println(terima.B);
        Serial.println(terima.G);
        Serial.println(terima.Y);
        Serial.println(terima.O);
        Serial.println(terima.R);
        Serial.println(terima.TSLvis);
        Serial.println(terima.TSLir);
        Serial.println(terima.adcLDR);
        Serial.println(terima.adcPD);
    }
    delay(1000);
}

```

### c. Program Aplikasi

```

package arduinojavadataacquisition;

import com.fazecast.jSerialComm.SerialPort;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import java.io.OutputStream;

import java.io.PrintWriter;

import java.util.Scanner;

import javax.swing.ImageIcon;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.filechooser.FileNameExtensionFilter;

import org.jfree.chart.ChartFactory;

```

```

import org.jfree.chart.ChartPanel;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.JFreeChart;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;

/**
 *
 * @author sireg
 */
public class GUI extends javax.swing.JFrame {

    XYSeries seriesPD, seriesLDR, seriesV_PD, seriesV_LDR;
    XYSeriesCollection datasetPD, datasetLDR, datasetV_PD, datasetV_LDR;
    JFreeChart chartPD, chartLDR, chartV_PD, chartV_LDR;
    JFrame window1, window2, window3, window4;

    SerialPort portserial;

    private OutputStream output = null;
    private static final String START = "F";
    private static final String STOP = "S";
    private static final int[] BaudRate = {300, 1200, 2400, 4800,
        9600, 19200, 38400, 57600,
        74880, 115200, 230400, 250000,
        500000, 1000000, 2000000
    };

    static int count = 0;
    //int time = x/6;

    int getBR;
    int setBR;

    String lineclock, linePD, lineLDR, lineV_PD, lineV_LDR;
    int clock, dataPD, dataLDR;

```

```

float dataV_PD, dataV_LDR;

FileWriter fileWriter;

/**
 * Creates new form GUI
 */
public GUI() {
    initComponents();

    ImageIcon iconApps = new ImageIcon("src/icon/Frame Data.png");
    setIconImage(iconApps.getImage());

    SerialPort[] namaPort = SerialPort.getCommPorts();
    for(int i = 0; i < namaPort.length; i++){
        cmbPort.addItem(namaPort[i].getSystemPortName());
    }
    for(int j = 0; j < BaudRate.length; j++){
        String Baud = Integer.toString(BaudRate[j]);
        cmbBaudrate.addItem(Baud);
    }

    btnConnect.addActionListener(new ActionListener(){
        @Override
        public void actionPerformed(ActionEvent arg0){
            if(btnConnect.getText().equals("Connect")){
                portserial = SerialPort.getCommPort(cmbPort.getSelectedItem().toString());
                portserial.setComPortTimeouts(SerialPort.TIMEOUT_SCANNER, 0, 0);
                getBR = portserial.getBaudRate();
                String s = cmbBaudrate.getSelectedItem().toString();
                setBR = Integer.parseInt(s);

                String getTime = txtTimeScale.getText();
                int TimeScale = Integer.parseInt(getTime);
            }
        }
    });
}

```

```

int EndTime = TimeScale + 1;

if(getBR == setBR){
    portserial.openPort();

    if(portserial.openPort()){
        output = portserial.getOutputStream();
        btnConnect.setText("Disconnect");
        cmbPort.disable();
        cmbBaudrate.disable();
        txtTimeScale.disable();
        JOptionPane.showMessageDialog(null, "Serial Port is Opened");
    }
}
else if(getBR != setBR){
    btnConnect.setText("Connect");
    //portserial.closePort();
    JOptionPane.showMessageDialog(null, "Baudrate Doesn't Match");
}

window1 = new JFrame();
window1.setTitle("Reading Sensor PD");
window1.setSize(1280, 720);
window1.setLayout(new BorderLayout());
window1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
seriesPD = new XYSeries("Sensor PD");
datasetPD = new XYSeriesCollection(seriesPD);
chartPD = ChartFactory.createXYLineChart("", "Time (seconds)", "ADC",
datasetPD);
pnlPD.add(new ChartPanel(chartPD), BorderLayout.CENTER);
window1.add(new ChartPanel(chartPD), BorderLayout.CENTER);

```

```

window2 = new JFrame();
window2.setTitle("Reading Sensor LDR");
window2.setSize(1280, 720);
window2.setLayout(new BorderLayout());
window2.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
seriesLDR = new XYSeries("Sensor LDR");
datasetLDR = new XYSeriesCollection(seriesLDR);
chartLDR = ChartFactory.createXYLineChart("", "Time (seconds)", "ADC",
datasetLDR);
pnlLDR.add(new ChartPanel(chartLDR), BorderLayout.CENTER);
window2.add(new ChartPanel(chartLDR), BorderLayout.CENTER);

window3 = new JFrame();
window3.setTitle("Reading Voltage PD");
window3.setSize(1280, 720);
window3.setLayout(new BorderLayout());
window3.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
seriesV_PD = new XYSeries("Voltage PD");
datasetV_PD = new XYSeriesCollection(seriesV_PD);
chartV_PD = ChartFactory.createXYLineChart("", "Time (seconds)", "Voltage",
datasetV_PD);
pnlV_PD.add(new ChartPanel(chartV_PD), BorderLayout.CENTER);
window3.add(new ChartPanel(chartV_PD), BorderLayout.CENTER);

window4 = new JFrame();
window4.setTitle("Reading Voltage LDR");
window4.setSize(1280, 720);
window4.setLayout(new BorderLayout());
window4.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
seriesV_LDR = new XYSeries("Voltage LDR");
datasetV_LDR = new XYSeriesCollection(seriesV_LDR);
chartV_LDR = ChartFactory.createXYLineChart("", "Time (seconds)",
"Voltage", datasetV_LDR);

```

```

pnlV_LDR.add(new ChartPanel(chartV_LDR), BorderLayout.CENTER);
window4.add(new ChartPanel(chartV_LDR), BorderLayout.CENTER);

```

```

Thread thread = new Thread(){
    @Override
    public void run(){
        try (Scanner scanner = new Scanner(portserial.getInputStream())) {
            while(scanner.hasNextLine()){
                try {
                    //Scan Input Data From Arduino
                    lineclock = scanner.nextLine();
                    linePD = scanner.nextLine();
                    lineLDR = scanner.nextLine();
                    lineV_PD = scanner.nextLine();
                    lineV_LDR = scanner.nextLine();

                    clock = Integer.parseInt(lineclock);
                    dataPD = Integer.parseInt(linePD);
                    dataLDR = Integer.parseInt(lineLDR);
                    dataV_PD = Float.parseFloat(lineV_PD);
                    dataV_LDR = Float.parseFloat(lineV_LDR);

                    if(clock <= TimeScale){
                        seriesPD.add(clock, dataPD);
                        seriesLDR.add(clock, dataLDR);
                        seriesV_PD.add(clock, dataV_PD);
                        seriesV_LDR.add(clock, dataV_LDR);

                        pnlPD.repaint(); pnlPD.revalidate();
                        pnlLDR.repaint(); pnlLDR.revalidate();
                        pnlV_PD.repaint(); pnlV_PD.revalidate();
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

```

pnlV_LDR.repaint(); pnlV_LDR.revalidate();

window1.repaint(); window2.repaint();
window3.repaint(); window4.repaint();

txtPD();txtLDR();txtV_PD();txtV_LDR();txtTime();

TabelData();

if(dataPD >= 0 && dataPD <= 5){
    ind1.setBackground(Color.GREEN);
    ind2.setBackground(Color.WHITE);
    ind3.setBackground(Color.WHITE);
    ind4.setBackground(Color.WHITE);
    ind5.setBackground(Color.WHITE);
    ind6.setBackground(Color.WHITE);
}
else if(dataPD >= 6 && dataPD <= 10){
    ind1.setBackground(Color.GREEN);
    ind2.setBackground(Color.GREEN);
    ind3.setBackground(Color.WHITE);
    ind4.setBackground(Color.WHITE);
    ind5.setBackground(Color.WHITE);
    ind6.setBackground(Color.WHITE);
}
else if(dataPD >= 11 && dataPD <= 15){
    ind1.setBackground(Color.GREEN);
    ind2.setBackground(Color.GREEN);
    ind3.setBackground(Color.YELLOW);
    ind4.setBackground(Color.WHITE);
    ind5.setBackground(Color.WHITE);
    ind6.setBackground(Color.WHITE);
}

```

```

}
else if(dataPD >= 16 && dataPD <= 20){
    ind1.setBackground(Color.GREEN);
    ind2.setBackground(Color.GREEN);
    ind3.setBackground(Color.YELLOW);
    ind4.setBackground(Color.YELLOW);
    ind5.setBackground(Color.WHITE);
    ind6.setBackground(Color.WHITE);
}
else if(dataPD >= 21 && dataPD <= 25){
    ind1.setBackground(Color.GREEN);
    ind2.setBackground(Color.GREEN);
    ind3.setBackground(Color.YELLOW);
    ind4.setBackground(Color.YELLOW);
    ind5.setBackground(Color.RED);
    ind6.setBackground(Color.WHITE);
}
else if(dataPD >= 26 && dataPD <= 30){
    ind1.setBackground(Color.GREEN);
    ind2.setBackground(Color.GREEN);
    ind3.setBackground(Color.YELLOW);
    ind4.setBackground(Color.YELLOW);
    ind5.setBackground(Color.RED);
    ind6.setBackground(Color.RED);
}
else{
    ind1.setBackground(Color.WHITE);
    ind2.setBackground(Color.WHITE);
    ind3.setBackground(Color.WHITE);
    ind4.setBackground(Color.WHITE);
    ind5.setBackground(Color.WHITE);
    ind6.setBackground(Color.WHITE);
}

```



```

        }

    }

    else if(clock >= EndTime){
        scanner.close();
    }

}

}catch(Exception e){

}

}
scanner.close();
}
};
thread.start();
}
else{
    closeSerial();
}
}
});
}

```

```

/*-----Send Data to Arduino-----*/
private void DataOut(String data) {
    try {
        output.write(data.getBytes());
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(null, "No Data for Sending");
    }
}

```

```

/*-----Input Data to txt File-----*/
private void txtPD(){
    try(FileWriter fw = new FileWriter("DataPD.txt", true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter out = new PrintWriter(bw))

    {
        out.println(dataPD);

    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "No Data");
    }
}

```

```

private void txtLDR(){
    try(FileWriter fw = new FileWriter("DataLDR.txt", true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter out = new PrintWriter(bw))

    {
        out.println(dataLDR);
    }
}

```

```

    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "No Data");
    }
}

private void txtV_PD(){
    try(FileWriter fw = new FileWriter("DataV_PD.txt", true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter out = new PrintWriter(bw))
    {
        out.println(dataV_PD);

    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "No Data");
    }
}

private void txtV_LDR(){
    try(FileWriter fw = new FileWriter("DataV_LDR.txt", true);
        BufferedWriter bw = new BufferedWriter(fw);
        PrintWriter out = new PrintWriter(bw))
    {
        out.println(dataV_LDR);

    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "No Data");
    }
}

private void txtTime(){
    try(FileWriter fw = new FileWriter("DataTime.txt", true);
        BufferedWriter bw = new BufferedWriter(fw);

```

```

        PrintWriter out = new PrintWriter(bw))
    {
        out.println(clock);

    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "No Data");
    }
}

/*-----Input Data to Jtbale-----*/
private void TabelData(){
    tabelData.setValueAt("PD", 0, 0);tabelData.setValueAt("LDR", 1, 0);
    tabelData.setValueAt("V_PD", 2, 0);tabelData.setValueAt("V_LDR", 3, 0);

    tabelData.setValueAt(clock, 0, 1);tabelData.setValueAt(clock, 1, 1);
    tabelData.setValueAt(clock, 2, 1);tabelData.setValueAt(clock, 3, 1);

    tabelData.setValueAt(dataPD, 0, 2);tabelData.setValueAt(dataLDR, 1, 2);
    tabelData.setValueAt(dataV_PD, 2, 2);tabelData.setValueAt(dataV_LDR, 3, 2);

    tabelData.setValueAt("ADC", 0, 3);tabelData.setValueAt("ADC", 1, 3);
    tabelData.setValueAt("Voltage", 2, 3);tabelData.setValueAt("Voltage", 3, 3);

}

/*-----Close Serial Port-----*/
private void closeSerial(){
    if (portserial != null) {
        portserial.closePort();
        cmbPort.enable();
        cmbBaudrate.enable();
        txtTimeScale.enable();
    }
}

```

```

        portserial.closePort();
        btnConnect.setText("Connect");
        seriesPD.clear();
        seriesLDR.clear();
        seriesV_PD.clear();
        seriesV_LDR.clear();

        JOptionPane.showMessageDialog(null, "Serial Port is Closed");
    }
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jMenu1 = new javax.swing.JMenu();
    jPanel1 = new javax.swing.JPanel();
    jPanel8 = new javax.swing.JPanel();
    pnlPD = new javax.swing.JPanel();
    jPanel16 = new javax.swing.JPanel();
    jLabel4 = new javax.swing.JLabel();
    btnPD = new javax.swing.JButton();
    btnSavePD = new javax.swing.JButton();
    jPanel15 = new javax.swing.JPanel();
    jPanel2 = new javax.swing.JPanel();
    jLabel7 = new javax.swing.JLabel();
    cmbPort = new javax.swing.JComboBox<>();

```

```
cmbBaudrate = new javax.swing.JComboBox<>();
jLabel8 = new javax.swing.JLabel();
btnConnect = new javax.swing.JButton();
jLabel10 = new javax.swing.JLabel();
txtTimeScale = new javax.swing.JTextField();
btnEnableData = new javax.swing.JButton();
jPanel18 = new javax.swing.JPanel();
jLabel6 = new javax.swing.JLabel();
jPanel10 = new javax.swing.JPanel();
pnlV_PD = new javax.swing.JPanel();
jPanel19 = new javax.swing.JPanel();
jLabel9 = new javax.swing.JLabel();
btnV_PD = new javax.swing.JButton();
btnSaveV_PD = new javax.swing.JButton();
jPanel22 = new javax.swing.JPanel();
pnlV_LDR = new javax.swing.JPanel();
jPanel24 = new javax.swing.JPanel();
jLabel12 = new javax.swing.JLabel();
btnSaveV_LDR = new javax.swing.JButton();
btnV_LDR = new javax.swing.JButton();
jPanel14 = new javax.swing.JPanel();
pnlLDR = new javax.swing.JPanel();
jPanel21 = new javax.swing.JPanel();
jLabel5 = new javax.swing.JLabel();
btnLDR = new javax.swing.JButton();
btnSaveLDR = new javax.swing.JButton();
jPanel17 = new javax.swing.JPanel();
jPanel20 = new javax.swing.JPanel();
jLabel13 = new javax.swing.JLabel();
jScrollPane6 = new javax.swing.JScrollPane();
tabelData = new javax.swing.JTable();
jPanel23 = new javax.swing.JPanel();
```

```

jPanel25 = new javax.swing.JPanel();
jLabel14 = new javax.swing.JLabel();
ind1 = new javax.swing.JPanel();
ind6 = new javax.swing.JPanel();
jLabel3 = new javax.swing.JLabel();
ind5 = new javax.swing.JPanel();
ind4 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
ind3 = new javax.swing.JPanel();
ind2 = new javax.swing.JPanel();
jLabel1 = new javax.swing.JLabel();
jMenuBar1 = new javax.swing.JMenuBar();
jMenu2 = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
menuOpenImage = new javax.swing.JMenuItem();
menuRefresh = new javax.swing.JMenuItem();
jMenu3 = new javax.swing.JMenu();
menuInform = new javax.swing.JMenuItem();
menuVersion = new javax.swing.JMenuItem();

jMenu1.setText("jMenu1");

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Arduino-Java Data Acquisition");
setMaximumSize(new java.awt.Dimension(1366, 768));
setMinimumSize(new java.awt.Dimension(100, 100));
setPreferredSize(new java.awt.Dimension(1366, 768));
getContentPane().setLayout(null);

jPanel1.setBackground(new java.awt.Color(255, 255, 255));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

jPanel8.setBackground(new java.awt.Color(51, 51, 51));

jPanel8.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));

jPanel8.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

pnlPD.setBackground(new java.awt.Color(255, 255, 255));

pnlPD.setLayout(new java.awt.BorderLayout());

jPanel8.add(pnlPD, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, 490,
270));

jPanel16.setBackground(new java.awt.Color(0, 153, 0));

jPanel16.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));

jPanel16.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel4.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
jLabel4.setForeground(new java.awt.Color(255, 255, 255));
jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel4.setText("Photodiode");

jPanel16.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 7, 360,
30));

btnPD.setBackground(new java.awt.Color(0, 153, 0));

btnPD.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/fullscreen.png"))); // NOI18N
btnPD.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnPDActionPerformed(evt);
    }
});

jPanel16.add(btnPD, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 40, 40));

btnSavePD.setBackground(new java.awt.Color(0, 153, 0));

btnSavePD.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/save.png"))); // NOI18N

```



```

btnSavePD.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSavePDActionPerformed(evt);
    }
});

jPanel16.add(btnSavePD, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 0, 40,
40));

jPanel8.add(jPanel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 510, 40));

jPanel11.add(jPanel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 10, 510,
330));

jPanel15.setBackground(new java.awt.Color(51, 51, 51));

jPanel15.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));

jPanel15.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel2.setBackground(new java.awt.Color(255, 255, 255));

jPanel2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel7.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N
jLabel7.setForeground(new java.awt.Color(0, 0, 0));
jLabel7.setText("Serial port :");
jPanel2.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, 110,
40));

cmbPort.setBackground(new java.awt.Color(0, 153, 0));

cmbPort.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N

cmbPort.setForeground(new java.awt.Color(255, 255, 255));

cmbPort.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED,
java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray,
java.awt.Color.darkGray));

```

```
jPanel2.add(cmbPort, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 10, 140, 40));
```

```
cmbBaudrate.setBackground(new java.awt.Color(0, 153, 0));
```

```
cmbBaudrate.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N
```

```
cmbBaudrate.setForeground(new java.awt.Color(255, 255, 255));
```

```
cmbBaudrate.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED, java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray));
```

```
jPanel2.add(cmbBaudrate, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 60, 140, 40));
```

```
jLabel8.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N
```

```
jLabel8.setForeground(new java.awt.Color(0, 0, 0));
```

```
jLabel8.setText("time scale :");
```

```
jPanel2.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 110, 110, 40));
```

```
btnConnect.setBackground(new java.awt.Color(0, 153, 0));
```

```
btnConnect.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N
```

```
btnConnect.setForeground(new java.awt.Color(255, 255, 255));
```

```
btnConnect.setText("Connect");
```

```
btnConnect.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED, java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray));
```

```
btnConnect.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        btnConnectActionPerformed(evt);
```

```
    }
```

```
});
```

```
jPanel2.add(btnConnect, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 170, 130, 40));
```

```
jLabel10.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N
```

```

jLabel10.setForeground(new java.awt.Color(0, 0, 0));
jLabel10.setText("baudrate  :");

jPanel2.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 60, 110,
40));

txtTimeScale.setBackground(new java.awt.Color(0, 153, 0));

txtTimeScale.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
txtTimeScale.setForeground(new java.awt.Color(255, 255, 255));

txtTimeScale.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED,
java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray,
java.awt.Color.darkGray));

jPanel2.add(txtTimeScale, new org.netbeans.lib.awtextra.AbsoluteConstraints(130, 110,
140, 40));

btnEnableData.setBackground(new java.awt.Color(0, 153, 0));

btnEnableData.setFont(new java.awt.Font("Felix Titling", 1, 14)); // NOI18N
btnEnableData.setForeground(new java.awt.Color(255, 255, 255));

btnEnableData.setText("Enable Data");

btnEnableData.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED,
java.awt.Color.darkGray, java.awt.Color.darkGray, java.awt.Color.darkGray,
java.awt.Color.darkGray));

btnEnableData.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnEnableDataActionPerformed(evt);

    }

});

jPanel2.add(btnEnableData, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 220,
130, 40));

jPanel15.add(jPanel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, 280,
270));

jPanel18.setBackground(new java.awt.Color(0, 153, 0));

```

```

jPanel18.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));

jPanel18.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel6.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
jLabel6.setForeground(new java.awt.Color(255, 255, 255));
jLabel6.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel6.setText("connection");
jPanel18.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 300, 40));

jPanel15.add(jPanel18, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 300,
40));

jPanel1.add(jPanel15, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10, 300,
330));

jPanel10.setBackground(new java.awt.Color(51, 51, 51));
jPanel10.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));
jPanel10.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

pnlV_PD.setBackground(new java.awt.Color(255, 255, 255));
pnlV_PD.setLayout(new java.awt.BorderLayout());
jPanel10.add(pnlV_PD, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, 490,
270));

jPanel19.setBackground(new java.awt.Color(0, 153, 0));
jPanel19.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));
jPanel19.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel9.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
jLabel9.setForeground(new java.awt.Color(255, 255, 255));
jLabel9.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel9.setText("VOLTAGE PD");

```

```

jPanel19.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 0, 350,
40));

btnV_PD.setBackground(new java.awt.Color(0, 153, 0));

btnV_PD.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/fullscreen.png"))); // NOI18N

btnV_PD.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnV_PDActionPerformed(evt);

    }

});

jPanel19.add(btnV_PD, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 40,
40));

btnSaveV_PD.setBackground(new java.awt.Color(0, 153, 0));

btnSaveV_PD.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/save.png"))); // NOI18N

btnSaveV_PD.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        btnSaveV_PDActionPerformed(evt);

    }

});

jPanel19.add(btnSaveV_PD, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 0,
40, 40));

jPanel10.add(jPanel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 510,
40));

jPanel1.add(jPanel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(320, 350, 510,
330));

jPanel22.setBackground(new java.awt.Color(51, 51, 51));

jPanel22.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));

jPanel22.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

pnlV_LDR.setBackground(new java.awt.Color(255, 255, 255));

pnlV_LDR.setLayout(new java.awt.BorderLayout());

jPanel22.add(pnlV_LDR, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50,
490, 270));

jPanel24.setBackground(new java.awt.Color(0, 153, 0));

jPanel24.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));

jPanel24.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel12.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
jLabel12.setForeground(new java.awt.Color(255, 255, 255));
jLabel12.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel12.setText("Voltage LDR");
jPanel24.add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 0, 350,
40));

btnSaveV_LDR.setBackground(new java.awt.Color(0, 153, 0));

btnSaveV_LDR.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/save.png"))); // NOI18N
btnSaveV_LDR.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnSaveV_LDRActionPerformed(evt);
    }
});

jPanel24.add(btnSaveV_LDR, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 0,
40, 40));

btnV_LDR.setBackground(new java.awt.Color(0, 153, 0));

btnV_LDR.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/fullscreen.png"))); // NOI18N
btnV_LDR.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnV_LDRActionPerformed(evt);
    }
});

```

```

    }
});

jPanel24.add(btnV_LDR, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 40,
40));

jPanel22.add(jPanel24, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 510,
40));

jPanel1.add(jPanel22, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 350, 510,
330));

jPanel14.setBackground(new java.awt.Color(51, 51, 51));

jPanel14.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));

jPanel14.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

pnlLDR.setBackground(new java.awt.Color(255, 255, 255));

pnlLDR.setLayout(new java.awt.BorderLayout());

jPanel14.add(pnlLDR, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50, 490,
270));

jPanel21.setBackground(new java.awt.Color(0, 153, 0));

jPanel21.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));

jPanel21.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel5.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
jLabel5.setForeground(new java.awt.Color(255, 255, 255));

jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel5.setText("LDR");

jPanel21.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(80, 7, 360,
30));

btnLDR.setBackground(new java.awt.Color(0, 153, 0));

```

```

        btnLDR.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/fullscreen.png"))); // NOI18N

        btnLDR.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLDRActionPerformed(evt);
            }
        });

jPanel21.add(btnLDR, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 40, 40));

        btnSaveLDR.setBackground(new java.awt.Color(0, 153, 0));

        btnSaveLDR.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/save.png"))); // NOI18N

        btnSaveLDR.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnSaveLDRActionPerformed(evt);
            }
        });

jPanel21.add(btnSaveLDR, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 0, 40,
40));

jPanel14.add(jPanel21, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 510,
40));

jPanel1.add(jPanel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(840, 10, 510,
330));

jPanel17.setBackground(new java.awt.Color(51, 51, 51));

jPanel17.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));

jPanel17.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jPanel20.setBackground(new java.awt.Color(0, 153, 0));

jPanel20.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));

jPanel20.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```



```

jLabel13.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
jLabel13.setForeground(new java.awt.Color(255, 255, 255));
jLabel13.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel13.setText("Data Sensor");

jPanel20.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 300,
40));

jPanel17.add(jPanel20, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 300,
40));

tabelData.setBackground(new java.awt.Color(255, 255, 255));
tabelData.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
tabelData.setFont(new java.awt.Font("Times New Roman", 1, 12)); // NOI18N
tabelData.setForeground(new java.awt.Color(0, 0, 0));
tabelData.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Sensor", "Time (s)", "Data", "Unit"
    }
) {
    Class[] types = new Class [] {
        java.lang.String.class, java.lang.Integer.class, java.lang.Integer.class,
java.lang.String.class
    };
};

```

```

        public Class getColumnClass(int columnIndex) {
            return types [columnIndex];
        }
    });
    jScrollPane6.setViewportView(tabelData);

    jPanel17.add(jScrollPane6, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 50,
280, 120));

    jPanel11.add(jPanel17, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 350, 300,
180));

    jPanel23.setBackground(new java.awt.Color(51, 51, 51));
    jPanel23.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new
java.awt.Color(0, 0, 0)));
    jPanel23.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jPanel25.setBackground(new java.awt.Color(0, 153, 0));
    jPanel25.setBorder(javax.swing.BorderFactory.createMatteBorder(1, 1, 0, 1, new
java.awt.Color(0, 0, 0)));
    jPanel25.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel14.setFont(new java.awt.Font("Felix Titling", 1, 18)); // NOI18N
    jLabel14.setForeground(new java.awt.Color(255, 255, 255));
    jLabel14.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel14.setText("Indicator");
    jPanel25.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 300,
40));

    jPanel23.add(jPanel25, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 300,
40));

    ind1.setBackground(new java.awt.Color(255, 255, 255));
    ind1.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 0, new
java.awt.Color(0, 0, 0)));

```

```

ind1.setForeground(new java.awt.Color(255, 255, 255));
ind1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jPanel23.add(ind1, new org.netbeans.lib.awtextra.AbsoluteConstraints(60, 60, 30, 60));

ind6.setBackground(new java.awt.Color(255, 255, 255));
ind6.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 0, 2, 2, new
java.awt.Color(0, 0, 0)));
ind6.setForeground(new java.awt.Color(255, 255, 255));
ind6.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel3.setForeground(new java.awt.Color(0, 0, 0));
jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel3.setText("H");
jLabel3.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
jLabel3.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
ind6.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20, 20, 20));

jPanel23.add(ind6, new org.netbeans.lib.awtextra.AbsoluteConstraints(210, 60, 30, 60));

ind5.setBackground(new java.awt.Color(255, 255, 255));
ind5.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 0, 2, 0, new
java.awt.Color(0, 0, 0)));
ind5.setForeground(new java.awt.Color(255, 255, 255));
ind5.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jPanel23.add(ind5, new org.netbeans.lib.awtextra.AbsoluteConstraints(180, 60, 30, 60));

ind4.setBackground(new java.awt.Color(255, 255, 255));
ind4.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 0, 2, 0, new
java.awt.Color(0, 0, 0)));
ind4.setForeground(new java.awt.Color(255, 255, 255));
ind4.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

jLabel2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(0, 0, 0));
jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel2.setText("M");
jLabel2.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
jLabel2.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
ind4.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20, 20, 20));

jPanel23.add(ind4, new org.netbeans.lib.awtextra.AbsoluteConstraints(150, 60, 30, 60));

ind3.setBackground(new java.awt.Color(255, 255, 255));
ind3.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 0, 2, 0, new
java.awt.Color(0, 0, 0)));
ind3.setForeground(new java.awt.Color(255, 255, 255));
ind3.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
jPanel23.add(ind3, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 60, 30, 60));

ind2.setBackground(new java.awt.Color(255, 255, 255));
ind2.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 0, 2, 0, new
java.awt.Color(0, 0, 0)));
ind2.setForeground(new java.awt.Color(255, 255, 255));
ind2.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jLabel1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 0, 0));
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("L");
jLabel1.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
jLabel1.setVerticalTextPosition(javax.swing.SwingConstants.BOTTOM);
ind2.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20, 20, 20));

jPanel23.add(ind2, new org.netbeans.lib.awtextra.AbsoluteConstraints(90, 60, 30, 60));

```

```
jPanel1.add(jPanel23, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 540, 300, 140));
```

```
getContentPane().add(jPanel1);
```

```
jPanel1.setBounds(0, 0, 1370, 750);
```

```
jMenuBar1.setBackground(new java.awt.Color(51, 51, 51));
```

```
jMenuBar1.setForeground(new java.awt.Color(0, 0, 0));
```

```
jMenuBar1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
```

```
jMenu2.setBackground(new java.awt.Color(51, 51, 51));
```

```
jMenu2.setForeground(new java.awt.Color(255, 255, 255));
```

```
jMenu2.setText("File");
```

```
jMenu2.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
```

```
jMenuItem1.setBackground(new java.awt.Color(51, 51, 51));
```

```
jMenuItem1.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
```

```
jMenuItem1.setForeground(new java.awt.Color(255, 255, 255));
```

```
jMenuItem1.setText("Open Data Sensor As Text File");
```

```
jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        jMenuItem1ActionPerformed(evt);
```

```
    }
```

```
});
```

```
jMenu2.add(jMenuItem1);
```

```
menuOpenImage.setBackground(new java.awt.Color(51, 51, 51));
```

```
menuOpenImage.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
```

```
menuOpenImage.setForeground(new java.awt.Color(255, 255, 255));
```

```
menuOpenImage.setText("Open Image");
```

```
menuOpenImage.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
        menuOpenImageActionPerformed(evt);
```

```

    }
});
jMenu2.add(menuOpenImage);

menuRefresh.setBackground(new java.awt.Color(51, 51, 51));
menuRefresh.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
menuRefresh.setForeground(new java.awt.Color(255, 255, 255));
menuRefresh.setText("Refresh");
menuRefresh.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        menuRefreshActionPerformed(evt);
    }
});
jMenu2.add(menuRefresh);

jMenuBar1.add(jMenu2);

jMenu3.setBackground(new java.awt.Color(51, 51, 51));
jMenu3.setForeground(new java.awt.Color(255, 255, 255));
jMenu3.setText("About");
jMenu3.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N

menuInform.setBackground(new java.awt.Color(51, 51, 51));
menuInform.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
menuInform.setForeground(new java.awt.Color(255, 255, 255));
menuInform.setText("Information");
menuInform.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        menuInformActionPerformed(evt);
    }
});
jMenu3.add(menuInform);

```

```

menuVersion.setBackground(new java.awt.Color(51, 51, 51));
menuVersion.setFont(new java.awt.Font("Times New Roman", 1, 14)); // NOI18N
menuVersion.setForeground(new java.awt.Color(255, 255, 255));
menuVersion.setText("Version");
menuVersion.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        menuVersionActionPerformed(evt);
    }
});
jMenu3.add(menuVersion);

jMenuBar1.add(jMenu3);

setJMenuBar(jMenuBar1);

pack();
setLocationRelativeTo(null);
} // </editor-fold>

private void btnConnectActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnLDRActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    window2.setVisible(true);
}

private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Data data = new Data();

```

```

        data.setVisible(true);
    }

private void menuInformActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Information inform = new Information();
    inform.setVisible(true);
}

private void menuVersionActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    Version ver = new Version();
    ver.setVisible(true);
}

private void menuRefreshActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    SerialPort[] namaPort = SerialPort.getCommPorts();
    for(int i = 0; i < namaPort.length; i++){
        cmbPort.addItem(namaPort[i].getSystemPortName());
    }
    for(int j = 0; j < BaudRate.length; j++){
        String Baud = Integer.toString(BaudRate[j]);
        cmbBaudrate.addItem(Baud);
    }
    seriesPD.clear();
    seriesLDR.clear();
    seriesV_PD.clear();
    seriesV_LDR.clear();
}

private void btnPDActionPerformed(java.awt.event.ActionEvent evt) {

```



```

// TODO add your handling code here:
window1.setVisible(true);
}

private void btnSavePDActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        ChartUtilities.saveChartAsJPEG(new File("PD.jpeg"), chartPD, 1280, 720);
        JOptionPane.showMessageDialog(this, "Saved!");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(null, "No Chart for PD!");
    }
}

private void btnSaveLDRActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        ChartUtilities.saveChartAsJPEG(new File("LDR.jpeg"), chartLDR, 1280, 720);
        JOptionPane.showMessageDialog(this, "Saved!");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(null, "No Chart for LDR!");
    }
}

private void btnV_PDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    window3.setVisible(true);
}

private void btnSaveV_PDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        ChartUtilities.saveChartAsJPEG(new File("V_PD.jpeg"), chartV_PD, 1280, 720);

```

```

        JOptionPane.showMessageDialog(this, "Saved!");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(null, "No Chart for V_PD!");
    }
}

private void btnSaveV_LDRActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        ChartUtilities.saveChartAsJPEG(new File("V_LDR.jpeg"), chartV_LDR, 1280, 720);
        JOptionPane.showMessageDialog(this, "Saved!");
    } catch (IOException ex) {
        JOptionPane.showMessageDialog(null, "No Chart for V_LDR!");
    }
}

private void btnV_LDRActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    window4.setVisible(true);
}

private void menuOpenImageActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    JFileChooser fc = new JFileChooser("dist/");
    FileNameExtensionFilter filter=new FileNameExtensionFilter("images", "jpeg",
"jpg", "gif", "png");
    fc.setFileFilter(filter);
    int returnVal = fc.showOpenDialog(fc);
    String filePath = null;
    if(returnVal == JFileChooser.APPROVE_OPTION){
        filePath = fc.getSelectedFile().getAbsolutePath();
    }
}

```

```

        ShowImage show = new ShowImage(filePath);
    }

    private void btnEnableDataActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if(btnEnableData.getText().equals("Enable Data")){
            DataOut(START);
            btnEnableData.setText("Disable Data");
        }
        else if(btnEnableData.getText().equals("Disable Data")){
            DataOut(STOP);
            btnEnableData.setText("Enable Data");
        }
    }

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
    feel.
    * For details see
    http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    }
}

```

```

    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(GUI.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new GUI().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton btnConnect;
private javax.swing.JButton btnEnableData;
private javax.swing.JButton btnLDR;
private javax.swing.JButton btnPD;
private javax.swing.JButton btnSaveLDR;

```

```
private javax.swing.JButton btnSavePD;
private javax.swing.JButton btnSaveV_LDR;
private javax.swing.JButton btnSaveV_PD;
private javax.swing.JButton btnV_LDR;
private javax.swing.JButton btnV_PD;
private javax.swing.JComboBox<String> cmbBaudrate;
private javax.swing.JComboBox<String> cmbPort;
private javax.swing.JPanel ind1;
private javax.swing.JPanel ind2;
private javax.swing.JPanel ind3;
private javax.swing.JPanel ind4;
private javax.swing.JPanel ind5;
private javax.swing.JPanel ind6;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JMenu jMenuItem1;
private javax.swing.JMenu jMenuItem2;
private javax.swing.JMenu jMenuItem3;
private javax.swing.JMenuBar jMenuItemBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JPanel jPanel1;
```

```

private javax.swing.JPanel jPanel10;
private javax.swing.JPanel jPanel14;
private javax.swing.JPanel jPanel15;
private javax.swing.JPanel jPanel16;
private javax.swing.JPanel jPanel17;
private javax.swing.JPanel jPanel18;
private javax.swing.JPanel jPanel19;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel20;
private javax.swing.JPanel jPanel21;
private javax.swing.JPanel jPanel22;
private javax.swing.JPanel jPanel23;
private javax.swing.JPanel jPanel24;
private javax.swing.JPanel jPanel25;
private javax.swing.JPanel jPanel8;
private javax.swing.JScrollPane jScrollPane6;
private javax.swing.JMenuItem menuItemInform;
private javax.swing.JMenuItem menuItemOpenImage;
private javax.swing.JMenuItem menuItemRefresh;
private javax.swing.JMenuItem menuItemVersion;
private javax.swing.JPanel pnlLDR;
private javax.swing.JPanel pnlPD;
private javax.swing.JPanel pnlV_LDR;
private javax.swing.JPanel pnlV_PD;
private javax.swing.JTable tabelData;
private javax.swing.JTextField txtTimeScale;
// End of variables declaration
}

```

#### d. Program Setelah Kalibrasi

```

#include "Adafruit_AS726x.h"
#include <SparkFunTSL2561.h>

```

```

#include <Adafruit_ADS1015.h>
#include <nRF24L01.h>
#include <printf.h>
#include <RF24.h>
#include <RF24_config.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

//Setup LCD
LiquidCrystal_I2C lcd(0x3F, 16, 2);

//Setup NRF24L01
RF24 radio(9, 10);
byte addresses[][6] = {"1Node", "2Node"};

//Setup ADS1115
Adafruit_ADS1115 ads;

//Setup TSL2561
SFE_TSL2561 light;
boolean gain; // Gain setting, 0 = X1, 1 = X16;
unsigned int ms; // Integration ("shutter") time in milliseconds

//Setup AS7262
Adafruit_AS726x ams;
uint16_t sensorValues[AS726x_NUM_CHANNELS];

//Declare Structure Program untuk dikirim
struct Kirim {
    uint16_t adcLDR, adcPD;
    unsigned int TSLvis, TSLir;
    uint16_t V, B, G, Y, O, R;

```

```

uint8_t temp;
double pHG;
};
typedef struct Kirim Package;
Package kirim;

//Declare variable and pin
const int buttonPin = 2; // the pin that the pushbutton is attached to
const int ledUngu = 3; // the pin that the LED is attached to

int buttonState;
int count = 0;
long durasi = 0;
long jeda = 1000;
long durasiCapture = 0;
long jedaCapture = 10000;
int sensor1 = 0;
int mundur = 3;
void(* reset) (void) = 0;

//Function Regression Linear
double Ga = 15947.71;
double Gb = -1070.61;

void setup() {
  pinMode(buttonPin, INPUT_PULLUP);
  pinMode(ledUngu, OUTPUT);
  //digitalWrite(ledUngu, LOW);
  Serial.begin(9600);
  //Start LCD
  lcd.begin();
  lcd.setCursor(0,0);

```



```

lcd.print(" Biosensor pH ");
lcd.setCursor(0,1);
lcd.print(" 0711175004200 ");
delay(1000);
//Start AS7262
ams.begin();
//Start TSL2561
light.begin();
gain = 0;
unsigned char time = 2;
light.setTiming(gain,time,ms);
light.setPowerUp();
//Start ADS1115
ads.begin();
//Start NRF24L01
radio.begin();
ConfigNRF();
lcd.clear();
}

void loop() {
//Stop recieve data
radio.stopListening();
/*-----AS7262-----*/
kirim.temp = ams.readTemperature();
ams.startMeasurement(); //begin a measurement
//wait till data is available
bool rdy = false;
while(!rdy){
delay(5);
rdy = ams.dataReady();
}
}

```

```

//read the values!
ams.readRawValues(sensorValues);

//ams.readCalibratedValues(calibratedValues);
kirim.V = sensorValues[AS726x_VIOLET];
kirim.B = sensorValues[AS726x_BLUE];
kirim.G = sensorValues[AS726x_GREEN];
kirim.Y = sensorValues[AS726x_YELLOW];
kirim.O = sensorValues[AS726x_ORANGE];
kirim.R = sensorValues[AS726x_RED];
Serial.print("V:"); Serial.print(kirim.V);
Serial.print(", B:"); Serial.print(kirim.B);
Serial.print(", G:"); Serial.print(kirim.G);
Serial.print(", Y:"); Serial.print(kirim.Y);
Serial.print(", O:"); Serial.print(kirim.O);
Serial.print(", R:"); Serial.print(kirim.R);
Serial.print(", Temp:"); Serial.println(kirim.temp);

/*-----TSL2561-----*/
if (light.getData(kirim.TSLvis,kirim.TSLir))
{
  Serial.print("TSL Visible:");
  Serial.print(kirim.TSLvis);
  Serial.print(", TSL IR1:");
  Serial.println(kirim.TSLir);
}

/*-----ADS1115-----*/
kirim.adcPD = ads.readADC_SingleEnded(0);
kirim.adcLDR = ads.readADC_SingleEnded(1);
Serial.print("LDR:");
Serial.print(kirim.adcLDR);
Serial.print(", Photodiode:");

```

```

Serial.println(kirim.adcPD);

//Capture Data pH
buttonState = digitalRead(buttonPin);
if (buttonState == LOW) {

//Fungsi Regresi Linear Sederhana
kirim.pHG = (kirim.G - Ga) / Gb;
digitalWrite(ledUngu, HIGH);
if(millis() - durasi > jeda){
    count++;
    mundur--;
    durasi = millis();
}

if(count >= 0 && count <= 2){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Reading....");
    lcd.setCursor(0,1);
    lcd.print("Wait:");
    lcd.print(mundur);
    //delay(500);
    //digitalWrite(ledUngu, HIGH);
}
else if(count >= 3 && count <= 3){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("pH:");lcd.print(kirim.pHG);
    delay(10000);
}

```

```

else{
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Reading Done!");
    lcd.setCursor(0,1);
    lcd.print("Please Reset");
    digitalWrite(ledUngu, LOW);
}
}
else{
    digitalWrite(ledUngu, LOW);
    reset();
}

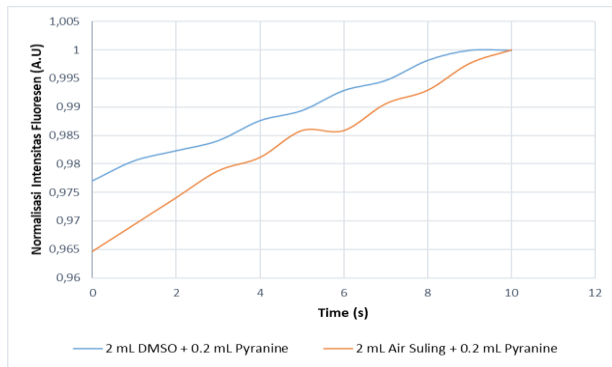
/*-----Send data-----*/
radio.write(& kirim, sizeof(kirim));

}

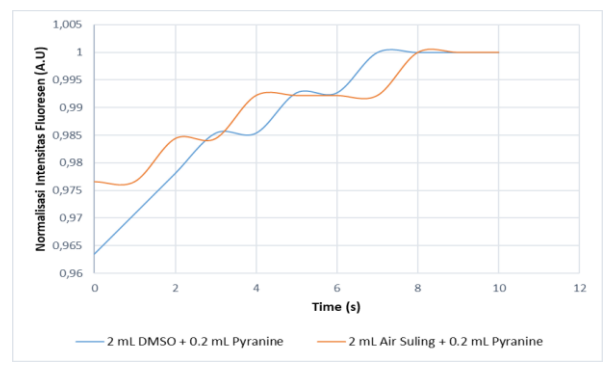
void ConfigNRF() {
    radio.setPALevel(RF24_PA_MIN);
    radio.setDataRate(RF24_2MBPS);
    radio.setChannel(124);
    radio.openWritingPipe(addresses[1]);
    radio.openReadingPipe(1, addresses[0]);
}

```

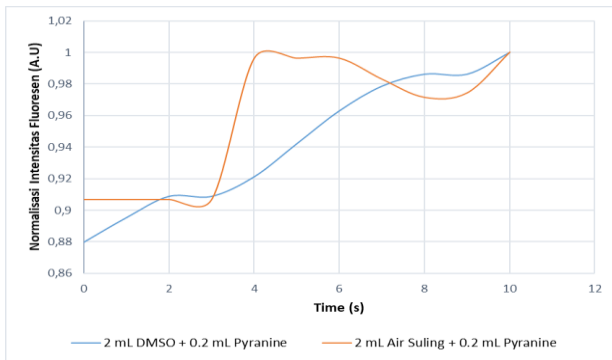
Gambar tambahan dari beberapa pengujian



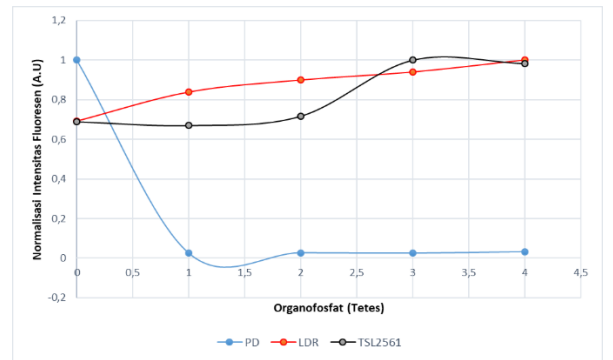
Gambar 1. Intensitas fluoresen pada 2 mL DMSO dengan 0.2 mL pyranine dan 2 mL air suling dengan 0.2 mL pyranine menggunakan fotodioda



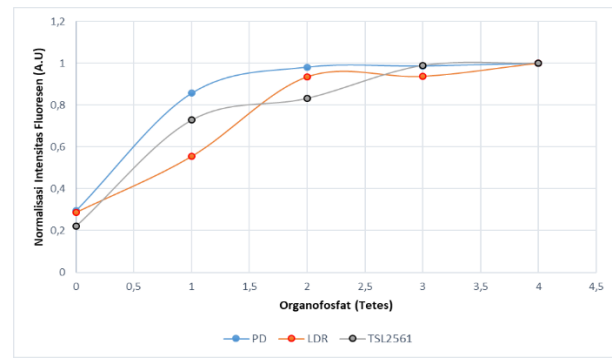
Gambar 2. Intensitas fluoresen pada 2 mL DMSO dengan 0.2 mL pyranine dan 2 mL air suling dengan 0.2 mL pyranine menggunakan LDR



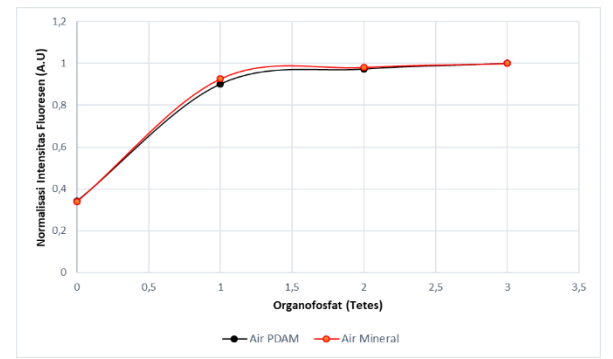
Gambar 3. Intensitas fluoresen pada 2 mL DMSO dengan 0.2 mL pyranine dan 2 mL air suling dengan 0.2 mL pyranine menggunakan TSL2561



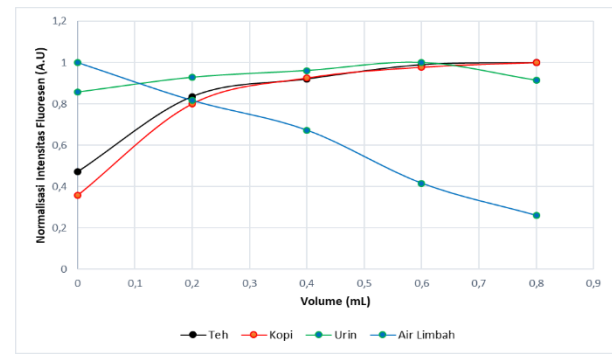
Gambar 4. Intensitas fluoresen pada DMSO dengan sampel organofosfat



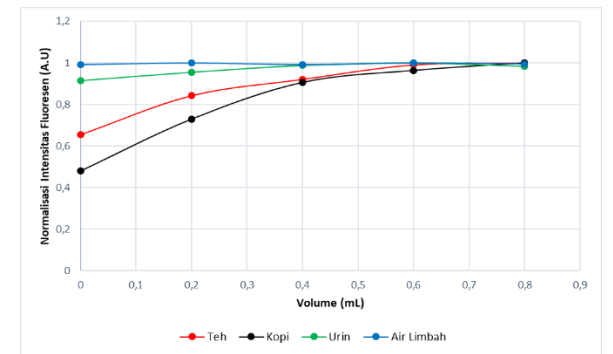
Gambar 5. Intensitas fluoresen pada air suling dengan sampel organofosfat



Gambar 6. Intensitas fluoresen pada air PDAM dan air mineral dengan sampel organofosfat menggunakan fotodioda



Gambar 7. Intensitas fluoresen pada air sampel teh, kopi, urin dan air limbah menggunakan fotodioda



Gambar 8. Intensitas fluoresen pada air sampel teh, kopi, urin dan air limbah menggunakan LDR

*Halaman ini sengaja dikosongkan*

## **DAFTAR INDEX**

*Halaman ini sengaja dikosongkan*



## RIWAYAT HIDUP



Rahmat Fauzi Siregar, Dilahirkan di Kota Padangsidempuan, Sumatera Utara pada tanggal 21 November 1994. Anak Kedua dari tiga bersaudara pasangan dari Gongmatua Siregar dan Ratna Sari. Peneliti menyelesaikan pendidikan Sekolah Dasar di SD N 15 Padangsidempuan pada tahun 2007, SMP di Nurul I'Imi Padangsidempuan pada tahun 2010 dan SMA N 2 Padangsidempuan pada tahun 2013. Pada tahun 2017, peneliti menyelesaikan program Sarjana Teknik Elektro di Universitas Muhammadiyah Yogyakarta. Pada tahun 2018, peneliti melanjutkan program pascasarjana Teknik Elektro di Institut Teknologi Sepuluh Nopember.