



**RANCANG BANGUN ANTENA MIKROSTRIP 2,4 GHZ  
*RECTANGULAR PATCH ARRAY 1X2* UNTUK PEMANTAUAN  
KUALITAS AIR DAN PEMBERI PAKAN IKAN KOI  
BERBASIS ANDROID**

***“RANCANG BANGUN PEMANTAUAN KUALITAS AIR DAN PEMBERI  
PAKAN IKAN KOI BERBASIS ANDROID”***

**TUGAS AKHIR**

**AFIF RIO SYAPUTRA  
1803332062**

**PROGRAM STUDI TELEKOMUNIKASI  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI JAKARTA  
2021**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN ANTENA MIKROSTRIP 2,4 GHZ  
RECTANGULAR PATCH ARRAY 1X2 UNTUK PEMANTAUAN  
KUALITAS AIR DAN PEMBERI PAKAN IKAN KOI  
BERBASIS ANDROID**

*“RANCANG BANGUN PEMANTAUAN KUALITAS AIR DAN PEMBERI  
PAKAN IKAN KOI BERBASIS ANDROID”*

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Diploma Tiga**

**AFIF RIO SYAPUTRA  
1803332062**

**PROGRAM STUDI TELEKOMUNIKASI  
JURUSAN TEKNIK ELEKTRO  
POLITEKNIK NEGERI JAKARTA**

**2021**



## HALAMAN PERNYATAAN ORISINALITAS

Tugas akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Afif Rio Syaputra

NIM : 1803332062

Tanda Tangan :

Tanggal : Juli 2021

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PENGESAHAN  
TUGAS AKHIR

Tugas Akhir diajukan oleh :

Nama : Afif Rio Syaputra  
NIM : 1803332062  
Program Studi : Teknik Telekomunikasi  
Judul Tugas Akhir : Rancang Bangun Antena Mikrostrip 2,4 GHz  
*Rectangular Patch Array* 1x2 untuk Pemantauan  
Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis  
Android

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada hari **Senin, 2 Agustus 2021** dan dinyatakan **LULUS**.

Pembimbing : Benny Nixon. S.T., M.T.  
NIP. 1968 1107 200003 1 001 (.....*11/8/21*.....)

**POLITEKNIK  
NEGERI  
JAKARTA**  
Depok, Agustus 2021  
Disahkan oleh  
Ketua Jurusan Teknik Elektro



Ir. Sri Danaryani, M.T.

NIP. 1963 0503 199103 2 001





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik.

Tugas akhir ini berjudul “Rancang Bangun Antena Mikrostrip 2,4 GHz *Rectangular Patch Array* 1x2 untuk Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android” guna membantu dalam pemeliharaan ikan koi dengan memantau kualitas air pada akuarium ikan dan pemberi pakan ikan secara otomatis maupun manual.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Benny Nixon, S.T., M.T., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan laporan Tugas Akhir ini;
2. Seluruh Staf Pengajar dan Karyawan Jurusan Teknik Elektro Politeknik Negeri Jakarta, khususnya Program Studi Telekomunikasi;
3. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
4. Naomi Lisda Nevita Banjarnahor yang telah menjadi rekan penulis serta selalu sabar terhadap penulis dalam penyusunan Tugas Akhir ini; dan
5. Sahabat yang telah banyak membantu penulis dalam menyelesaikan Tugas Akhir ini,

Akhir kata, penulis berharap semoga Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga laporan Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Bekasi, Juli 2021  
Penulis



## Rancang Bangun Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android

### ABSTRAK

*Ikan koi merupakan salah satu ikan hias yang memiliki bentuk tubuh dan warna yang indah sehingga bernilai ekonomis tinggi. Kualitas air secara umum menunjukkan kondisi air dalam keadaan baik atau tidak. Rancang bangun antenna microstrip 2,4GHz rectangular patch array 1x2 untuk pemantauan kualitas air dan pemberi pakan ikan koi berbasis android merupakan sistem yang dirancang untuk membantu masyarakat dalam budidaya maupun memelihara ikan koi. Ikan koi memerlukan kondisi air dengan suhu 25°C-30°C, tingkat keasaman (pH) air 6-7, dan kekeruhan air. Pada tugas akhir ini akan dirancang sistem pemantauan kualitas air dan pemberi pakan ikan koi berbasis android. Tujuan pembuatan sistem ini untuk memberikan notifikasi dan memantau suhu air, tingkat keasaman (pH) air, dan kekeruhan air melalui aplikasi android. Pada sistem ini menggunakan sensor suhu DS18B20, sensor pH dengan modul SEN0161, sensor turbidity dengan modul SEN0189, modul Real Time Clock (RTC), motor servo yang terhubung dengan mikrokontroler, dan aplikasi android. Motor servo berfungsi untuk memberikan pakan ikan secara otomatis maupun manual. Notifikasi akan bekerja apabila suhu air berada dibawah 25°C dan diatas 30°C, dan pH air berada dibawah 6 dan diatas 7 melalui Firebase Cloud Messaging. Hasil pengujian menunjukkan bahwa sensor suhu didapatkan nilai dengan selisih sekitar 0,01-0,25°C, sensor pH didapatkan nilai dengan selisih 0,03-1,33, sensor kekeruhan berfungsi sesuai dengan kondisi, dan motor servo bekerja dengan baik. Aplikasi android dapat mengakses data serta memberi input ke sistem melalui Firebase Realtime Database karena performasi dari jaringan internet dinilai bagus yaitu packet loss 0% dan besar delay 2,4ms-24ms dimana <150ms.*

**Kata Kunci :** *Android, Firebase, Sensor suhu DS18B20, Sensor pH SEN0161, Sensor turbidity SEN0189.*

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK  
NEGERI  
JAKARTA





## *Design and Build Water Quality Monitoring and Feeder of Koi Fish Based on Android*

### **ABSTRACT**

*Koi fish is one of the ornamental fish that has a beautiful body shape and color so it has high economic value. Water quality generally shows the condition of the water in good condition or not. The design of a 2.4GHz rectangular 1x2 microstrip antenna for monitoring water quality and feeding koi fish based on Android is a system designed to assist the community in cultivating and maintaining koi fish. Koi fish require water conditions with a temperature of 25°C-30°C, acidity (pH) of water 6-7, and water turbidity. In this final project, a water quality monitoring system and an android-based koi fish feeder will be designed. The purpose of making this system is to provide notifications and monitor water temperature, acidity (pH) of water, and water turbidity through an android application. This system uses a DS18B20 temperature sensor, a pH sensor with a SEN0161 module, a turbidity sensor with a SEN0189 module, a Real Time Clock (RTC) module, a servo motor connected to a microcontroller, and an android application. Servo motor serves to provide fish feed automatically or manually. The notification will work if the water temperature is below 25°C and above 30°C, and the water pH is below 6 and above 7 via Firebase Cloud Messaging. The test results show that the temperature sensor value is obtained with a difference of about 0.01-0.25 °C, the pH sensor is obtained a value with a difference of 0.03-1.33, the turbidity sensor functions according to the conditions, and the servo motor works well. Android applications can access data and provide input to the system through the Firebase Realtime Database because the performance of the internet network is considered good, namely 0% packet loss and 2.4ms-24ms delay where <150ms.*

**Keywords:** *Android, Firebase, Temperature sensor DS18B20, pH sensor SEN0161, Turbidity sensor SEN0189.*

#### **Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**POLITEKNIK  
NEGERI  
JAKARTA**



## DAFTAR ISI

	Halaman
HALAMAN SAMPUL.....	i
HALAMAN JUDUL .....	ii
HALAMAN PERNYATAAN ORISINALITAS .....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR.....	v
ABSTRAK .....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL .....	xii
DAFTAR LAMPIRAN .....	xiii
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	2
1.4 Luaran .....	2
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>4</b>
2.1 Ikan Koi .....	4
2.2 Arduino Uno .....	5
2.3 NodeMCU ESP8266.....	6
2.4 Arduino Integrated Development Environment (IDE).....	7
2.5 Sensor Suhu DS18B20.....	9
2.6 Sensor pH Modul SEN0161.....	10
2.7 Sensor <i>Turbidity</i> Modul SEN0189.....	11
2.8 <i>Real Time Clock</i> (RTC) .....	12
2.9 Motor Servo .....	13
2.10 Catu Daya.....	14
2.11 <i>Quality of Service</i> (QoS).....	15
2.11.1. <i>Delay</i> .....	15
2.11.2. <i>Throughput</i> .....	16
2.11.3. <i>Packet Loss</i> .....	16
2.12 <i>Firestore Realtime Database</i> .....	17
2.13 <i>Firestore Cloud Messaging</i> (FCM).....	18
2.14 XAMPP.....	18
2.15 <i>Android Studio Development Kit</i> (SDK) .....	19
<b>BAB III PERANCANGAN DAN REALISASI .....</b>	<b>22</b>
3.1 Perancangan Sistem .....	22
3.1.1 Deskripsi Sistem.....	22
3.1.2 Cara Kerja Sistem.....	23
3.1.3 Spesifikasi Alat.....	27

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.2	Perancangan Sistem dan Aplikasi Android.....	28
3.2.1	Perancangan Catu Daya ( <i>Power Supply</i> ).....	28
3.2.2	Perancangan Sistem Mikrokontroler .....	29
3.2.3	Perancangan Arduino Uno .....	30
3.2.4	Perancangan ESP8266.....	37
3.2.5	Perancangan Firebase <i>Realtime Database</i> .....	43
3.2.6	Perancangan Notifikasi.....	45
3.2.7	Perancangan Aplikasi Android.....	45
<b>BAB IV</b>	<b>PEMBAHASAN.....</b>	<b>48</b>
4.1.	Pengujian.....	48
4.2.	Pengujian Catu Daya.....	48
4.2.1	Prosedur Pengujian Catu Daya.....	48
4.2.2	Hasil Pengujian Catu Daya.....	49
4.2.3	Analisa Data Hasil Pengujian Catu Daya.....	49
4.3.	Pengujian Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Pada Sistem Mikrokontroler .....	50
4.3.1	Prosedur Pengujian Sistem Mikrokontroler .....	50
4.3.2	Hasil Pengujian Sistem Mikrokontroler .....	50
4.3.3	Analisa Data Hasil Pengujian Sistem Mikrokontroler .....	54
4.4.	Pengujian Kualitas Jaringan Internet .....	55
4.4.1	Prosedur Pengujian Kualitas Jaringan Internet.....	55
4.4.2	Data Hasil Pengujian .....	56
4.4.3	Analisa Hasil Pengujian .....	56
4.5	Pengujian Aplikasi Android.....	57
4.5.1	Prosedur Pengujian Aplikasi Android.....	57
4.5.2	Data Hasil Pengujian Aplikasi Android .....	57
4.5.3	Analisa Hasil Pengujian .....	59
4.6	Pengujian Keseluruhan Sistem .....	59
4.6.1	Analisa Pengujian Keseluruhan Sistem.....	60
<b>BAB V</b>	<b>PENUTUP.....</b>	<b>61</b>
5.1	Simpulan .....	61
5.2	Saran .....	62
<b>DAFTAR PUSTAKA .....</b>	<b>63</b>	
<b>DAFTAR RIWAYAT HIDUP PENULIS .....</b>	<b>65</b>	
<b>LAMPIRAN.....</b>	<b>86</b>	

## DAFTAR GAMBAR

	<b>Halaman</b>
Gambar 2. 1 Ikan Koi.....	4
Gambar 2. 2 <i>Pinout</i> Arduino Uno .....	5
Gambar 2. 3 <i>Pinout</i> NodeMCU ESP8266 .....	7
Gambar 2. 4 Tampilan Awal <i>Software</i> Arduino IDE.....	9
Gambar 2. 5 <i>Pinout</i> Sensor Suhu DS18B20 .....	9
Gambar 2. 6 Sensor pH Modul SEN0161 .....	10
Gambar 2. 7 Sensor <i>Turbidity</i> Modul SEN0189.....	11
Gambar 2. 8 Grafik Hubungan <i>Turbidity</i> dengan Tegangan.....	12
Gambar 2. 9 Bentuk Fisik RTC DS3231 .....	13
Gambar 2. 10 Bentuk Fisik Motor Servo MG996R.....	14
Gambar 2. 11 Rangkaian Catu Daya Menggunakan IC 7809.....	15
Gambar 2. 12 Logo Firebase <i>Realtime Database</i> .....	17
Gambar 2. 13 Logo Firebase <i>Cloud Messaging</i> .....	18
Gambar 2. 14 Tampilan Awal <i>Software</i> XAMPP.....	19
Gambar 2. 15 Tampilan Awal <i>Software</i> XAMPP.....	20
Gambar 2. 16 Logo Aplikasi <i>Android Studio</i> .....	21
Gambar 3. 1 Ilustrasi Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android .....	23
Gambar 3. 2 Diagram Blok Sistem Pemantauan Kualitas Air.....	23
Gambar 3. 3 Diagram Blok Sistem Pemberi Pakan Ikan Koi.....	25
Gambar 3. 4 <i>Flowchart</i> Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android .....	26
Gambar 3. 5 Diagram Skematik Rangkaian Catu Daya.....	28
Gambar 3. 6 <i>Layout</i> Rangkaian Catu Daya.....	29
Gambar 3. 7 Rangkaian Skematik Sistem Mikrokontroler.....	29
Gambar 3. 8 Tampilan Firebase <i>Console</i> .....	43
Gambar 3. 9 Variabel Data pada <i>Realtime Database</i> .....	44
Gambar 3. 10 Tampilan Indikasi Sukses Melakukan Koneksi Firebase.....	44
Gambar 3. 11 Tampilan Token yang Didapatkan .....	45
Gambar 3. 8 Tampilan <i>Splash Screen</i> .....	46
Gambar 3. 9 Tampilan <i>layer</i> ke-dua aplikasi android.....	47
Gambar 4. 1 Hasil Pengukuran Tegangan Keluaran Menggunakan Multimeter..	49
Gambar 4. 2 Tampilan <i>Serial Monitor</i> Nilai Suhu, pH, dan <i>Turbidity</i> pada Air..	50
Gambar 4. 3 Hasil Pembacaan Sensor Suhu pada rentang 25-30°C.....	51
Gambar 4. 4 Hasil Pembacaan Sensor Suhu di bawah 25°C .....	51
Gambar 4. 5 Hasil Pembacaan Sensor pH di atas 7 .....	51
Gambar 4. 6 Hasil Pembacaan Sensor pH pada rentang 6-7.....	51
Gambar 4. 7 Hasil Pembacaan Sensor pH di bawah 6.....	51
Gambar 4. 8 Hasil Pembacaan Sensor <i>Turbidity</i> pada Air Normal .....	52

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





© Hak Cipta milik Politeknik Negeri Jakarta

Gambar 4. 9 Hasil Pembacaan Sensor *Turbidity* pada Air Keruh..... 52  
Gambar 4. 10 Tampilan Utama Pemantauan Kualitas Air pada Aplikasi..... 53  
Gambar 4. 11 Parameter QoS..... 55  
Gambar 4. 12 Perbandingan Firebase dengan Aplikasi Android..... 58  
Gambar 4. 13 Tampilan Notifikasi Suhu Air ..... 59  
Gambar 4. 14 Tampilan Notifikasi pH Air ..... 59  
Gambar 4. 12 Diagram Blok Pengujian Sistem ..... 59



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## DAFTAR TABEL

	<b>Halaman</b>
Tabel 2. 1 Spesifikasi Arduino Uno .....	6
Tabel 2. 2 Spesifikasi Sensor pH Modul SEN0161 .....	10
Tabel 2. 3 Spesifikasi Sensor <i>Turbidity</i> Modul SEN0189 .....	11
Tabel 2. 4 Spesifikasi Modul RTC DS3231.....	13
Tabel 2. 5 Spesifikasi Motor Servo MG996R.....	14
Tabel 2. 6 Rekomendasi ITU-T G.114 untuk <i>Delay</i> .....	16
Tabel 2. 7 Rekomendasi ITU-T G.114 untuk <i>Throughput</i> .....	16
Tabel 2. 8 Rekomendasi ITU-T G.114 untuk <i>Packet Loss</i> .....	17
Tabel 3. 1 Spesifikasi Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android .....	27
Tabel 3. 2 Penggunaan <i>Pin</i> Arduino Uno .....	30
Tabel 4. 1 Hasil Pengujian Tegangan Catu Daya Keluaran 9V .....	49
Tabel 4. 2 Hasil Pengujian Sensor Suhu .....	51
Tabel 4. 3 Hasil Pengujian Sensor pH .....	52
Tabel 4. 4 Hasil Pengujian Sensor <i>Turbidity</i> .....	52
Tabel 4. 5 Hasil Pengujian Modul RTC dan Motor Servo.....	53
Tabel 4. 6 Hasil Performansi Jaringan .....	56
Tabel 4. 7 Hasil Pengujian Notifikasi Aplikasi Android .....	58

POLITEKNIK  
NEGERI  
JAKARTA



## DAFTAR LAMPIRAN

Diagram Sistem Keseluruhan.....	L-1
Skematik Rangkaian Catu Daya.....	L-2
Skematik Rangkaian Sistem Mikrokontroler .....	L-3
Tampak Depan dan Tampak Belakang Casing .....	L-4
Tampak Bawah Casing .....	L-5
Datasheet Arduino Uno.....	L-6
Datasheet ESP8266 .....	L-7
Datasheet Sensor Suhu.....	L-8
Datasheet Sensor pH.....	L-9
Datasheet Sensor <i>Turbidity</i> .....	L-10
Datasheet Motor Servo.....	L-11
Datasheet Modul RTC.....	L-12
<i>Sketch</i> Pemrograman Arduino IDE (UNO).....	L-13
<i>Sketch</i> Pemrograman Arduino IDE (ESP) .....	L-14
<i>Code</i> Pemrograman Android.....	L-15
Dokumentasi .....	L-16

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

POLITEKNIK  
NEGERI  
JAKARTA





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Budidaya ikan hias merupakan salah satu sektor potensial yang dapat dikembangkan untuk meningkatkan kesejahteraan masyarakat di Indonesia. Ikan hias mempunyai kemampuan hidup yang beragam dengan keadaan yang dipengaruhi oleh kandungan dan kecerahan air. Kualitas air secara umum menunjukkan kondisi air dalam keadaan yang baik atau tidak, sama halnya dalam budidaya ikan koi di kolam air tawar. Budidaya ikan koi sebagai salah satu ikan hias yang sangat di gemari oleh pecinta ikan hias karena memiliki warna yang cerah dengan corak yang warna warni. Pemantauan kualitas air kolam merupakan salah satu hal yang dapat dilakukan untuk meningkatkan produksi perikanan.

Beberapa parameter seperti suhu, derajat keasaman (pH), oksigen dan tingkat kekeruhan pada kolam dimana merupakan habitat dan tempat hidup ikan, dapat berpengaruh terhadap pertumbuhan dan daya tahan ikan koi. Lingkungan kehidupan yang ideal untuk ikan koi rata-rata adalah untuk suhu 25-30°C, pH 6-7, dan tingkat kekeruhan. Selain pengaruh kualitas air pada kolam, pemberian pakan pada ikan koi juga dapat memengaruhi pertumbuhan dan daya tahan ikan budidaya. Pemberian pakan ikan yang kurang efisien akan berpengaruh terhadap penumpukan sisa pakan sehingga dapat menyebabkan penurunan kualitas kolam budidaya ikan dan secara tidak langsung dapat memengaruhi produktivitas kolam.

Seiring dengan kemajuan teknologi saat ini yang mampu mempermudah kegiatan manusia, maka dalam pemeliharaan dan budidaya ikan koi dapat dilakukan dengan pemantauan terhadap kualitas air kolam budidaya dan pengontrolan pemberian pakan ikan koi dengan sistem kendali jarak jauh. Proses pemberian pakan dengan sistem kendali jarak jauh dapat dilakukan menggunakan android maupun secara otomatis dengan memanfaatkan modul *Real Time Clock* (RTC), sehingga mempermudah dan menghemat waktu seseorang dalam proses pemberian pakan. Untuk merealisasikan perancangan sistem ini digunakan teknologi arduino uno dan nodemcu esp8266 yang terhubung secara *wireless* ke jaringan internet sebagai sistem pengiriman data pemantauan kualitas air seperti suhu, kadar pH, dan tingkat kekeruhan. Sistem mikrokontroler yang terhubung ke jaringan internet





## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dengan akses *wireless* menggunakan *access point* sebagai antarmuka pemantauan sistem dengan aplikasi berbasis android. Sistem ini dapat memudahkan masyarakat dalam melakukan pemantauan kualitas air dengan memberikan notifikasi terhadap keadaan tertentu dan pemberi pakan ikan koi secara otomatis pada saat jam yang sudah ditentukan tanpa perlu melakukannya secara manual.

Berdasarkan uraian tersebut, maka pada laporan ini akan dibahas mengenai “Rancang Bangun Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android”. Sistem ini dirancang dengan kendali jarak jauh menggunakan mikrokontroler yang tersambung ke *smartphone* melalui jaringan internet.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, maka permasalahan yang akan dibahas dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara merancang sistem mikrokontroler untuk melakukan pemantauan kualitas air dan pemberi pakan ikan koi?
2. Data apa saja yang ditampilkan pada android sebagai penerima?
3. Bagaimana melakukan pengujian performansi *Quality of Service* dalam penerimaan data dari mikrokontroler sampai ke android yang terintegrasi dengan jaringan internet?

### 1.3 Tujuan

Tujuan dari tugas akhir ini adalah:

1. Mampu merancang sistem pemantauan kualitas air pada akuarium ikan koi dengan perangkat arduino uno dan nodemcu esp8266.
2. Mampu membuat aplikasi android untuk pemantauan kualitas air dan pemberi pakan ikan koi.
3. Mampu melakukan pengujian terhadap performansi *Quality of Service* jaringan internet.

### 1.4 Luaran

Luaran yang hendak dicapai dalam tugas akhir ini adalah:

1. Menghasilkan sebuah Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android untuk membantu pemelihara ikan koi dalam

memantau keadaan air akuarium dan memberi makan ikan koi secara otomatis.

2. Menghasilkan sebuah buku laporan tugas akhir dari Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android.
3. Menghasilkan sebuah jurnal dari Sistem Pemantauan Kualitas Air dan Pemberi Pakan Ikan Koi Berbasis Android.



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## BAB V PENUTUP

### 5.1 Simpulan

Simpulan yang dapat diambil dari hasil pembuatan dan pengujian tugas akhir adalah sebagai berikut:

1. Sistem mikrokontroler yang dirancang untuk pemantauan kualitas air dan pemberi pakan ikan koi sudah berjalan dengan baik, dilihat dari keluaran nilai pembacaan sensor yang menyerupai hasil pembacaan alat ukur pembanding dan motor servo yang bergerak secara otomatis pada saat jam 08.00 WIB, 13.00 WIB. Dan 18.00 WIB.
2. Pada sistem mikrokontroler, setelah sensor membaca nilai, data tersebut akan dikirimkan ke Firebase melalui jaringan internet dengan perantara NodeMCU ESP8266 menggunakan API Key dan *Database* URL pada Fireabase *Realtime Database*. Pada aplikasi akan terlihat tiga variabel data, yaitu nilai suhu air, pH air, dan kekeruhan air. Pada saat sensor suhu membaca nilai suhu air 29,31°C pada aplikasi juga terlihat 29,31°C. Pada saat sensor pH membaca nilai pH air 6,87 pada aplikasi juga terlihat 6,87. Pada saat sensor *turbidity* membaca nilai kekeruhan air “normal” pada aplikasi juga terlihat kata “normal”. Aplikasi android juga dapat melakukan pemberi pakan secara manual dengan menekan *button* pada aplikasi android.
3. Pengujian performansi jaringan internet dengan cara menggunakan pada tempat yang berbeda, yaitu didalam rumah dan diluar rumah. Pada pengujian didalam rumah didapatkan *delay* sebesar 2,4 ms, *throughput* sebesar 3795kbps, dan *packet loss* sebesar 0% dan pada pengujian diluar rumah didapatkan *delay* sebesar 24 ms, *throughput* sebesar 249kbps, dan *packet loss* sebesar 0%. Berdasarkan pengujian QoS tersebut, pada saat didalam rumah sangat cocok untuk melakukan pemantauan kualitas air dan pemberi pakan ikan koi, karena memiliki *delay* yang kecil, *throughput* yang besar untuk kecepatan pengiriman data, dan tidak terdapat *packet loss* yang artinya tidak ada data yang hilang.

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## 5.2 Saran

Diharapkan dari hasil pembuatan tugas akhir ini dapat dikembangkan lebih lanjut terkait sensor yang digunakan, adanya pengembangan sistem menjadi lebih kompleks seperti pembuatan sistem menjadi skala yang lebih besar pada kolam peternakan budidaya ikan koi, dan sensor dikalibrasi agar lebih presisi atau lebih akurat saat dilakukan perbandingan dengan alat ukur pembanding. Apabila menggunakan android *studio*, usahakan tidak membuka aplikasi atau *software* lain agar tidak memberatkan aplikasi android *studio*.

### © Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





## DAFTAR PUSTAKA

- Alldatasheet. <https://html.alldatasheet.com/html-pdf/1131873/ETC2/MG996R/>. [19 Juli 2021].
- Andriani, Yuli, dkk. 2019. Peningkatan Kualitas Ikan Koi (*Cyprinus carpio*) di Kelompok PBC *Fish Farm* di Kecamatan Cisaat, Sukabumi. *Jurnal Ilmiah Pengabdian kepada Masyarakat*, vol 5, no 1, halaman 33-38.
- Barus, Eltra E., dkk. 2018. Otomatis Sistem Kontrol pH dan Informasi Suhu pada Akuarium Menggunakan Arduino Uno dan Raspberry Pi 3. *Jurnal Fisika*, vol 3, no 2, halaman 117-125.
- Developer Android. <https://developer.android.com/studio>. [19 Juli 2021].
- Fezari, Mohamed, dan Ali Al Dahoud. 2018. *Integrated Development Environment "IDE" for Arduino*. [https://www.researchgate.net/profile/Mohamed-Fezari-2/publication/328615543\\_Integrated\\_Development\\_Environment\\_IDE\\_For\\_Arduino/links/5bd8c6d24585150b2b9206df/Integrated-Development-Environment-IDE-For-Arduino.pdf](https://www.researchgate.net/profile/Mohamed-Fezari-2/publication/328615543_Integrated_Development_Environment_IDE_For_Arduino/links/5bd8c6d24585150b2b9206df/Integrated-Development-Environment-IDE-For-Arduino.pdf). [13 Juli 2021].
- Firestore *Cloud Messaging*. <https://firebase.google.com/docs/cloud-messaging>. [19 Juli 2021].
- Firestore *Realtime Database*. <https://firebase.google.com/docs/database>. [19 Juli 2021].
- Hutagaol, C.A. 2017. Mendeteksi Kekeruhan Air Menggunakan *Turbidity* Sensor Berbasis Arduino ATmega328 Berdasarkan Prinsip Hamburan Cahaya. <http://repositori.usu.ac.id/bitstream/handle/123456789/3294/142408017.pdf?sequence=1&isAllowed=y>. [13 Juli 2021].
- Ichwan, Muhammad, dkk. 2013. Pembangunan Prototipe Sistem Pengendalian Peralatan Listrik pada Platform Android. *Jurnal Informatika*, vol 4, no 1, halaman 13-25.
- Internet of Things Skills Development Training Program*. <http://www.it.kmitl.ac.th/~panwit/IoT003.pdf>. [19 Juli 2021].
- Kurniatuty, Surya Agung. 2019. Rancang Bangun Sistem Kontrol Pakan Ikan dan Kekeruhan Air yang Dilengkapi dengan Monitoring Kualitas Air Berbasis *Internet of Things* (IoT). <http://eprints.itn.ac.id/4323/8/Jurnal%20Skripsi.pdf>. [13 Juli 2021].
- Palit, Randi, V., dkk. 2015. Rancangan Sistem Informasi Keuangan Gereja Berbasis *Web* di Jemaat GMIM Bukit Moria Malalayang. *E-Journal Teknik Elektro dan Komputer*, vol 4, no 7, halaman 1-7.

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Rezarduno. 2019. Modul RTC DS3231. <https://arduino.rezaervani.com/2019/03/02/modul-rtc-ds3231/>. [19 Juli 2021].
- Sitohang, Ely P., dkk. 2018. Rancang Bangun Catu Daya DC Menggunakan Mikrokontroler ATmega8535. *Jurnal Teknik Elektro dan Komputer*, vol 7, no 2, halaman 135-142.
- Sokop, Steven Jendri, dkk. 2016. *Trainer* Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno. *E-Journal Teknik Elektro dan Komputer*, vol 5, no 3, halaman 13-23.
- Suryanto, Muhamad J.D. 2019. Rancang Bangun Alat Pencatat Biaya Pemakaian Energi Listrik pada Kamar Kos Menggunakan Modul *Global System For Mobile Communications*(GSM) 800L Berbasis Arduino Uno. *Jurnal Teknik Elektro*, vol 8, no 1, halaman 47-55.
- Wicaksono, Mochamad Fajar. 2019. *Aplikasi Arduino dan Sensor*. Bandung: Informatika Bandung.
- Yamato, dan Evyta Wismiana. 2014. Analisa Performansi Jaringan *Local Area Network* (LAN) IPTV. *Jurnal Teknologi*, vol 1, no 25, halaman 32-42.

POLITEKNIK  
NEGERI  
JAKARTA



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## DAFTAR RIWAYAT HIDUP PENULIS



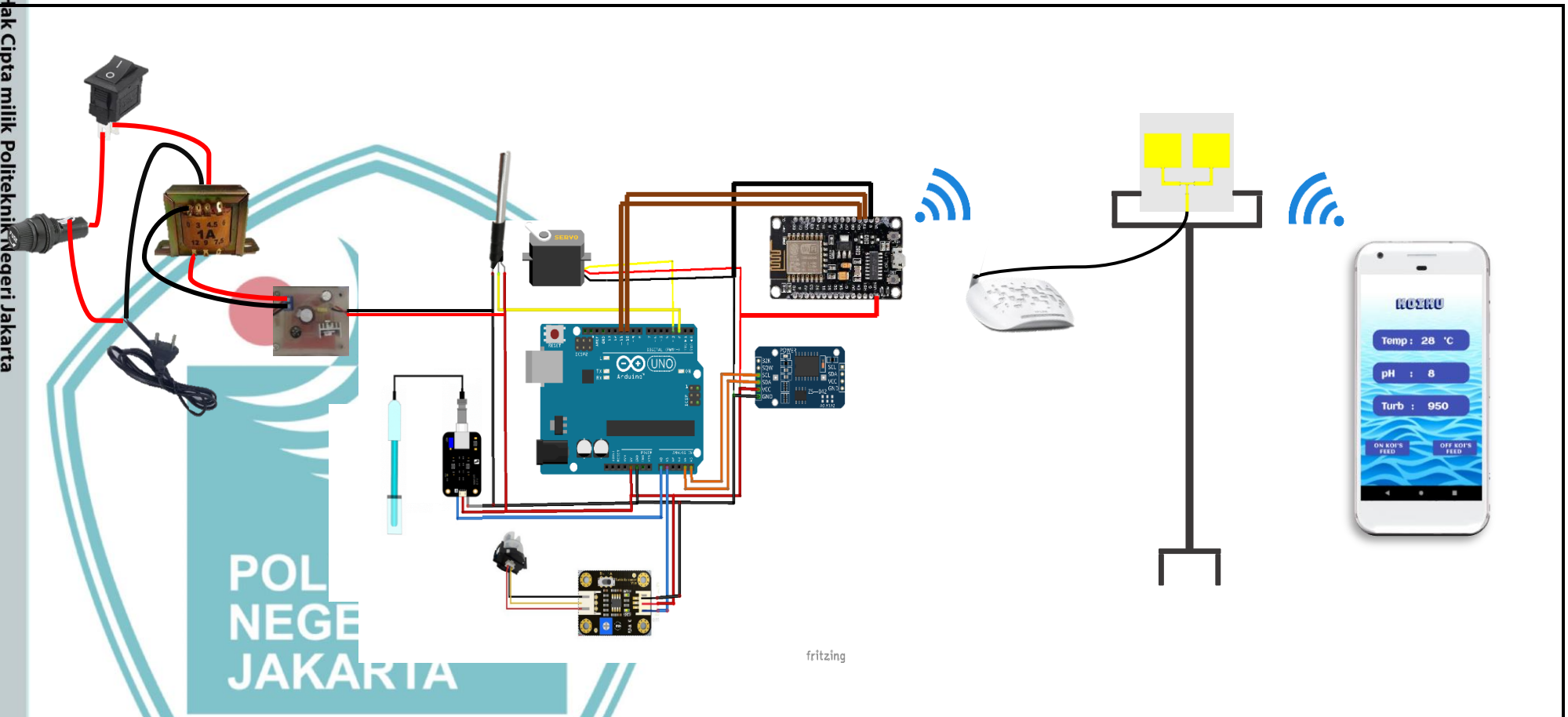
### **Afif Rio Syaputra**

Lahir di Bekasi, Jawa Barat pada tanggal 21 Oktober 2000. Merupakan anak ketiga dari tiga bersaudara. Memulai pendidikan dasar di SDN Kaliabang Tengah 7 dan lulus pada tahun 2012. Kemudian melanjutkan pendidikan di SMPN 19 Bekasi dan lulus pada tahun 2015. Kemudian melanjutkan pendidikan di SMAN 14 Bekasi dan lulus pada tahun 2018. Kemudian melanjutkan pendidikan D3 Program Studi Telekomunikasi, Jurusan Teknik Elektro, di Politeknik Negeri Jakarta, dan lulus pada tahun 2021.



POLITEKNIK  
NEGERI  
JAKARTA

- Hak Cipta :
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
    - a. Pengutipannya untuk kepentingan pendidikan, penelitian, pemertanian, karya ilmiah, pemertanian, laporan, pemertanian, kritik atau tinjauan suatu masalah.
    - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
  2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



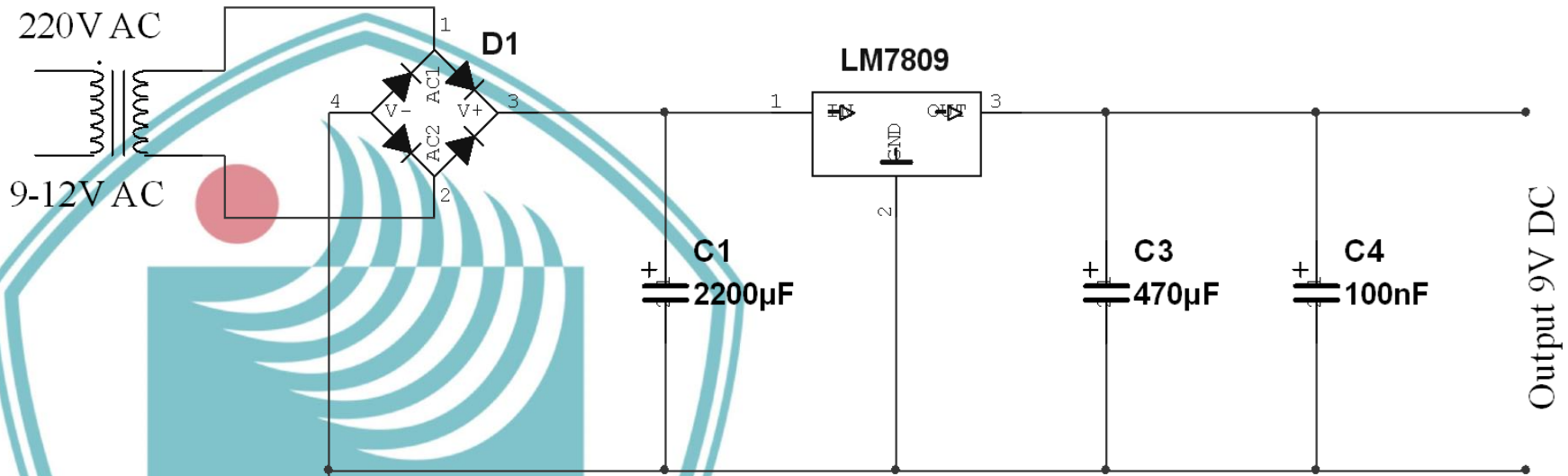
# DIAGRAM SISTEM KESELURUHAN

01

**PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA**

Digambar	Afif Rio Syaputra
Diperiksa	Benny Nixon, S.T., M.T.
Tanggal	25 Juli 2021





Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipannya untuk kepentingan akademik, penelitian, pemsan karya ilmiah, pemsantapuram, pemsari kritik atau tingdian suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

02

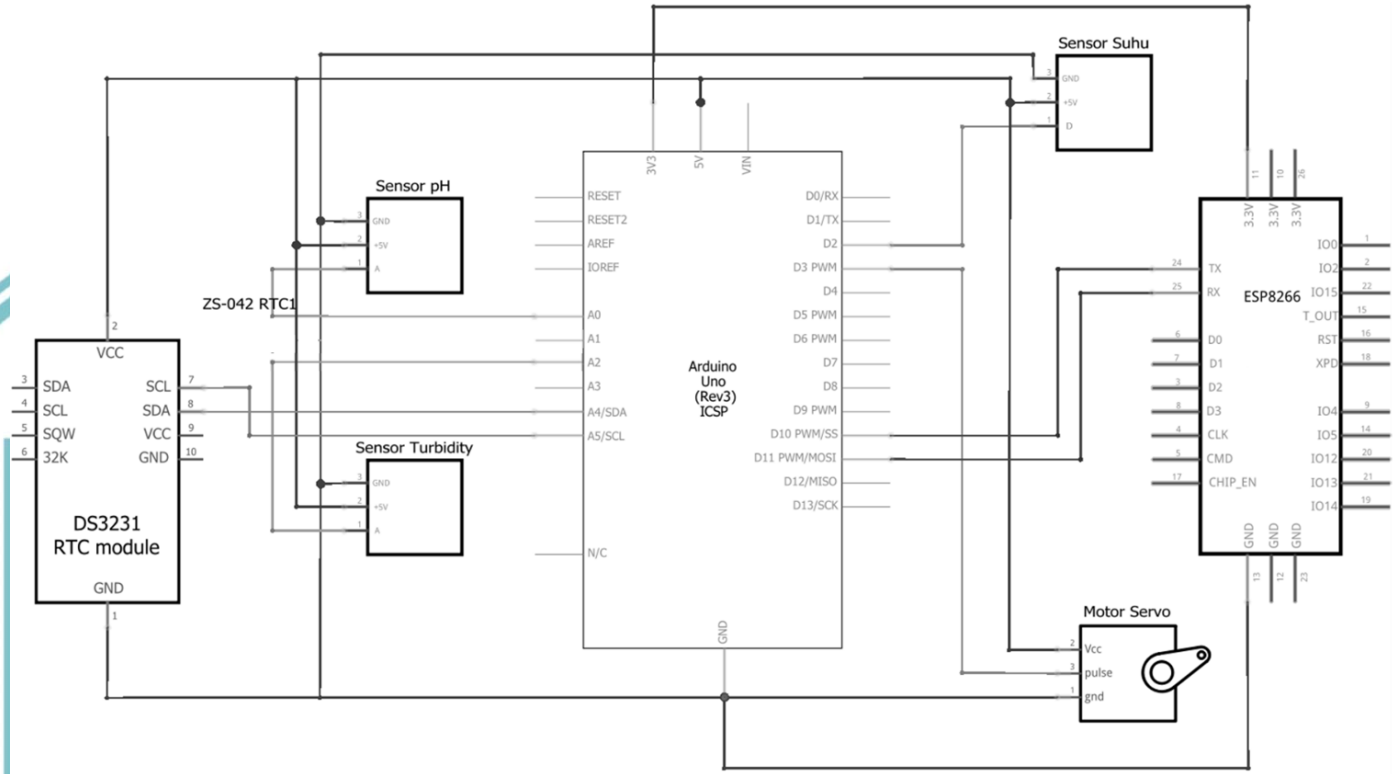
## SKEMATIK RANGKAIAN CATU DAYA



**PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK  
ELEKTRO – POLITEKNIK NEGERI JAKARTA**

Digambar	Afif Rio Syaputra
Diperiksa	Benny Nixon, S.T., M.T.
Tanggal	25 Juli 2021





Hak Cipta milik Politeknik Negeri Jakarta

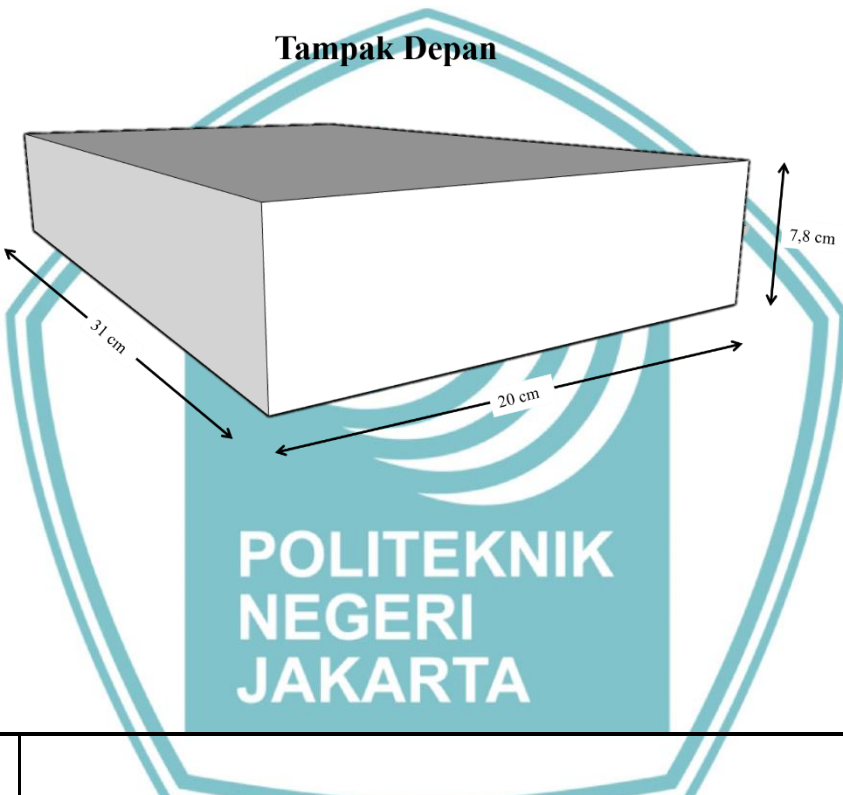
Hak Cipta :  
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :  
 a. Pengutipannya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan artikel atau tulisan untuk masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta  
 2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**03 SKEMATIK RANGKAIAN SISTEM MIKROKONTROLER**

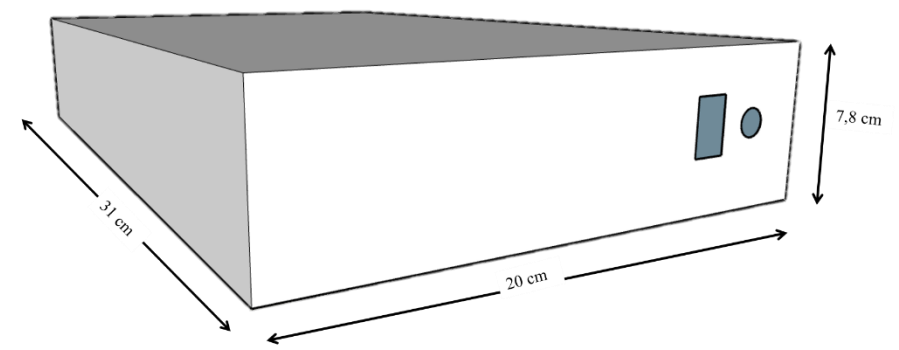
**PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA**

Digambar	Affif Rio Syaputra
Diperiksa	Benny Nixon, S.T., M.T.
Tanggal	25 Juli 2021





Tampak Depan



Tampak Belakang

Hak Cipta :  
 1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, pemrosesan karya ilmiah, dan penyempurnaan atau perkembangan ilmu pengetahuan dan teknologi.  
 b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta  
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

04

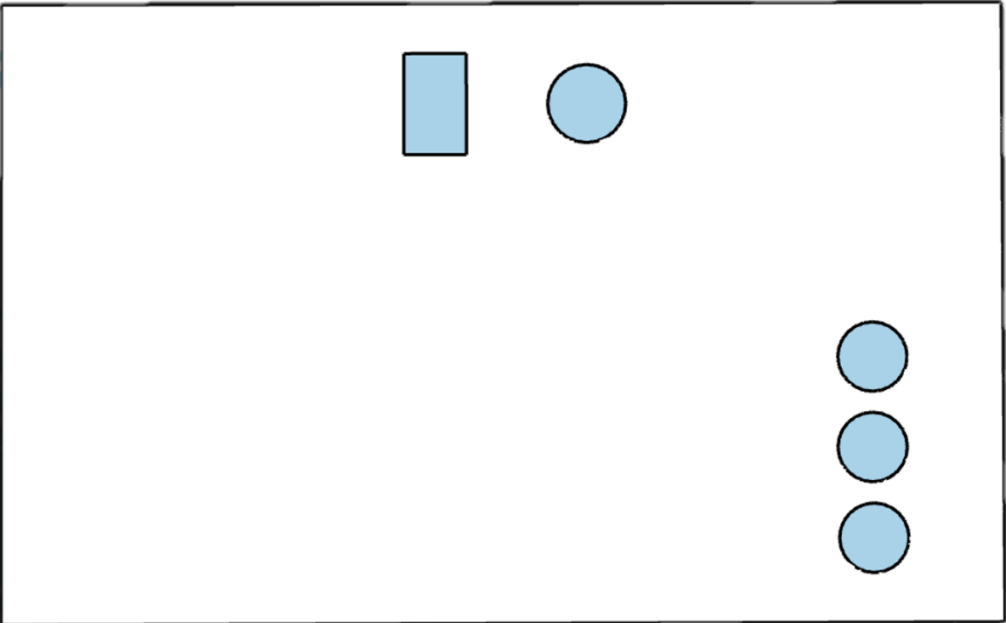
# TAMPAK DEPAN DAN TAMPAK BELAKANG CASING



**PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA**

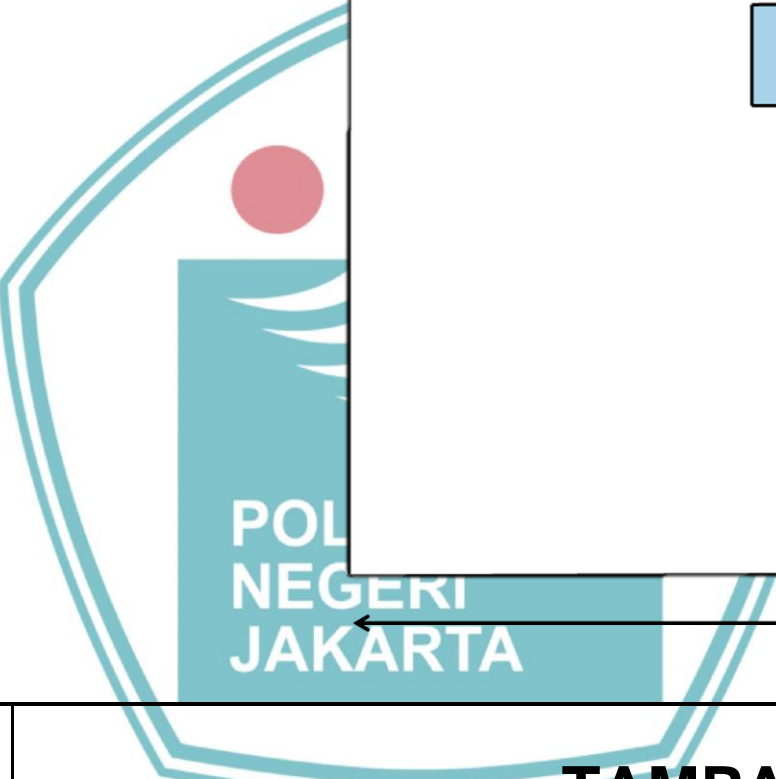
Digambar	Afif Rio Syaputra
Diperiksa	Benny Nixon, S.T., M.T.
Tanggal	25 Juli 2021

### Tampak Bawah



20 cm

31 cm



## TAMPAK BAWAH CASING

05

**PROGRAM STUDI TELEKOMUNIKASI JURUSAN TEKNIK ELEKTRO – POLITEKNIK NEGERI JAKARTA**

Digambar Afif Rio Syaputra

Diperiksa Benny Nixon, S.T., M.T.

Tanggal 25 Juli 2021



Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan harus menyebutkan sumber, pendahuluan, penutup dan kesimpulan karya ilmiah; penulisan laporan, penulisan kritik atau dijabarkan secara rasional.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta






© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Arduino UNO**

CE

**Product Overview**

The Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP<sup>®</sup> header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform, for a comparison with previous versions, see the [page of Arduino boards](#).

**Index**

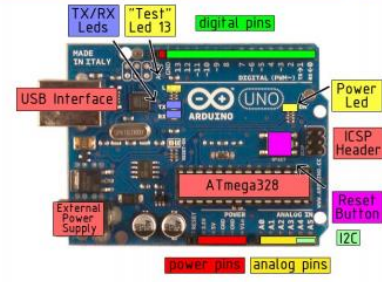
Technical Specifications	Page 2
How to use Arduino Programming Environment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Environmental Policies <small>half szgn of green via Impatto Zero®</small>	Page 7

**Technical Specification**

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current per 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



**Power**

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**. Ground pins.

**Memory**

The ATmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

**Input and Output**

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX)**. Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3**. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11**. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPi: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)**. These pins support SPi communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13**. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **FC: 4 (SDA) and 5 (SCL)**. Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF**. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [pinmap between Arduino pins and ATmega328 pins](#).

**Communication**

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART (TTL) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .inf file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I<sup>2</sup>C (TWI) and SPi communication. The Arduino software includes a Wire library to simplify use of the I<sup>2</sup>C bus; see the [documentation](#) for details. To use the SPi communication, please see the ATmega328 datasheet.

**Programming**

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorial](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmega8U2 software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ICSP header with an external programmer (overwriting the DFU bootloader).



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

<p>ESP8266 802.11bgn Smart Device</p>  <h3>1 Introduction</h3> <p>Espressif Systems' Smart Connectivity Platform (ESCP) of high performance wireless SOCs, for mobile platform designers, provides unsurpassed ability to embed Wi-Fi capabilities within other systems, at the lowest cost with the greatest functionality.</p>  <p>Figure 1-1: ESP8266 Block Diagram</p> <p>4   Page      Espressif Systems      Oct 12, 2013</p>	<p>ESP8266 802.11bgn Smart Device</p>  <h3>2 Technology Overview</h3> <p>ESP8266 offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.</p> <p>When ESP8266 hosts the application, and when it is the only application processor in the device, it is able to boot up directly from an external flash. It has integrated cache to improve the performance of the system in such applications, and to minimize the memory requirements.</p> <p>Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller-based design with simple connectivity through UART interface or the CPU AHB bridge interface.</p> <p>ESP8266 on-board processing and storage capabilities allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. With its high degree of on-chip integration, which includes the antenna switch balun, power management controllers, it requires minimal external circuitry, and the entire solution, including base-band module, is designed to occupy minimal PCB area.</p> <p>Sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.</p> <p>5   Page      Espressif Systems      Oct 12, 2013</p>
<p>ESP8266 802.11bgn Smart Device</p>  <h3>3 Features</h3> <ul style="list-style-type: none"> <li>• 802.11 b/g/n protocol</li> <li>• Wi-Fi Direct (P2P), soft-AP</li> <li>• Integrated TCP/IP protocol stack</li> <li>• Integrated TR switch, balun, LNA, power amplifier and matching network</li> <li>• Integrated PLL, regulators, and power management units</li> <li>• +19.5dBm output power in 802.11b mode</li> <li>• Integrated temperature sensor</li> <li>• Supports antenna diversity</li> <li>• Power down leakage current of &lt; 10uA</li> <li>• Integrated low power 32-bit CPU could be used as application processor</li> <li>• SDIO 2.0, SPI, UART</li> <li>• STBC, 1x1 MIMO, 2x1 MIMO</li> <li>• A-MPDU &amp; A-MSDU aggregation &amp; 0.4us guard interval</li> <li>• Wake up and transmit packets in &lt; 2ms</li> <li>• Standby power consumption of &lt; 10mW (DTIMS)</li> </ul> <p>6   Page      Espressif Systems      Oct 12, 2013</p>	<p>ESP8266 802.11bgn Smart Device</p>  <h3>6 ESP8266 Applications</h3> <ul style="list-style-type: none"> <li>• Smart power plugs</li> <li>• Home automation</li> <li>• Mesh network</li> <li>• Industrial wireless control</li> <li>• Baby monitors</li> <li>• IP Cameras</li> <li>• Sensor networks</li> <li>• Wearable electronics</li> <li>• Wi-Fi location-aware devices</li> <li>• Security ID tags</li> <li>• Wi-Fi position system beacons</li> </ul> <p>9   Page      Espressif Systems      Oct 12, 2013</p>



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**PRELIMINARY**  
**DS18B20**  
**Programmable Resolution**  
**1-Wire® Digital Thermometer**

**DALLAS SEMICONDUCTOR**

[www.dallassemi.com](http://www.dallassemi.com)

**FEATURES**

- Unique 1-Wire interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Zero standby power required
- Measures temperatures from -55°C to +125°C. Fahrenheit equivalent is -67°F to +257°F
- ±0.5°C accuracy from -10°C to +85°C
- Thermometer resolution is programmable from 9 to 12 bits
- Converts 12-bit temperature to digital word in 750 ms (max.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

**DESCRIPTION**

The DS18B20 Digital Thermometer provides 9 to 12-bit (configurable) temperature readings which indicate the temperature of the device.

Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

**DETAILED PIN DESCRIPTION** Table 1

PIN #	8-PIN SOIC	PIN TO92	SYMBOL	DESCRIPTION
5	1		GND	Ground.
4	2		DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See "Parasite Power" section.)
3	3		V <sub>DD</sub>	Optional V <sub>DD</sub> pin. See "Parasite Power" section for details of connection. V <sub>DD</sub> must be grounded for operation in parasite power mode.

DS18B20Z (8-pin SOIC): All pins not specified in this table are not to be connected.

**OVERVIEW**

The block diagram of Figure 1 shows the major components of the DS18B20. The DS18B20 has four main data components: 1) 64-bit lasered ROM, 2) temperature sensor, 3) nonvolatile temperature alarm triggers TH and TL, and 4) a configuration register. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS18B20 may also be powered from an external 3 volt - 5.5 volt supply.

Communication to the DS18B20 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out a specific device if many are present on the 1-Wire line as well as indicate to the bus master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS18B20 to perform a temperature measurement. The result of this measurement will be placed in the DS18B20's scratchpad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of 1 byte EEPROM each. If the alarm search command is not applied to the DS18B20, these registers may be used as general purpose user memory. The scratchpad also contains a configuration byte to set the desired resolution of the temperature to digital conversion. Writing TH, TL, and the configuration byte is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

**PIN ASSIGNMENT**

**BOTTOM VIEW**

**PIN DESCRIPTION**

GND - Ground  
DQ - Data In/Out  
V<sub>DD</sub> - Power Supply Voltage  
NC - No Connect

1 of 27 059400

2 of 27

**DS18B20 BLOCK DIAGRAM** Figure 1

**PARASITE POWER**

The block diagram (Figure 1) shows the parasite-powered circuitry. This circuitry "steals" power whenever the DQ or V<sub>DD</sub> pins are high. DQ will provide sufficient power as long as the specified timing and voltage requirements are met (see the section titled "1-Wire Bus System"). The advantages of parasite power are twofold: 1) by parasitizing off this pin, no local power source is needed for remote sensing of temperature, and 2) the ROM may be read in absence of normal power.

In order for the DS18B20 to be able to perform accurate temperature conversions, sufficient power must be provided over the DQ line when a temperature conversion is taking place. Since the operating current of the DS18B20 is up to 1.5 mA, the DQ line will not have sufficient drive due to the 5k pullup resistor. This problem is particularly acute if several DS18B20s are on the same DQ and attempting to convert simultaneously.

There are two ways to assure that the DS18B20 has sufficient supply current during its active conversion cycle. The first is to provide a strong pullup on the DQ line whenever temperature conversions or copies to the E<sup>2</sup> memory are taking place. This may be accomplished by using a MOSFET to pull the DQ line directly to the power supply as shown in Figure 2. The DQ line must be switched over to the strong pull-up within 10 ns maximum after issuing any protocol that involves copying to the E<sup>2</sup> memory or initiates temperature conversions. When using the parasite power mode, the V<sub>DD</sub> pin must be tied to ground.

Another method of supplying current to the DS18B20 is through the use of an external power supply tied to the V<sub>DD</sub> pin, as shown in Figure 3. The advantage to this is that the strong pullup is not required on the DQ line, and the bus master need not be tied up holding that line high during temperature conversions. This allows other data traffic on the 1-Wire bus during the conversion time. In addition, any number of DS18B20s may be placed on the 1-Wire bus, and if they all use external power, they may all simultaneously perform temperature conversions by issuing the Skip ROM command and then issuing the Convert T command. Note that as long as the external power supply is active, the GND pin may not be floating.

The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS18B20 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that V<sub>DD</sub> be applied to the DS18B20.

3 of 27

For situations where the bus master does not know whether the DS18B20s on the bus are parasite powered or supplied with external V<sub>DD</sub>, a provision is made in the DS18B20 to signal the power supply scheme used. The bus master can determine if any DS18B20s are on the bus which require the strong pullup by sending a Skip ROM protocol, then issuing the read power supply command. After this command is issued, the master then issues read time slots. The DS18B20 will send back "0" on the 1-Wire bus if it is parasite powered; it will send back a "1" if it is powered from the V<sub>DD</sub> pin. If the master receives a "0," it knows that it must supply the strong pullup on the DQ line during temperature conversions. See "Memory Command Functions" section for more detail on this command protocol.

**STRONG PULLUP FOR SUPPLYING DS18B20 DURING TEMPERATURE CONVERSION** Figure 2

**USING V<sub>DD</sub> TO SUPPLY TEMPERATURE CONVERSION CURRENT** Figure 3


4 of 27






### Hak Cipta :


1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



PH meter(SKU: SEN0161)



Analog pH Meter Kit SKU: SEN0161



Analog pH Meter Kit SKU: SEN0169

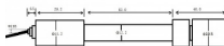
Contents

- 1 Introduction
- 2 Specification
- 3 Precautions
- 4 pH Electrode Characteristics
- 5 Usage
  - 5.1 Connecting Diagram
  - 5.2 Method 1. Software Calibration
  - 5.3 Method 2. Hardware Calibration through potentiometer
- 6 FAQ

**Introduction**

Need to measure water quality and other parameters but haven't got any low cost pH meter? Find it difficult to use with Arduino? Here comes an analog pH meter, specially designed for Arduino controllers and has built-in simple, convenient and practical connection and features. It has an LED which works as the Power Indicator, a BNC connector and PH2.0 sensor interface. You can just connect the pH sensor with BNC connector, and plug the PH2.0 interface into any analog input on Arduino controller to read pH value easily.

**Specification**



SEN0161 dimension

- Module Power: 5.00V
- Circuit Board Size: 43mm\*32mm
- pH Measuring Range: 0-14
- Measuring Temperature: 0-60 °C
- Accuracy: ± 0.1pH (25 °C)
- Response Time: ≤ 1min
- pH Sensor with BNC Connector
- PH2.0 Interface ( 3 foot patch )
- Gain Adjustment Potentiometer
- Power Indicator LED

**Precautions**

- Before and after use of the pH electrode every time, you need to use (pure)water to clean it.
- The electrode plug should be kept clean and dry in case of short circuit.
- **Preservation:** Electrode reference preservation solution is the 3N KCL solution.
- Measurement should be avoided staggered pollution between solutions, so as not to affect the accuracy of measurement.
- Electrode blub or sand core is defiled which will make PTS decline, slow response. So, it should be based on the characteristics of the pollutant, adapted to the cleaning solution, the electrode performance recovery.

- Electrode when in use, the ceramic sand core and liquid outlet rubber ring should be removed, in order to make salt bridge solution to maintain a certain velocity.

**NOTE:** Differences between the probes, SEN0161 and SEN0169

Their usages/ specifications are almost the same. The differences locates at

**Long-firing Operation:** SEN0169 supports, while SEN0161 NOT, i.e. you can not immerse SEN0161 in water for Continuous Testing.

**Life Span:** In 25 °C, pure water, do Continuous Testing with them both, SEN0169 can work two years, while SEN0161 can only last for 6 months. And just for reference, if put them in turbid, strongly acid and alkali solution, 25°C, the life span would drop to one year (SEN0169), 1 month(or shorter, SEN0161). Temperature, pH, turbidity of the water effect the probe life span a lot.

**Waterproof:** You can immerse the whole probe SEN0169 into the water, while you can only immerse the front part of the probe SEN0161, the electrode glass bulb, into water, the rear part, from the white shell to the cable, MUST NOT be under water.

**Strongly Acid and Alkali:** SEN0169 are preferred for strongly acid and alkali test. And if your testing range is usually within pH5-8, then SEN0161 is capable for that.

**pH Electrode Characteristics**

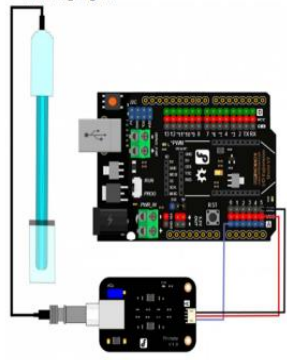
The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 °C):

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

**NOTE:** It is normal that if your reading is much different with the table since you are not reading from the electrode directly but from the voltage adapter, it has converted the original voltage (-5V ~ +5V) to Arduino compatible voltage, i.e. 0 ~ 5V. See the discussion on Forum.

**Usage**

**Connecting Diagram**




**NOTE:**




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Turbidity sensor SKU: SEN0189



**Contents**

- 1 Introduction
- 2 Specification
- 3 Connection Diagram
- 4 Examples

**Introduction**

The turbidity sensor detects water quality by measuring the levels of turbidity. It uses light to detect suspended particles in water by measuring the light transmittance and scattering rate, which changes with the amount of total suspended solids (TSS) in water. As the TSS increases, the liquid turbidity level increases.

Turbidity sensors are used to measure water quality in rivers and streams, wastewater and effluent measurements, control instrumentation for settling ponds, sediment transport research and laboratory measurements.

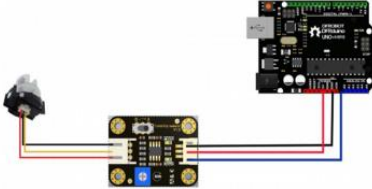
This sensor provides analog and digital signal output modes. The threshold is adjustable when in digital signal mode. You can select the mode according to your MCU.

**Note:** The top of probe is not waterproof.

**Specification**

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
  - Analog output: 0-4.5V
  - Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5°C-90°C
- Storage Temperature: -10°C-90°C
- Weight: 30g
- Adapter Dimensions: 38mm\*28mm\*10mm/1.5inches\*1.1inches\*0.4inches

**Connection Diagram**



**Interface Description:**

1. "D/A" Output Signal Switch
1. "A": Analog Signal Output, the output value will decrease when in liquids with a high turbidity
2. "D": Digital Signal Output, high and low levels, which can be adjusted by the threshold potentiometer
2. Threshold Potentiometer: you can change the trigger condition by adjusting the threshold potentiometer in digital signal mode.

**Examples**

Here are two examples:  
 Example 1 uses Analog output mode  
 Example 2 uses Digital output mode

**Example 1**

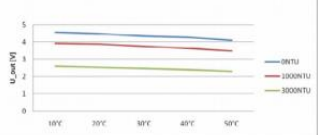
```
void setup() {
  Serial.begin(9600); //Baud rate: 9600
}
void loop() {
  int sensorValue = analogRead(A0); // read the input on analog pin 0;
  float voltage = sensorValue * (5.0 / 1024.0); // Convert the analog read
  ing (which goes from 0 - 1023) to a voltage (0 - 5V);
  Serial.println(voltage); // print out the value you read;
  delay(500);
}
```

**Example 2**

```
int ledPin = 13; // Connect an LED on pin 13, or use the on
board one
int sensor_in = 2; // Connect turbidity sensor to Digital
Pin 2
```

```
void setup() {
  pinMode(ledPin, OUTPUT); // Set ledPin to output mode
  pinMode(sensor_in, INPUT); //Set the turbidity sensor pin to input
  mode
}
void loop() {
  if(digitalRead(sensor_in)==LOW) { //read sensor signal
    digitalWrite(ledPin, HIGH); // If sensor is LOW, then turn on
    the led
  } else{
    digitalWrite(ledPin, LOW); // If sensor is HIGH, then turn off
    the led
  }
}
```

This is a reference chart for the mapping from the output voltage to the NTU according to different temperature. e.g. If you leave the sensor in the pure water, that is NTU < 0.5, it should output "4.1±0.3V" when temperature is 10-50°C.

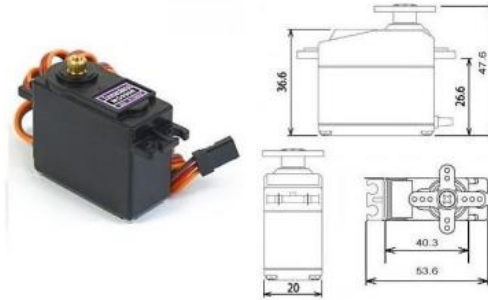


**characteristic curve "Voltage —Temperature**

**Note:** In the diagram, the unit measuring turbidity is shown as NTU, also it is known as JTU (Jackson Turbidity Unit), 1JTU = 1NTU = 1 mg/L. Refer to Turbidity wikipedia

Powered By DFRobot © 2008-2017

**MG996R High Torque Metal Gear Dual Ball Bearing Servo**



This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwith and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

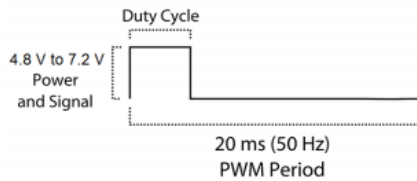
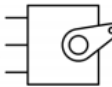
This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

**Specifications**

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)

- Operating voltage: 4.8 V a 7.2 V
- Running Current 500 mA – 900 mA (6V)
- Stall Current 2.5 A (6V)
- Dead band width: 5 μs
- Stable and shock proof double ball bearing design
- Temperature range: 0 °C – 55 °C

PWM=Orange (⏏)  
 Vcc = Red (+)  
 Ground=Brown (-)



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**DS3231** Extremely Accurate I2C-Integrated RTC/TCXO/Crystal

**General Description**

The DS3231 is a low-cost, extremely accurate I<sup>2</sup>C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-ml SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I<sup>2</sup>C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V<sub>CC</sub> to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a μP reset.

**Benefits and Features**

- Highly Accurate RTC Completely Manages All Timekeeping Functions
  - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
  - Accuracy ±2ppm from 0°C to 40°C
  - Accuracy ±3.5ppm from -40°C to +85°C
  - Digital Temp Sensor Output: ±3°C Accuracy
  - Register for Aging Trim
  - RST Output/Pushbutton Reset Debounce Input
  - Two Time-of-Day Alarms
  - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
  - Fast (400kHz) I<sup>2</sup>C Interface
- Battery-Backup Input for Continuous Timekeeping
  - Low Power Operation Extends Battery-Backup Run Time
  - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (+40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

**Applications**

- Utility Power Meters
- Servers
- Telematics
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

**Typical Operating Circuit**

Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

**maxim**  
integrated.

19-0770 Rev 10 2010

**DS3231** Extremely Accurate I2C-Integrated RTC/TCXO/Crystal

**Absolute Maximum Ratings**

Voltage Range on Any Pin Relative to Ground: -0.3V to +6.0V  
 Junction-to-Ambient Thermal Resistance (R<sub>JA</sub>) (Note 1): 107°C/W  
 Junction-to-Case Thermal Resistance (R<sub>JC</sub>) (Note 1): 22°C/W  
 Operating Temperature Range: 0°C to +70°C (DS3231S), -40°C to +85°C (DS3231N)

Junction Temperature: +125°C  
 Storage Temperature Range: -40°C to +85°C  
 Lead Temperature (soldering, 10s): +260°C  
 Soldering Temperature (reflow, 2 times max): +260°C (see the Handling, PCB Layout, and Assembly section)

**Notes:**  
 Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to [www.maximintegrated.com/thermal-subtotal](http://www.maximintegrated.com/thermal-subtotal).  
 (T<sub>A</sub> = T<sub>MAX</sub> to T<sub>MIN</sub>, unless otherwise noted.) (Notes 2, 3)

Devices beyond those listed under "Absolute Maximum Rating" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or at other conditions beyond those indicated in the operational sections of the specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Recommended Operating Conditions**  
 (T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>CC</sub>		2.3	3.3	5.5	V
	V <sub>BAT</sub>		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V <sub>IH</sub>		0.7 × V <sub>CC</sub>			V
			V <sub>CC</sub>		0.3	V
Logic 0 Input SDA, SCL	V <sub>IL</sub>		-0.3		0.3 × V <sub>CC</sub>	V

**Electrical Characteristics**  
 V<sub>CC</sub> = 2.3V to 5.5V, V<sub>CC</sub> = Active Supply (see Table 1), T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Typical values are at V<sub>CC</sub> = 3.3V, V<sub>BAT</sub> = 3.0V, and T<sub>A</sub> = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I <sub>CCA</sub>	(Notes 4, 5) V <sub>CC</sub> = 3.63V V <sub>CC</sub> = 5.5V		200		μA
Standby Supply Current	I <sub>CCS</sub>	PC bus inactive, 32kHz output on, SQW output of (Note 5) V <sub>CC</sub> = 3.63V V <sub>CC</sub> = 5.5V		110		μA
Temperature Conversion Current	I <sub>CCSQW</sub>	PC bus inactive, 32kHz output on, SQW output of V <sub>CC</sub> = 3.63V V <sub>CC</sub> = 5.5V		575		μA
Power-Fail Voltage	V <sub>WF</sub>		2.45	2.675	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	I <sub>OL</sub>	I <sub>L</sub> = 3mA		0.4		V
Logic 0 Output, RST	I <sub>OL</sub>	I <sub>L</sub> = 1mA		0.4		V
Output Leakage Current 32kHz, INT/SQW, SDA	I <sub>LO</sub>	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I <sub>LI</sub>			-1	+1	μA
RST Pin I/O Leakage	I <sub>LI</sub>	RST high impedance (Note 6)	-200		+10	μA
V <sub>BAT</sub> Leakage Current (V <sub>CC</sub> Active)	I <sub>BATLKG</sub>			25	100	nA

www.maximintegrated.com Maxim Integrated | 2

**DS3231** Extremely Accurate I2C-Integrated RTC/TCXO/Crystal

**Electrical Characteristics (continued)**  
 V<sub>CC</sub> = 2.3V to 5.5V, V<sub>CC</sub> = Active Supply (see Table 1), T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Typical values are at V<sub>CC</sub> = 3.3V, V<sub>BAT</sub> = 3.0V, and T<sub>A</sub> = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f <sub>OUT</sub>	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V		32 768		kHz
Frequency Stability vs. Temperature (Commercial)	ΔF <sub>OUT</sub>	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V, aging offset = 00h 0°C to +40°C		±2		ppm
		-40°C to +70°C		±3.5		ppm
Frequency Stability vs. Temperature (Industrial)	ΔF <sub>OUT</sub>	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V, aging offset = 00h 0°C to +40°C		±2		ppm
		-40°C to +85°C		±3.5		ppm
Frequency Stability vs. Voltage	ΔFV			1		ppm/V
		+40°C		0.7		ppm/V
		+25°C		0.1		ppm/V
		+70°C		0.4		ppm/V
		+85°C		0.8		ppm/V
Trim Register Frequency Sensitivity per LSB	ΔMSB	Specified at:				ppm
Temperature Accuracy	Temp	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V		-3		°C
Crystal Aging	ΔF <sub>TC</sub>	After reflow, not production tested	First year	±1.0		ppm
			0-10 years	±5.0		ppm

**Electrical Characteristics**  
 V<sub>CC</sub> = 0V, V<sub>BAT</sub> = 3.3V to 5.5V, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Battery Current	I <sub>BATA</sub>	EOSC = 0, BBSQW = 0, SCL = 400kHz (Note 5) V <sub>BAT</sub> = 3.63V V <sub>BAT</sub> = 5.5V		70	150	μA
Timekeeping Battery Current	I <sub>BATT</sub>	EOSC = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V <sub>BAT</sub> (Note 5) V <sub>BAT</sub> = 5.5V		0.84	3.0	μA
Temperature Conversion Current	I <sub>BATTC</sub>	EOSC = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V <sub>BAT</sub> V <sub>BAT</sub> = 5.5V		575		μA
Data-Retention Current	I <sub>BATDRS</sub>	EOSC = 1, SCL = SDA = 0V, +25°C		100		nA

www.maximintegrated.com Maxim Integrated | 3

**DS3231** Extremely Accurate I2C-Integrated RTC/TCXO/Crystal

**AC Electrical Characteristics**  
 V<sub>CC</sub> = V<sub>CC(MIN)</sub> to V<sub>CC(MAX)</sub> or V<sub>BAT</sub> = V<sub>BAT(MIN)</sub> to V<sub>BAT(MAX)</sub>, V<sub>BAT</sub> > V<sub>CC</sub>, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f <sub>SCL</sub>	Fast mode Standard mode	100	400		kHz
Bus Free Time Between STOP and START Conditions	t <sub>BUF</sub>	Fast mode Standard mode	1.3			μs
Hold Time (Repeated) START Condition (Note 7)	t <sub>HOLDSTA</sub>	Fast mode Standard mode	0.6			μs
Low Period of SCL Clock	t <sub>LOW</sub>	Fast mode Standard mode	1.3			μs
		Fast mode Standard mode	4.7			μs
High Period of SCL Clock	t <sub>HIGH</sub>	Fast mode Standard mode	4.0			μs
		Fast mode Standard mode	9			μs
Data Hold Time (Notes 8, 9)	t <sub>HOLDATA</sub>	Fast mode Standard mode	0	0.9		μs
		Fast mode Standard mode	100			ns
Data Setup Time (Note 10)	t <sub>SDATA</sub>	Fast mode Standard mode	250			ns
START Setup Time	t <sub>SDSTA</sub>	Fast mode Standard mode	0.6			μs
Rise Time of Both SDA and SCL Signals (Note 11)	t <sub>R</sub>	Fast mode Standard mode	20 + 0.1C <sub>B</sub>		300	ns
Fall Time of Both SDA and SCL Signals (Note 11)	t <sub>F</sub>	Fast mode Standard mode	20 + 0.1C <sub>B</sub>		300	ns
Setup Time for STOP Condition	t <sub>SU310</sub>	Fast mode Standard mode	0.6			μs
Capacitive Load for Each Bus Line	C <sub>B</sub>	(Note 11)		400		pF
Capacitance for SDA, SCL	C <sub>SD</sub>			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t <sub>SP</sub>			30		ns
Propagation Delay	t <sub>PDR</sub>			250		ns
Reset Active Time	t <sub>RST</sub>			250		ms
Oscillator Stop Flag (OSF) Delay	t <sub>OSF</sub>	(Note 12)		100		ms
Temperature Conversion Time	t <sub>CCONV</sub>			125	200	ms

**Power-Switch Characteristics**  
 (T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V <sub>CC</sub> Fall Time: V <sub>FH(MIN)</sub> to V <sub>FL(MIN)</sub>	t <sub>VCCF</sub>			300		μs
V <sub>CC</sub> Rise Time: V <sub>FL(MIN)</sub> to V <sub>FH(MIN)</sub>	t <sub>VCCR</sub>			0		μs
Recovery at Power-Up	t <sub>REC</sub>	(Note 13)		250	300	ms

www.maximintegrated.com Maxim Integrated | 4

## L-13 Sketch Pemrograman Arduino IDE (UNO)



### © Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#include <SoftwareSerial.h>
SoftwareSerial linkSerial(10, 11);

#include <ArduinoJson.h>

#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature ds18b20(&oneWire);

#define pHPin A0 //pH meter Analog output to Arduino
Analog Input 0
#define Offset 0.00 //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40 //times of collection
int pHArray[ArrayLenth]; //store the average value of the sensor
feedback
int pHArrayIndex = 0;

#include "RTClib.h"
RTC_DS3231 rtc;
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};

#include <Servo.h>
Servo myservo; //create servo object to control a servo

String turbClass = "";

void setup() {
  Serial.begin(9600);
  linkSerial.begin(115200);
  ds18b20.begin(); //start up the library
  pinMode(LED, OUTPUT);
  turbClass = "normal";
  myservo.attach(3);
  myservo.write(0);
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
    Serial.flush();
    abort();
  }
  if (rtc.lostPower()) {
    Serial.println("RTC lost power, let's set the time!");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}

void loop() {
  static uint8_t flag = 0;
  static uint32_t millisSensor, timeSensor = 500;

  if (linkSerial.available() > 0 && flag == 0) {
    String incomingString = linkSerial.readStringUntil('\n');
    //Serial.print("I received: ");
```

```

//Serial.println(incomingString);
if (incomingString == "*b0#") {
  myservo.write(0);
  Serial.println("SERVO TERTUTUP");
  Serial.println();
  delay(10);
} else if (incomingString == "*b1#") {
  myservo.write(90);
  Serial.println("SERVO TERBUKA");
  Serial.println();
  delay(10);
}
}

if (millis() - millisSensor >= 500) {
  //deteksi suhu
  ds18b20.requestTemperatures();
  float suhu = ds18b20.getTempCByIndex(0);
  Serial.print("Temperature: ");
  Serial.print(suhu);
  Serial.print((char)176);//shows degrees character
  Serial.println("C");
  Serial.println();

  //deteksi kekeruhan
  int turbVal = analogRead(A2);//read the input on analog pin 0:
  float turbVolt = turbVal * (5.0 / 1024.0); //convert the
  analog reading (which goes from 0 - 1023) to a turbVolt (0 - 5V):
  Serial.print("Turbidity Value: ");
  Serial.print(turbVal);
  Serial.print(" || Turbidity Voltage: ");
  Serial.println(turbVolt); //print out the value you read:
  Serial.println();
  if (turbVolt >= 3.5) {
    turbClass = "normal";
  } else if (turbVolt < 3.2) {
    turbClass = "keruh";
  }
}

//deteksi ph
static unsigned long samplingTime = millis();
static unsigned long printTime = millis();
static float pHValue, pHVolt;
if (millis() - samplingTime > samplingInterval)
{
  pHArray[pHArrayIndex++] = analogRead(phPin);
  if (pHArrayIndex == ArrayLenth)pHArrayIndex = 0;
  pHVolt = avergarray(pHArray, ArrayLenth) * 5.0 / 1024;
  pHValue = 3.5 * pHVolt + Offset;
  samplingTime = millis();
}
if (millis() - printTime > printInterval) //every 800
milliseconds, print a numerical, convert the state of the LED
indicator
{
  Serial.print("pH voltage: ");
  Serial.print(phVolt, 2);
  Serial.print("    pH value: ");
  Serial.println(pHValue, 2);
}

```

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



```

digitalWrite(LED, digitalRead(LED) ^ 1);
Serial.println();
printTime = millis();
}

//baca waktu
DateTime now = rtc.now();
int jam = now.hour();
int menit = now.minute();
int detik = now.second();
//Date Only
String tanggal =
String(now.timestamp(DateTime::TIMESTAMP_DATE));
//Serial.println(tanggal);
//Full Timestamp
String waktu =
String(now.timestamp(DateTime::TIMESTAMP_TIME));
//Serial.println(waktu);
Serial.print(now.year(), DEC);
Serial.print('/');
Serial.print(now.month(), DEC);
Serial.print('/');
Serial.print(now.day(), DEC);
Serial.print(" ");
Serial.print(daysOfTheWeek[now.dayOfTheWeek()]);
Serial.print(" ");
Serial.print(now.hour(), DEC);
Serial.print(':');
Serial.print(now.minute(), DEC);
Serial.print(':');
Serial.print(now.second(), DEC);
Serial.println();
Serial.print("RTC Temperature: ");
Serial.print(rtc.getTemperature());
Serial.println(" C");
Serial.println();
if (jam == 8 && menit == 0 && detik == 0) {
    flag = 1;
    myservo.write(45);
    Serial.println("SERVO TERBUKA");
    delay(1000);
} else if (jam == 13 && menit == 0 && detik == 0) {
    flag = 1;
    myservo.write(45);
    Serial.println("SERVO TERBUKA");
    delay(1000);
} else if (jam == 18 && menit == 0 && detik == 0) {
    flag = 1;
    myservo.write(45);
    Serial.println("SERVO TERBUKA");
    delay(1000);
} else {
    flag = 0;
}
//create the JSON document
StaticJsonDocument<200> doc;
doc["tanggal"] = tanggal;
doc["waktu"] = waktu;
doc["temp"] = suhu;

```

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## L-13 Sketch Pemrograman Arduino IDE (UNO)

```
doc["turb"] = turbClass;
doc["ph"] = pHValue;
```

```
//send the JSON document over the "link" serial port
serializeJson(doc, Serial);
serializeJson(doc, linkSerial);
Serial.println();
```

```
Serial.println("=====
=====");
    //delay(500);
}
}

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

### © Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## L-14 Sketch Pemrograman Arduino IDE (ESP)



### © Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
#include <ArduinoJson.h>
#if defined(ESP32)
#include <WiFi.h>
#include <FirebaseESP32.h>
#elif defined(ESP8266)
#include <ESP8266HTTPClient.h>
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#endif

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

/* 1. Define the WiFi credentials */
const char* WIFI_SSID = "KOIKU";
const char* WIFI_PASSWORD = "koiku2021";

/* 2. Define the API Key */
#define API_KEY "AIzaSyBQiY_6lNy4TYezyvk19IamKKbqLR2L0HY"

/* 3. Define the RTDB URL */
#define DATABASE_URL "koiku-1e306-default-rtdb.firebaseio.com"
//<databaseName>.firebaseio.com or
<databaseName>.<region>.firebasedatabase.app

/* 4. Define the user Email and password that already registered
or added in your project */
#define USER_EMAIL "koiku2021@gmail.com"
#define USER_PASSWORD "AfifNaomi?!"

/* 5. IP Address PC Server*/
const char* HOST = "192.168.43.164";

//Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;

int count = 0;

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

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

Serial.println();

Serial.printf("Firebase Client v%s\n\n",
FIREBASE_CLIENT_VERSION);

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

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

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

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

Firebase.begin(&config, &auth);

//Or use legacy authenticate method
//Firebase.begin(DATABASE_URL, "<database secret>");
}

void loop() {
  static uint8_t flag = 0, btn;
  static uint32_t millisBtn, timeBtn = 250;

  if (millis() - millisBtn > timeBtn) {
    if (Firebase.ready()) {
      if (Firebase.getInt(fbdo, "/btn")) {
        if (fbdo.dataType() == "int") {
          btn = fbdo.intData();
          //Serial.print("btn = ");
          //Serial.println(btn);
          if (btn == 1) {
            String outStr = String("*b1#") + String('\n');
            Serial.print(outStr);
            delay(10);
          } else if (btn == 0) {
            String outStr = String("*b0#") + String('\n');
            Serial.print(outStr);
            delay(10);
          }
        }
      } else {
        Serial.println(fbdo.errorReason());
      }
    }
    millisBtn = millis();
  }

  while (Serial.available()) {
    StaticJsonDocument<200> doc;
    DeserializationError err = deserializeJson(doc, Serial);
    if (err == DeserializationError::Ok)
    {

```

## L-14 Sketch Pemrograman Arduino IDE (ESP)



### © Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
const char* tanggal = doc["tanggal"];
const char* waktu = doc["waktu"];
float suhu = doc["temp"];
  if (suhu > 30)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water temperature
is too high");
  }
  else if (suhu < 25)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water temperature
is too low");
  }
const char* keruh = doc["turb"];
float ph = doc["ph"];
  if (ph > 7.5)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water pH is too
high");
  }
  else if (ph < 6)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water pH is too
low");
  }
  /*if (suhu > 30)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water temperature
is too high");
  }
  else if (suhu < 25)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water temperature
is too low");
  }

  if (ph > 7)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water pH is too
high");
  }
  else if (ph < 6)
  {
    // Send Notification
    kirimnotifikasi("KOIKU NOTIFICATION", "Water pH is too
low");
  }
  */

  /*Serial.println(tanggal);
  Serial.println(waktu);
  Serial.println(suhu);
```

## L-14 Sketch Pemrograman Arduino IDE (ESP)



### © Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
Serial.println(keruh);
Serial.println(ph);
Serial.println();*/

if (Firebase.ready())
{
    Serial.printf("Set tanggal... %s\n",
    Firebase.setString(fbdo, "/tanggal", tanggal) ? "ok" :
    fbdo.errorReason().c_str());
    Serial.printf("Set waktu... %s\n",
    Firebase.setString(fbdo, "/waktu", waktu) ? "ok" :
    fbdo.errorReason().c_str());
    Serial.printf("Set temp... %s\n", Firebase.setFloat(fbdo,
    "/temp", suhu) ? "ok" : fbdo.errorReason().c_str());
    Serial.printf("Set turb... %s\n", Firebase.setString(fbdo,
    "/turb", keruh) ? "ok" : fbdo.errorReason().c_str());
    Serial.printf("Set ph... %s\n", Firebase.setFloat(fbdo,
    "/ph", ph) ? "ok" : fbdo.errorReason().c_str());
}
else
{
    Serial.print("deserializeJson() returned ");
    Serial.println(err.c_str());

    while (Serial.available() > 0)
        Serial.read();
}
}

void kirimnotifikasi(String judul, String isi){
    // Merubah spasi menjadi %20
    judul.replace(" ", "%20");
    isi.replace(" ", "%20");

    // Cek koneksi ke server
    WiFiClient client;
    if(!client.connect(HOST, 80)) // 80 adalah port yang dilihat
    dari XAMPP
    {
        Serial.println("Connection Failed");
        return; // Mencoba koneksi kembali
    }
    // Mengirim data ke server untuk trigger notification
    String Link;
    Link =
    "http://192.168.43.164/Notifikasi/kirimnotifikasi.php?judul=" +
    judul + "&isi=" + isi;

    // Eksekusi Link
    HTTPClient http;
    http.begin(Link);
    // Methods GET
    http.GET();
    http.end();

    delay (1000);
}
```




**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**Activity 1 – Splash Screen**

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SplashScreen">

<View
    android:id="@+id/view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/wallpaper"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="0dp"
    tools:ignore="MissingConstraints" />

<TextView
    android:id="@+id/textView20"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="2021"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.977" />

<TextView
    android:id="@+id/textView19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="POLITEKNIK NEGERI JAKARTA"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.943" />

<TextView
    android:id="@+id/textView18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="JURUSAN TEKNIK ELEKTRO"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
  
```

```

app:layout_constraintBottom_toBottomOf="@+id/view"
app:layout_constraintEnd_toEndOf="@+id/view"
app:layout_constraintStart_toStartOf="@+id/view"
app:layout_constraintTop_toTopOf="@+id/view"
app:layout_constraintVertical_bias="0.909" />

```

```

<TextView
    android:id="@+id/textView17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="PROGRAM STUDI TELEKOMUNIKASI"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.875" />

```

```

<TextView
    android:id="@+id/textView16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/bungee_shade"
    android:text="===== "
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.816" />

```

```

<TextView
    android:id="@+id/textView15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="1803332035"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.496"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.742" />

```

```

<TextView
    android:id="@+id/textView14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="NAOMI LISDA NEVITA B"

```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



```

android:textColor="#22369A"
android:textSize="15sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="@+id/view"
app:layout_constraintEnd_toEndOf="@+id/view"
app:layout_constraintHorizontal_bias="0.493"
app:layout_constraintStart_toStartOf="@+id/view"
app:layout_constraintTop_toTopOf="@+id/view"
app:layout_constraintVertical_bias="0.708" />

```

```

<TextView
    android:id="@+id/textView13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="1803332062"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.496"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.653" />

```

```

<TextView
    android:id="@+id/textView12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/paytone_one"
    android:text="AFIF RIO SYAPUTRA"
    android:textColor="#22369A"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.62" />

```

```

<TextView
    android:id="@+id/textView11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/aclonica"
    android:text="powered by :"
    android:textColor="#22369A"
    android:textSize="10sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.561" />

```

```

<TextView

```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



```

android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="@font/aclonica"
android:text="Pemantauan Kualitas Air"
android:textColor="#3F66B5"
android:textSize="15sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="@+id/view"
app:layout_constraintEnd_toEndOf="@+id/view"
app:layout_constraintHorizontal_bias="0.496"
app:layout_constraintStart_toStartOf="@+id/view"
app:layout_constraintTop_toTopOf="@+id/view"
app:layout_constraintVertical_bias="0.032" />

```

```

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/aclonica"
    android:text="dan"
    android:textColor="#3F66B5"
    android:textSize="15sp"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.067" />

```

```

<TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/aclonica"
    android:text="Pemberi Pakan Ikan Koi"
    android:textColor="#3F66B5"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.494"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.102" />

```

```

<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/press_start_2p"
    android:text="KOIKU"
    android:textColor="#3F66B5"
    android:textSize="25sp"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"

```

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

app:layout_constraintVertical_bias="0.467" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="147dp"
    android:layout_height="143dp"
    app:layout_constraintBottom_toBottomOf="@+id/view"
    app:layout_constraintEnd_toEndOf="@+id/view"
    app:layout_constraintStart_toStartOf="@+id/view"
    app:layout_constraintTop_toTopOf="@+id/view"
    app:layout_constraintVertical_bias="0.212"
    app:srcCompat="@drawable/koiku" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

### Activity 2 – Main

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <View
        android:id="@+id/view2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#6E96E6"
        tools:ignore="MissingConstraints"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="0dp" />

    <ImageView
        android:id="@+id/imageView7"
        android:layout_width="493dp"
        android:layout_height="758dp"
        app:layout_constraintBottom_toBottomOf="@+id/view2"
        app:layout_constraintStart_toStartOf="@+id/view2"
        app:layout_constraintTop_toTopOf="@+id/view2"
        app:srcCompat="@drawable/air3" />

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:background="@drawable/kotakyapip"
        android:orientation="horizontal"
        app:layout_constraintBottom_toBottomOf="@+id/view2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHeight_percent=".10"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/view2"
        app:layout_constraintVertical_bias="0.635"
        app:layout_constraintWidth_percent=".80">

        <TextView
            android:id="@+id/textView25"

```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta





```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="15dp"
android:fontFamily="@font/aclonica"
android:padding="10dp"
android:paddingStart="10dp"
android:text="Turbidity:"
android:textColor="#FFFFFF"
android:textSize="10sp"
android:textStyle="bold" />

```

```

<TextView
    android:id="@+id/turbidity"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="15dp"
    android:fontFamily="@font/aclonica"
    android:padding="10dp"
    android:paddingStart="10dp"
    android:text="0"
    android:textColor="#FFFFFF"
    android:textSize="10sp"
    android:textStyle="bold" />

```

```
</LinearLayout>
```

```

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:background="@drawable/kotakyapip"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="@+id/view2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHeight_percent=".10"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/view2"
    app:layout_constraintVertical_bias="0.463"
    app:layout_constraintWidth_percent=".80">

```

```

<TextView
    android:id="@+id/textView24"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="15dp"
    android:fontFamily="@font/aclonica"
    android:padding="10dp"
    android:paddingStart="10dp"
    android:text="pH:"
    android:textColor="#FFFFFF"
    android:textSize="10sp"
    android:textStyle="bold" />

```

```

<TextView
    android:id="@+id/phmeter"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="15dp"
    android:fontFamily="@font/aclonica"
    android:padding="10dp"

```

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



```

        android:paddingStart="10dp"
        android:text="0"
        android:textColor="#FFFFFF"
        android:textSize="10sp"
        android:textStyle="bold" />
</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:background="@drawable/kotakyapip"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="@+id/view2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHeight_percent=".10"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/view2"
    app:layout_constraintVertical_bias="0.291"
    app:layout_constraintWidth_percent=".80">
    <TextView
        android:id="@+id/textView23"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:fontFamily="@font/aclonica"
        android:padding="10dp"
        android:paddingStart="10dp"
        android:text="Temp :"
        android:textColor="#FFFFFF"
        android:textSize="10sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/temperatur"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:fontFamily="@font/aclonica"
        android:padding="10dp"
        android:paddingStart="10dp"
        android:text="0"
        android:textColor="#FFFFFF"
        android:textSize="10sp"
        android:textStyle="bold" />
    <TextView
        android:id="@+id/temperatur2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:fontFamily="@font/aclonica"
        android:padding="10dp"
        android:paddingStart="10dp"
        android:text="°C"
        android:textColor="#FFFFFF"
        android:textSize="10sp"
        android:textStyle="bold" />
</LinearLayout>

```

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

<Button
    android:id="@+id/button2"
    android:layout_width="150dp"
    android:layout_height="60dp"
    android:backgroundTint="#3F51B5"
    android:fontFamily="@font/aclonica"
    android:text=" Off Koi's Feed "
    android:textColor="#FFFFFF"
    android:textSize="10sp"
    app:layout_constraintBottom_toBottomOf="@+id/view2"
    app:layout_constraintEnd_toEndOf="@+id/view2"
    app:layout_constraintHorizontal_bias="0.938"
    app:layout_constraintStart_toStartOf="@+id/view2"
    app:layout_constraintTop_toTopOf="@+id/view2"
    app:layout_constraintVertical_bias="0.852"
    tools:ignore="MissingConstraints" />

<TextView
    android:id="@+id/textView21"
    android:layout_width="172dp"
    android:layout_height="93dp"
    android:fontFamily="@font/bungee_shade"
    android:text="KOIKU"
    android:textColor="#051571"
    android:textSize="35sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="@+id/view2"
    app:layout_constraintEnd_toEndOf="@+id/view2"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="@+id/view2"
    app:layout_constraintTop_toTopOf="@+id/view2"
    app:layout_constraintVertical_bias="0.073" />

<Button
    android:id="@+id/button"
    android:layout_width="150dp"
    android:layout_height="60dp"
    android:backgroundTint="#3F51B5"
    android:fontFamily="@font/aclonica"
    android:text=" On Koi's Feed"
    android:textColor="#FFFFFF"
    android:textSize="10sp"
    app:layout_constraintBottom_toBottomOf="@+id/view2"
    app:layout_constraintEnd_toEndOf="@+id/view2"
    app:layout_constraintHorizontal_bias="0.061"
    app:layout_constraintStart_toStartOf="@+id/view2"
    app:layout_constraintTop_toTopOf="@+id/view2"
    app:layout_constraintVertical_bias="0.852"
    tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

### Java 1 – Splash Screen

```

package com.example.koiku;

import android.content.Intent;
import android.os.Bundle;

```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkannya dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import android.os.Handler;
import android.view.Window;
import androidx.appcompat.app.AppCompatActivity;

public class SplashScreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //menghilangkan ActionBar
        this.requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_splash_screen);

        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(getApplicationContext(),
                MainActivity.class));
                finish();
            }
        }, 5000L); //5000 L = 5 detik
    }
}
```

**Java 2 – Main**

```
package com.example.koiku;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import com.firebase.client.DataSnapshot;
import com.firebase.client.Firebase;
import com.firebase.client.FirebaseError;
import com.firebase.client.ValueEventListener;

public class MainActivity extends AppCompatActivity {

    //inisialisasi textview temperatur, phmeter, turbidity
    private TextView temperatur ;
    private TextView phmeter ;
    private TextView turbidity ;
    private Button button;
    private Button button2;

    //buat reference untuk firebase (koneksi server/host firebase)
    private Firebase mRef ;
    private Firebase mRef1 ;
    private Firebase mRef2 ;
    private Firebase mRef3 ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

setContentView(R.layout.activity_main);

//membaca komponen temperatur, phmeter, turbidity
(textview)
    temperatur = (TextView)findViewById(R.id.temperatur);
    phmeter = (TextView)findViewById(R.id.phmeter);
    turbidity = (TextView)findViewById(R.id.turbidity);
    button = (Button)findViewById(R.id.button);
    button2 = (Button)findViewById(R.id.button2);

//buka koneksi ke host firebase
mRef= new Firebase("https://koiku-1e306-default-
rtdb.firebaseio.com/temp");
    mRef1= new Firebase("https://koiku-1e306-default-
rtdb.firebaseio.com/ph");
    mRef2= new Firebase("https://koiku-1e306-default-
rtdb.firebaseio.com/turb");
    mRef3= new Firebase("https://koiku-1e306-default-
rtdb.firebaseio.com/btn");

//proses untuk membaca secara realtime
mRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        //mengambil nilai field dari temp, ph, turb
        String temp = dataSnapshot.getValue(String.class);
        //menampilkan pada komponen temperatur
        temperatur.setText(temp);
    }
    @Override
    public void onCancelled(FirebaseError firebaseError) {
    }
});
//proses untuk membaca secara realtime
mRef1.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        //mengambil nilai field dari temp, ph, turb
        String ph = dataSnapshot.getValue(String.class);
        //menampilkan pada komponen phmeter
        phmeter.setText(ph);
    }
    @Override
    public void onCancelled(FirebaseError firebaseError) {
    }
});
//proses untuk membaca secara realtime
mRef2.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        //mengambil nilai field dari temp, ph, turb
        String turb = dataSnapshot.getValue(String.class);
        //menampilkan pada komponen turbidity
        turbidity.setText(turb);
    }
    @Override
    public void onCancelled(FirebaseError firebaseError) {

```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

    }
  });
  //proses untuk merubah 0 menjadi 1
  button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
      mRef3.setValue(1);
    }
  });
  //proses untuk mengembalikan 1 menjadi 0
  button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
      mRef3.setValue(0);
    }
  });
}
}

```

**Java 3 – MyService Notifikasi**

```

package com.example.koiku;

import android.app.Notification;
import android.app.NotificationManager;
import android.util.Log;
import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;
import static com.example.koiku.App.CHANNEL_1_ID;

public class MyService extends FirebaseMessagingService {
    public static final String TAG = "MyTag";

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);
        Log.d(TAG, "OnMessageReceived: Called");
        Log.d(TAG, "OnMessageReceiver: MessageReceived From:"
+remoteMessage.getFrom());

        if (remoteMessage.getNotification() !=null) {
            String title =
remoteMessage.getNotification().getTitle();
            String body =
remoteMessage.getNotification().getBody();
            Notification notification = null;
            if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O) {
                notification = new Notification.Builder(this,
CHANNEL_1_ID).setSmallIcon(android.R.drawable.ic_popup_reminder)
                    .setContentTitle(title)
                    .setContentText(body)
                    .build();
            }
            NotificationManager manager =
(NotificationManager) getSystemService (NOTIFICATION_SERVICE);

```



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        manager.notify(1002, notification);
    }

    if (remoteMessage.getData().size()>0) {
        Log.d(TAG, "onMessageReceived: Data:"
+remoteMessage.getData().toString());
    }
}

@Override
public void onDeletedMessages() {
    super.onDeletedMessages();
    Log.d(TAG, "onDeletedMessages: Called");
}

@Override
public void onNewToken(String token) {
    super.onNewToken(token);
    Log.d(TAG, "NewToken: " +token);
}
}

```

**Java 4 – Application Notifikasi**

```

package com.example.koiku;

import android.app.Application;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.os.Build;

import com.firebase.client.Firebase;

public class App extends Application {
    public static final String CHANNEL_1_ID = "channell";
    @Override
    public void onCreate() {
        super.onCreate();
        createNotificationChannels();
        Firebase.setAndroidContext(this);
    }
    private void createNotificationChannels() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
            NotificationChannel channell = new
NotificationChannel(
                CHANNEL_1_ID,
                "Channel 1",
                NotificationManager.IMPORTANCE_HIGH
            );
            channell.setDescription("This is Channel 1");
            NotificationManager manager = (NotificationManager)
getSystemService(NotificationManager.class);
            manager.createNotificationChannel(channell);
        }
    }
}

```





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengummumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

