



TUGAS AKHIR - EE184801
UJI KUALITAS MINUMAN MENGGUNAKAN SENSOR
POTENSIOMETRIK, KONDUKTIVITAS LISTRIK, OPTIK
DAN METODE JARINGAN SYARAF TIRUAN

YUSRI SAPUTRA
NRP 0711154000130

Dosen Pembimbing
Dr. Muhammad Rivai ST., MT.
Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Insitut Teknologi Sepuluh Nopember
Surabaya 2019



TUGAS AKHIR – EE184801
UJI KUALITAS MINUMAN MENGGUNAKAN
SENSOR POTENSIOMETRIK, KONDUKTIVITAS
LISTRIK, OPTIK DAN METODE JARINGAN
SYARAF TIRUAN

YUSRI SAPUTRA
NRP 07111540000130

Dosen Pembimbing
Dr. Muhammad Rivai ST., MT.
Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019



**FINAL PROJECT – EE184801
DRINK QUALITY ASSESSMENT USING
POTENTIOMETRIC, ELECTRICAL
CONDUCTIVITY, OPTICAL SENSORS AND
NEURAL NETWORK METHOD**

**YUSRI SAPUTRA
NRP 07111540000130**

**Supervisor
Dr. Muhammad Rivai ST., MT.
Muhammad Attamimi, B.Eng., M.Eng., Ph.D.**

**ELECTRICAL ENGINEERING DEPARTEMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019**

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "Uji Kualitas Minuman Menggunakan Sensor Potensiometrik, Konduktivitas Listrik, Optik dan Metode jaringan Syaraf Tiruan" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

Surabaya,

Yusi Saputra
07111540000130

-Halaman Ini Sengaja Dikosongkan-

**UJI KUALITAS MINUMAN MENGGUNAKAN
SENSOR POTENSIOMETRIK, KONDUKTIVITAS
LISTRIK, OPTIK DAN METODE JARINGAN SYARAF
TIRUAN**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk Memperoleh
Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

Dosen Pembimbing I



Dr. Muhammad Rivai ST., MT.

Dosen Pembimbing II



Muhammad Attamimi, M.Eng., Ph.D.



-Halaman Ini Sengaja Dikosongkan-

UJI KUALITAS MINUMAN MENGGUNAKAN SENSOR POTENSIOMETRIK, KONDUKTIVITAS LISTRIK, OPTIK DAN METODE JARINGAN SYARAF TIRUAN

Nama : Yusri Saputra
Pembimbing I : Dr. Muhammad Rivai ST., MT.
Pembimbing II : Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

ABSTRAK

Teknologi pengemasan pada minuman banyak diaplikasikan sehingga minuman bisa awet dan disimpan lebih lama. Namun apa jadinya ketika minuman tersebut dibiarkan terbuka dalam suhu ruangan. Pastinya akan terjadi pembusukan dikarenakan minuman akan lebih mudah bercampur dengan mikroba yang membuat minuman tersebut cepat basi. Kondisi basi ditandai dengan munculnya beberapa kondisi yaitu munculnya bau busuk, terjadi penggumpalan, rasa yang berubah menjadi asam, serta warna yang lebih pucat. Maka dibuatlah sebuah sistem yang dapat mengenali minuman segar, kurang segar, dan basi. Dari kondisi tersebut memudahkan manusia dalam mengidentifikasi kualitas minuman. Parameter yang digunakan dalam tugas akhir ini adalah derajat keasaman (pH) dengan pengukuran menggunakan sensor potensiometrik, TDS (*Total Dissolved Solids*) dengan pengukuran menggunakan sensor konduktivitas listrik dan kekeruhan atau NTU (Nephelometric Turbidity Unit) dengan pengukuran menggunakan sensor optik serta menggunakan arduino mega 2560 sebagai mikrokontroler dan LCD berukuran 20x4 untuk menampilkan nilai pembacaan. 5 jenis sampel yang digunakan antara lain : Susu Kedelai, Susu UHT (*ultra high temperature*), Air Jeruk, Susu Sapi dengan merek Bear Brand dan Air Kelapa diidentifikasi menggunakan metode jaringan syaraf tiruan dengan masing-masing pengujian 3 kategori, maka dapat disimpulkan bahwa sistem ini bekerja dengan keakurasian mencapai 73.34%

Kata Kunci : Sensor Potensiometrik, Sensor Konduktivitas Listrik, Sensor Optik, Jaringan Syaraf Tiruan, Kualitas Minuman

--Halaman Ini Sengaja Dikosongkan--

Name : Yusri Saputra
Supervisor 1st : Dr. Muhammad Rivai ST., MT.
Supervisor 2nd : Muhammad Attamimi, B.Eng., M.Eng., Ph.D.

ABSTRACT

The packaging technology in many drinks is applied so that the beverage can last longer and be stored longer. But what happens when the drink is left open at room temperature. Surely decay will occur because the drink will be more easily mixed with microbacteria which makes the drink spoiled quickly. Spoiled conditions are characterized by the emergence of several conditions, namely the smell bad, clumping, taste sour, and a paler color. So a system was created that could recognize fresh, Sour and Spoiled. From these conditions make it easier for humans to identify the quality of drinks. The parameters used in this research are the acidity (pH) measured by using a potentiometric sensor, TDS (Total Dissolved Solids) measured by using electrical conductivity and turbidity sensors or NTU (Nephelometric Turbidity Unit) measured by using optical sensor and Arduino Mega 2560 as a microcontroller and 20x4 LCD for display the value. The 5 types of samples used include: Soymilk, UHT (Ultra high temperature) milk, Orange Water, cow milk with "Bear Brand" brand and Coconut Water brand identified using artificial neural network methods with each testing 3 categories, it can be concluded that the system this works with accuracy reaching 73.34%

Keywords: *Potentiometric Sensor, Conductivity Sensor, Optical Sensor, Neural Networks, Beverage Quality*

-Halaman Ini Sengaja Dikosongkan-

KATA PENGANTAR

Puji syukur atas rahmat yang diberikan oleh Tuhan Yang Maha Esa. Karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan penelitian ini. Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak baik dukungan secara moril maupun materil. Terima kasih yang sebesar-besarnya penulis sampaikan kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini, diantaranya :

1. Dr. Muhammad Rivai ST., MT. dan Muhammad Attamimi, B.Eng., M.Eng., Ph.D. Selaku dosen pembimbing pertama dan kedua atas bimbingan, arahan dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini
2. Kepada Bapak Dr.Eng., Ir. Totok Mujiono M.Ikom., Dr. Astria Nur Irfansyah, ST., M.Eng., Ir. Harris Pirngadi, MT., dan Ir. Tasripan, MT. sebagai Dosen Penguji yang telah menguji dan mengevaluasi tugas akhir ini
3. Kepada Kepala Departemen Teknik Elektro Bapak Dr. Eng. Ardyono Priyadi, ST., M.Eng.
4. Kedua Orang Tua tercinta, Bapak H. Safri dan Ibu Hj. Nurbaya yang tak henti-hentinya mendoakan, memberikan motivasi, nasihat dan dukungannya
5. Kepada rekan-rekan asisten Lab. Elektronika Industri B402 yang senantiasa membantu dan memberikan semangat dalam mengerjakan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran dan kritik serta masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya,

Yusri Saputra
07111540000130

-Halaman Ini Sengaja Dikosongkan-

DAFTAR ISI

KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	2
1.5. Metodologi penelitian.....	2
1.6. Sistematika Penulisan.....	4
1.7. Relevansi	5
BAB 2 TEORI PENUNJANG	7
2.1. Kualitas Minuman.....	7
2.2. Pembusukan Susu	8
2.3. Pembusukan Sari Kedelai	9
2.4. Pembusukan Air Jeruk	10
2.5. Pembusukan Air Kelapa.....	10
2.6. Sensor	11
2.6.1. Sensor Potensiometrik	11
2.4.2. Sensor Konduktivitas Listrik	15
2.4.3. Sensor Optik	17
2.7. Metode Jaringan Syaraf Tiruan.....	19
2.8. Komponen Jaringan Syaraf	20
2.9. Fungsi Aktivasi	21
2.10. I2C (<i>Inter Integrated Circuit</i>).....	22
2.11. LCD (<i>Liquid Crystal Display</i>).....	24
2.12. Arduino.....	26
BAB 3 PERANCANGAN SISTEM	29
3.1. Diagram Blok Sistem	29
3.2. Perancangan Perangkat Keras	29
3.2.1. Perangkat Elektrik	30
3.2.1.1. Sensor SEN0161	30
3.2.1.2. Sensor SEN0244	32
3.2.1.3. Sensor SEN0189	33
3.2.1.4. Arduino Mega 2560.....	34
3.2.1.5. Modul Serial I2C LCD 20x4.....	36
3.2.2. Perancangan Letak Posisi Sensor	36
3.3. Perangkat Lunak	37
3.4. Proses Jaringan Syaraf Tiruan.....	39

BAB 4 PENGUJIAN DAN ANALISIS	43
4.1. Pengujian Perangkat Keras	43
4.1.1. Sensor SEN0161	43
4.1.2. Sensor SEN0244	45
4.1.3. Sensor SEN0198	47
4.2. Pengujian Sampel Minuman	48
4.3. Pengujian Sistem Jaringan Syaraf Tiruan	57
4.3.1. Proses Learning Dengan Menggunakan Dev++	58
4.3.2. Proses Identifikasi Minuman Dengan Menggunakan Dev++	59
4.4. Pengujian Sistem	60
BAB 5 PENUTUP	65
5.1. Kesimpulan	65
5.2. Saran	65
DAFTAR PUSTAKA	67
LAMPIRAN A	69
LAMPIRAN B	87
LAMPIRAN C	95
LAMPIRAN D	99
LAMPIRAN E	103
LAMPIRAN F	115
BIODATA PENULIS	119

DAFTAR GAMBAR

Gambar 2.1. Pengukuran Potensiometrik.....	12
Gambar 2.2. Proses Oksidasi Pada Anoda	13
Gambar 2.3. Proses Reduksi Pada Katoda	14
Gambar 2.4. Pengukuran pH.....	14
Gambar 2.5. Pergerakan Ion Dalam Larutan.....	15
Gambar 2.6. Struktur Ion	16
Gambar 2.7. Cara Kerja Sensor Optik.....	22
Gambar 2.8. Rangkaian Sensor Fotodiode	23
Gambar 2.9. Susunan Syaraf Manusia	19
Gambar 2.10. Struktur Neuron Jaringan Syaraf Tiruan	20
Gambar 2.11. Jaringan Syaraf Dengan 3 Lapisan.	21
Gambar 2.12. Fungsi Aktivasi : Sigmoid Biner	22
Gambar 2.13. Modul Inter Integrated Circuit	22
Gambar 2.14. Kondisi Sinyal Start dan Stop.....	23
Gambar 2.15. Sinyal ACK dan NACK	23
Gambar 2.16. Trasfer Bit pada I ² C bus	24
Gambar 2.17. Liquid Crystal Display 16x4	25
Gambar 2.18. Arduino Uno.....	26
Gambar 3.1. Diagram Blok Perancangan Elektrik	30
Gambar 3.2. Kotak Wadah Rangkaian.....	30
Gambar 3.3. SEN0161 Sensor Potensiometrik	30
Gambar 3.4. Dimensi SEN0161.....	31
Gambar 3.5. Modul Sensor SEN0161	31
Gambar 3.6. Gravity Analog TDS Sensor Meter SKU SEN0244.....	32
Gambar 3.7. Modul Sensor SEN0244.....	33
Gambar 3.8. Turbidity Sensor SKU SEN0189.....	33
Gambar 3.9. Connection Diagram	34
Gambar 3.10. Tampilan Arduino Mega 2560	35
Gambar 3.11. Arduino Mega 2560 I2C-LCD	36
Gambar 3.12. Posisi lubang sensor	37
Gambar 3.13. Tata Letak Posisi Sensor	37
Gambar 3.14. Flowchart Pembacaan Nilai Ketiga Sensor	38
Gambar 3.15. Proses Learning Jaringan Syaraf Tiruan.....	39
Gambar 3.16. Flowchart Proses Learning JST	40
Gambar 3.17. Proses Jaringan Syaraf Tiruan	41
Gambar 4.1. Kalibrasi Dengan pH Buffer 7.....	44
Gambar 4.2. Kalibrasi Dengan pH Buffer 4.....	44
Gambar 4.3. Kandungan TDS Dalam Air	45
Gambar 4.4. Pengujian Sensor Konduktivitas Listrik	46
Gambar 4.5. Perbandingan Kekeruhan dan Tegangan	47

Gambar 4.6. Pengujian Sensor Optik.....	48
Gambar 4.7. Grafik Pengujian Susu Kedelai	49
Gambar 4.8. Grafik Pengujian Susu UHT	51
Gambar 4.9. Grafik Pengujian Jus Jeruk.....	53
Gambar 4.10. Grafik Pengujian Bear Brand	55
Gambar 4.11. Grafik Pengujian Air Kelapa.....	57
Gambar 4.12. Training Jaringan Syaraf Tiruan.....	58
Gambar 4.13. Backpropagation	58
Gambar 4.14. Proses <i>Learning</i> Dengan Dev++	59
Gambar 4.15. Proses Identifikasi Dengan Menggunakan Dev++	59
Gambar 4.16. <i>Display</i> Pembacaan Susu Kedelai Segar	60
Gambar 4.17. <i>Display</i> Pembacaan Susu Kedelai Kurang Segar	60
Gambar 4.18. <i>Display</i> Pembacaan Susu Kedelai Basi	61
Gambar 4.19. <i>Display</i> Pembacaan Susu UHT Segar	61
Gambar 4.20. <i>Display</i> Pembacaan Susu UHT Kurang Segar	61
Gambar 4.21. <i>Display</i> Pembacaan Susu UHT Basi	61
Gambar 4.22. <i>Display</i> Pembacaan Air Jeruk Segar	62
Gambar 4.23. <i>Display</i> Pembacaan Air Jeruk Kurang Segar	62
Gambar 4.24. <i>Display</i> Pembacaan Air Jeruk Basi	62
Gambar 4.25. <i>Display</i> Pembacaan Susu Bear Brand Segar	62
Gambar 4.26. <i>Display</i> Pembacaan Susu Bear Brand Kurang Segar	63
Gambar 4.27. <i>Display</i> Pembacaan Susu Bear Brand Basi	63
Gambar 4.28. <i>Display</i> Pembacaan Air Kelapa Segar.....	63
Gambar 4.29. <i>Display</i> Pembacaan Air Kelapa Kurang Segar.....	63
Gambar 4.30. <i>Display</i> Pembacaan Air Kelapa Basi.....	64

DAFTAR TABEL

Tabel 2.1. Tipe sensor yang dipakai.....	11
Tabel 2.2. Konfigurasi Pin LCD	26
Tabel 3.1. Fungsi Dari Masing-Masing Port.....	31
Tabel 3.2. Konfigurasi Pin Modul Sensor SEN0244	33
Tabel 4.1. Kalibrasi dengan pH <i>Buffer</i> 7.....	44
Tabel 4.2. Kalibrasi dengan pH <i>Buffer</i> 4.....	44
Tabel 4.3. Kandungan TDS dalam Air Menurut WHO.....	46
Tabel 4.4. Data Pengukuran Kandungan TDS dalam Air	46
Tabel 4.5. Pengujian Kekeruhan Air	48
Tabel 4.6. Pengujian Susu Kedelai	48
Tabel 4.7. Pengujian Susu UHT.....	50
Tabel 4.8. Pengujian Jus Jeruk.....	52
Tabel 4.9. Pengujian Bear Brand	54
Tabel 4.10. Pengujian Air Kelapa	56
Tabel 4.11. Hasil Identifikasi Minuman Menggunakan Dev++	60

-Halaman Ini Sengaja Dikosongkan-

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Minuman merupakan salah satu bahan makanan yang sangat gampang untuk ditemui. Minuman yang sering kita jumpai terbagi dalam beberapa jenis, salah satu yang paling sering kita jumpai adalah minuman jus atau olahan buah-buahan dan minuman susu. Saat ini minuman-minuman tersebut dapat kita jumpai dijual dalam bergabai jenis macam bentuk serta dikemas dengan kemasan yang praktis dan menarik, seperti dalam kaleng, plastik atau dalam kertas karton. Manfaat dari pengemasan ini adalah minuman dapat disajikan dalam bentuk yang lebih praktis dan kondisi nutrisinya dapat terjaga. Namun kualitas minuman dapat berubah saat kemasan minuman terbuka dan didiamkan dalam beberapa saat dalam kondisi suhu ruangan, kondisi ini dapat memicu minuman menjadi cepat basi dikarenakan adanya aktivitas mikroorganisme di dalam minuman yang berpengaruh dalam proses pembusukan minuman. Perubahan rasa, bau, dan bentuk minuman menjadi parameter manusia dalam menentukan kualitas minuman apakah masih baik untuk diminum atau bahkan sudah tidak layak dan justru akan membahayakan kesehatan. Beberapa kasus ditemukan bahwa masalah keracunan banyak disebabkan dari minuman yang basi, hal ini dikarenakan kandungan bakteri jahat yang tumbuh dalam minuman basi berbahaya bagi kesehatan [1]. Parameter pembusukan sebenarnya mempunyai banyak indikasi. Namun, pada penelitian ini hanya digunakan tiga parameter, yakni pH, TDS (*Total Dissolved Solids*), dan kekeruhan (turbiditas).

Metode pengujian ini menggunakan sensor potensiometri yang akan membantu untuk menghitung nilai kadar keasaman (pH), menggunakan sensor konduktivitas yang akan membantu untuk mendapatkan informasi mengenai konsentrasi ionik total (misalnya, senyawa terlarut), dan sensor optik yang akan membantu untuk memonitor kadar kemurnian atau kekeruhan dalam minuman [2].

Tingkat kekeruhan diukur dengan memanfaatkan perubahan tegangan sensor akibat perubahan kekeruhan. Perancangan sensor sistem monitoring ini terdiri dari perancangan perangkat keras dan lunak.

Perangkat keras terdiri dari sistem sensor, Arduino Mega 2560. Sedangkan perangkat lunak menggunakan Algoritma Jaringan Syaraf Tiruan yang akan membantu membedakan jenis kualitas minuman. Dari data analog tersebut akan dikonversi oleh *Analog to Digital Converter* pada Arduino Mega 2560 menjadi data digital yang merepresentasikan tingkat perkembangan mikrobakteri yang ada pada minuman. Dengan menggunakan Sistem Elektronika Cerdas yang memanfaatkan Metode Jaringan Saraf Tiruan (*Neural Network*), diharapkan alat ini dapat mengenali minuman yang layak ataupun tidak layak untuk dikonsumsi.

1.2. Perumusan Masalah

1. Bagaimana cara membaca parameter kualitas minuman.
2. Jenis mikrokontroler apa yang digunakan.
3. Bagaimana cara mengenali kualitas dari minuman.

1.3. Tujuan Penelitian

1. Penggunaan sensor potensiometrik, konduktivitas dan optik untuk menentukan parameter kualitas minuman.
2. Implementasi mikrokontroler arduino mega 2560 untuk membaca dan memproses kondisi minuman
3. Metode Jaringan Syaraf Tiruan digunakan untuk mengenali kualitas minuman.

1.4. Batasan Masalah

1. Pembacaan parameter minuman dalam ruang lingkup derajat keasaman (pH), Total Zat Padat Terlarut (TDS), Kekeruhan
2. Dapat membedakan minuman yang segar, kurang segar dan basi pada minuman
3. Sampel minuman adalah Susu Kedelai, Jus Jeruk, Susu Bear Brand Air Kelapa dan Susu UHT (*Ultra High Temperature*)

1.5. Metodologi penelitian

Dalam penyelesaian tugas akhir ini digunakan metodologi sebagai berikut :

1. Studi Literatur Uji Kualitas Minuman
Studi literatur berisi serangkaian kegiatan pencarian dan pengkajian sumber-sumber yang relevan dan terpercaya yang menjadi acuan dalam penulisan tugas akhir ini.
2. Perencanaan Sistem
Perencanaan yang di maksud bertujuan untuk mendapatkan desain dan mekanisme yang optimal dari alat pengujian kualitas minuman. Perancangan system ini terdiri atas 2 bagian, yaitu :
 - a. Perancangan *Hardware*
Pada tugas akhir ini menggunakan sensor potensiometrik (SEN0161), sensor konduktivitas listrik (SEN0244) dan sensor optik (SEN0189) sebagai alat yang digunakan untuk membaca kondisi dari sebuah minuman. Menggunakan Arduino Mega 2560 sebagai mikrokontroler dan LCD 20x4 + I2C yang digunakan sebagai alat yang akan menampilkan nilai sensor dan *output* dari Proses Jaringan Syaraf Tiruan
 - b. Perancangan *Software*
Perancangan perangkat lunak atau *software* meliputi proses pemograman pembacaan nilai dari ketiga sensor menggunakan Arduino Ide dan proses pemograman Jaringan Syaraf Tiruan menggunakan Dev++
3. Perangkaian Komponen Sensor, Mikrokontroler dan LCD
Perangkaian seluruh elemen komponen ini dimaksudkan untuk rancang bangun alat pengujian kualitas minuman dari sisi mekanis. Tahap ini dilakukan penganan tata letak komponen. Wadah yang digunakan adalah kotak elektronik berwarna hitam yang dilubangi untuk memberikan akses port yang digunakan dan untuk menampilkan LCD.
4. Pengujian Data Dari Sampel Minuman
Pengujian ini dimaksudkan untuk memastikan bahwa kinerja masing-masing komponen dari hasil perealisasiian sistem dapat berfungsi sesuai dengan yang diharapkan. Tahap ini adalah tahap pengujian sensor dimana ketiga sensor dapat bekerja sebagai berikut :
 - a. Sensor SEN0161 adalah sensor yang digunakan untuk mengukur nilai pH dari minuman tersebut.

- b. Sensor SEN0244 adalah sensor yang digunakan untuk mengukur nilai dari Total Zat Padat (TDS) yang terlarut pada minuman tersebut.
 - c. Sensor SEN0189 adalah sensor yang digunakan untuk mengukur nilai NTU atau kekeruhan dari minuman tersebut.
5. **Pengelolaan Data dengan Metode Jaringan Saraf Tiruan**
Langkah ini bertujuan untuk mengolah proses *learning* dari data yang didapatkan oleh sensor, yang akan selanjutnya diolah ke dalam metode feed forward untuk mengetahui kondisi dari kualitas minuman yang diuji.
 6. **Penyusunan Laporan Akhir**
Tahap penulisan laporan Tugas Akhir dilakukan pada saat tahap pengujian sistem dimulai serta setelahnya.
 7. **Penulisan Makalah Jurnal POMITS**
Penulisan jurnal dilakukan sebagai sarana publikasi penelitian. Hal ini juga dilakukan agar penelitian mendapatkan legalitas dan pengakuan resmi dari dunia pendidikan.

1.6. Sistematika Penulisan

Laporan tugas akhir ini terdiri dari Lima Bab dengan sistematika penulisan sebagai berikut :

- **Bab 1 : Pendahuluan**
Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi penelitian, sistematika penulisan dan relevansi.
- **Bab 2 : Teori Penunjang**
Bab ini menjelaskan tentang dasar-dasar teori yang dibutuhkan dalam pengerjaan tugas akhir ini, yang meliputi teori dasar tentang Minuman, Pembusukan Minuman, pH, TDS, Kekeruhan, Sensor, Jaringan Syaraf Tiruan, LCD, I2C dan Arduino.
- **Bab 3 : Perancangan Sistem**
Bab ini menjelaskan tentang diagram blok, perencanaan sistem perangkat keras, elektrik dan mekanik, perancangan tata letak

komponen, pembacaan nilai dari ketiga sensor, proses Jaringan Syaraf Tiruan dan menampilkan hasil proses Jaringan Syaraf Tiruan.

- Bab 4 : Pengujian dan Analisis
Bab ini menjelaskan tentang hasil yang didapat dari pengujian secara keseluruhan.
- Bab 5 : Penutup
Bab ini menjelaskan tentang kesimpulan dan saran untuk pengembangan kedepan.

1.7. Relevansi

Hasil dari tugas akhir ini ditujukan untuk :

1. Dapat membedakan Kualitas suatu minuman dengan baik
2. Dapat mengaplikasikan Metode Jaringan Syaraf tiruan pada pengujian sampel minuman dengan menggunakan sensor potensiometrik, sensor konduktivitas listrik, dan sensor optik

-Halaman Ini Sengaja Dikosongkan-

BAB 2

TEORI PENUNJANG

2.1. Kualitas Minuman

Minuman atau beverage mempunyai pengertian bahwa semua jenis cairan yang dapat diminum (drinkable liquid) kecuali obat-obatan. Minuman bagi kehidupan manusia mempunyai beberapa fungsi yang mendasar yaitu sebagai penghilang rasa haus, perangsang nafsu makan, sebagai penambah tenaga, dan sebagai sarana untuk membantu pencernaan makanan [3]. Minuman seharusnya tidak mengandung kuman patogen yang dapat membahayakan kesehatan manusia. Tidak mengandung zat kimia yang dapat mengubah fungsi tubuh, tidak korosif, dan tidak merugikan secara ekonomis. Pada hakekatnya, hal ini bertujuan untuk mencegah terjadi serta meluasnya penyakit bawaan air atau water borne disease [4].

Pembusukan adalah proses perubahan komposisi (dekomposisi) makanan baik sebagian atau seluruhnya pada minuman dari keadaan yang normal menjadi keadaan yang tidak normal yang tidak dikehendaki sebagai akibat pematangan alam (maturasi), pencemaran (kontaminasi) atau sebab lain [5]. Penyebab terjadinya pembusukan adalah:

1. Oksidasi Alami

Oksidasi merupakan reaksi yang sangat umum terjadi di alam. Secara sederhana, oksidasi adalah interaksi antara molekul oksigen (yang biasanya berada di udara) dengan zat lain, seperti logam hingga jaringan hidup. Oksidasi ternyata ikut bertanggung jawab dalam turunnya kualitas minuman. Interaksi antara oksigen yang ada di udara dengan molekul-molekul penyusun makanan membuat struktur molekul makanan berubah. Contoh minuman yang mudah teroksidasi adalah vitamin dan lemak. Bau tengik dari minuman basi bisa diakibatkan oleh molekul lemak yang sudah teroksidasi. Sedangkan buah, proses oksidasi pigmen yang terdapat di buah tersebut juga dapat mengubah warna buah menjadi gelap. Selain membuat apel yang baru dipotong jadi kecoklatan, reaksi oksidasi juga menyebabkan besi menjadi berkarat hingga yang membuat minyak bumi dan kayu bisa dibakar (molekul oksigen bereaksi dengan molekul karbon) [6].

2. Pengaruh Mikroorganisme

Bahan makanan sering terkontaminasi oleh mikroorganisme jika kebersihannya kurang dijaga. Mikroorganisme, seperti bakteri atau jamur, sangat berperan sebagai organisme pengurai. Sebagai pengurai, organisme-organisme tersebut mengubah molekul-molekul yang ada di

dalam minuman sehingga kualitas minuman turun. Mikroorganisme itu dapat berkembang biak dengan cepat saat media tumbuhnya cocok (nutrisi cocok, suhunya pas, dan lain-lain), mikroorganisme menjadi salah satu faktor utama yang dapat membuat minuman jadi cepet basi apabila minuman tidak disimpan dengan baik [6].

3. Reaksi Enzimatik

Enzim terdapat di dalam tubuh makhluk hidup. Enzim adalah senyawa yang berguna sebagai katalisator atau mempercepat proses suatu reaksi. Misalnya, terjadi reaksi dalam tubuh kita yang seharusnya berlangsung selama 2 hari. Namun dengan bantuan enzim, reaksi tersebut dapat berjalan hanya dalam waktu 2 jam saja. Contoh enzim pada manusia adalah enzim-enzim pada sistem pencernaan (enzim amilase, pepsin, lipase, dll) yang berfungsi untuk membantu proses perombakan makanan di sistem pencernaan kita [6].

Enzim juga terdapat di bahan makanan yang kita konsumsi sehari-hari. Baik enzim alami yang berasal dari bahan makanan, dari mikroorganisme yang mengkontaminasi bahan makanan, atau bahkan dari air liur yang manusia keluarkan ketika menggigit makanan. Dalam konteks mengurangi kualitas minuman, enzim berperan dalam pembusukan dan perubahan tekstur dari makanan hasil bumi yang belum diolah, seperti daging, sayur dan buah-buahan. Mayoritas enzim yang dapat merusak makanan adalah enzim yang bersifat mempercepat proses oksidasi dari makanan. Selain itu, enzim yang mengatur pematangan buah juga secara alami dapat menyebabkan buah menjadi lembek dan berwarna kehitaman [6].

2.2. Pembusukan Susu

Gejala kerusakan pada susu dapat diketahui melalui adanya perubahan rasa, bau dan kenampakkannya. Gejala kerusakan pada susu adalah adanya perubahan rasa susu menjadi asam, adanya penggumpalan pada susu, terbentuknya gas, terbentuk lendir dan berbau busuk.

1. Perubahan susu menjadi asam

Adanya perubahan rasa susu menjadi asam, disebabkan oleh pertumbuhan bakteri pembentuk asam, terutama bakteri asam laktat dan bakteri *E. coli*. Bakteri asam laktat dan *E. coli* dalam susu akan memfermentasi laktosa menjadi asam laktat sehingga rasa susu menjadi asam. Proses tersebut akan berlanjut di mana asam laktat akan membuat kasein menggumpal melalui proses koagulasi. Hal ini memungkinkan

gejala kerusakan yang ditandai dengan perubahan rasa menjadi asam diikuti dengan terbentuknya gumpalan pada susu [7].

2. Penggumpalan pada susu

Penggumpalan pada susu disebabkan oleh pemecahan protein susu oleh bakteri pemecah protein. Pemecahan protein mungkin disertai oleh terbentuknya asam atau tanpa asam. Penggumpalan tanpa adanya pengasaman dapat disebabkan oleh aktivitas bakteri *Bacillus cereus*. *Bacillus cereus* akan mengurai P-lipid pada membran butiran lemak sehingga lemak berkoalisi dan memisah ke permukaan. Hal tersebut yang menyebabkan susu menggumpal [7].

3. Terbentuknya gas

Terbentuknya gas, disebabkan oleh pertumbuhan dua kelompok mikroba, yaitu bakteri yang membentuk gas H₂ (Hidrogen) dan CO₂ (karbon dioksida) seperti bakteri *E.coli* dan bakteri pembentuk spora. Bakteri yang hanya membentuk CO₂ misalnya bakteri asam laktat tertentu dan yeast (jamur) [7].

4. Terbentuknya lendir

Terbentuknya lendir dapat disebabkan oleh adanya bakteri pembusuk seperti *Micrococcus* sp., *Pseudomonas* sp., dan *Bacillus* sp yang menguraikan protein menjadi asam amino dan merombak lemak dengan enzim lipase sehingga susu menjadi asam dan berlendir.

5. Bau busuk

Bau busuk disebabkan oleh pertumbuhan bakteri pemecah protein menjadi senyawa-senyawa berbau busuk. Adanya mikroorganisme pada produk susu dapat memungkinkan terjadi kontaminasi setelah terjadinya pemanasan. Kontaminasi ini terjadi karena rendahnya sanitasi pada saat pengemasan produk. Selain itu juga disebabkan oleh adanya penyimpanan produk yang kurang tepat serta adanya kerusakan pada kemasan susu.

2.3. Pembusukan Sari Kedelai

Sari kedelai yang rusak ditandai dengan berubahnya bau, warna, rasa, atau mengental, kemudian terjadi pemisahan air dengan endapan sari kedelai [8]. Adanya perubahan fisik baik dari bau, warna serta rasa berhubungan dengan berkembangnya mikroorganisme perusak pangan [9]. Semakin lama menyimpan susu kedelai pada suhu ruang, maka akan semakin banyak basa yang dihasilkan akibat semakin meningkatnya aktivitas mikroorganisme yang pada akhirnya mengakibatkan terjadinya

pembusukan. Proses pembusukan akan diikuti dengan peningkatan pH, dan dalam keadaan tersebut akan diikuti pula dengan peningkatan pertumbuhan bakteri (Suradi, 2012).

2.4. Pembusukan Air Jeruk

Sama seperti buah lain, jeruk akan mengalami pembusukan karena adanya kerusakan struktur. Faktor kerusakan biologis biasanya disebabkan oleh mikroorganisme. Mikroorganisme yang berperan adalah jamur dan bakteri. Jamur akan menyerang pada bagian kulit jeruk. Sementara bakteri menyerang pada bagian daging buah. Aktivitas bakteri ini menyebabkan daging buah menjadi busuk, kemudian mengeluarkan bau busuk.

1. Perubahan warna

Perubahan warna terjadi diakibatkan karena aktivitas mikroba penghasil pigmen yang banyak terdapat di permukaan bahan pangan. Beberapa mikroba tersebut antara lain adalah *Serratia mercenscens* dan *Rhodotorulla* (penyebab warna merah), *Penicillium* (penyebab warna hijau), *Pseudomonas fluorescens* (penyebab warna hijau dengan fluorescens), dan *Aspergillus niger* (penyebab warna hitam). Selain aktivitas mikroba, perubahan warna juga dapat terjadi akibat reaksi pencoklatan (*browning*) secara enzimatis maupun *non-enzimatis* [10].

2. Perubahan aroma

Perubahan aroma timbul karena terbentuknya senyawa-senyawa volatil (senyawa ringan yang mudah menguap), seperti amonia, H₂S, indol, dan senyawa-senyawa amin (seperti diamin kadaverin dan putresin), oleh aktivitas dari mikroba. Selain itu, beberapa proses lainnya yang turut mempengaruhi adalah proses oksidasi asam lemak tidak jenuh (ketengikan), seperti pada minyak goreng dan dekomposisi protein menjadi peptida (asam amino) [10].

2.5. Pembusukan Air Kelapa

Air kelapa memiliki komposisi kimia seperti protein, lemak, hidrat arang, vitamin C, vitamin B kompleks, kalsium dan mineral yang sangat baik untuk tubuh manusia. Komposisi kimia air kelapa adalah gula 2,56%, abu 0,46%, bahan padat 4,71%, minyak 0,74%, protein 0,55%, dan senyawa klorida 0,17%. Kandungan mineral kalium pada air kelapa juga

sangat tinggi yaitu 203,70 mg/100 g pada air kelapa muda dan 257,52 mg/100 g air kelapa tua [11].

Sifat kimia air kelapa ditentukan oleh nilai pH, keasaman total dan gula reduksi. Derajat keasaman atau pH digunakan untuk menyatakan tingkat keasaman yang dimiliki oleh suatu larutan. Air kelapa memiliki pH 4,5 – 5,3 per 100 ml air kelapa. Asam - asam organik yang terdapat pada air kelapa dapat mempengaruhi perubahan pH air kelapa. Komposisi gula reduksi air kelapa yaitu sekitar 1,7 – 2,6 %. Pada air kelapa terdapat gula yaitu sukrosa, glukosa dan fruktosa [12]. Hasil penelitian menunjukkan bahwa selama penyimpanan air kelapa pada suhu dingin di dalam refrigerator dapat menghambat turunnya kadar gula reduksi maupun pH air kelapa. Kadar gula reduksi air kelapa pada penelitian ini digunakan sebagai indikator terjadinya fermentasi, karena gula akan difermentasi oleh mikrobia yang menghasilkan asam dan alkohol, bila terjadi fermentasi maka kadar gula reduksi menurun dan total asam tinggi serta pH akan menjadi sangat asam. Ini dapat menggambarkan kerusakan sifat kimia air kelapa selama penyimpanan [13].

2.6. Sensor

Sensor adalah komponen elektronika yang berfungsi untuk mengubah besaran mekanis, magnetis, panas, sinar, dan kimia menjadi besaran listrik berupa tegangan, resistansi dan arus listrik. Sensor sering digunakan untuk pendeteksian pada saat melakukan pengukuran atau pengendalian. Pada tugas akhir ini, menggunakan jenis sensor yang ditunjukkan pada tabel 2.1 :

Tabel 2.1. Tipe sensor yang dipakai

No	Tipe	Jenis Sensor	Target Pengukuran
1	SEN0161	Potensiometrik	pH Meter
2	SEN0244	Konduktivitas Listrik	Total Zat Padat Terlarut (TDS)
3	SEN0189	Optik	Kekeruhan

2.6.1. Sensor Potensiometrik

Potensiometrik adalah suatu metode analitik pada ilmu kimia yang paling sering digunakan untuk analisis kimia, dimana cara kerjanya adalah pengukuran perubahan potensial dari elektroda untuk mengetahui

konsentrasi dari suatu larutan. Metode ini dapat langsung menentukan konsentrasi suatu ion dalam suatu larutan [14]. Besarnya potensial elektroda ini tergantung pada kepekatan ion-ion tertentu dalam larutan, atau bergantung pada bagaimana sejumlah analit terdistribusi/menyebar melintasi antarmuka dan berhubungan dengan aktivitas analit yang mengalami reaksi dalam sel [15].

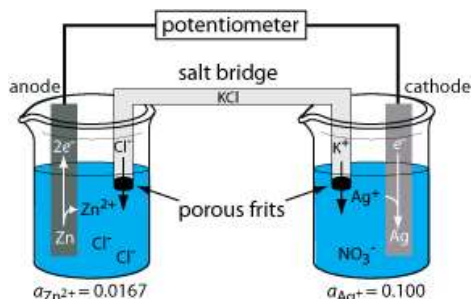
Pada larutan encer, maka aktivitas zat dianggap sama dengan konsentrasi zat dalam larutan sehingga besarnya potensial dapat dituliskan dalam persamaan :

$$E = E^\circ + \frac{RT}{nF} \ln [\text{ox}]^a / [\text{red}]^b \quad (2.1)$$

Hubungan antara potensial suatu elektroda dengan selektivitas ion dalam larutan dinyatakan dalam Persamaan Nernst [14] :

$$\text{pH} = E^\circ - E / (2,303 RT/F) \quad (2.2)$$

Dimana : E adalah potensial terukur, E° adalah potensial electrode standar, R adalah tatapan gas, T adalah suhu (kelvin), F adalah tetapan faraday, dan n adalah jumlah elektron yang di *Transfer*.



Gambar 2.1. Pengukuran Potensiometrik [14].

- **Elektroda**

Elektroda merupakan bagian dari komponen potensiometrik yang berfungsi sebagai sensor analit yang terdiri dari sebuah penghantar elektronik (misal logam) dan sebuah penghantar ionik (larutan). Dalam pengukuran secara potensiometrik elektroda yang digunakan harus bersifat *inert* sehingga dapat bereaksi dengan analit [16].

- **Elektroda kerja**

Elektroda kerja merupakan elektroda yang potensialnya bergantung pada aktivitas analit. Dua jenis elektroda yang umum digunakan dalam

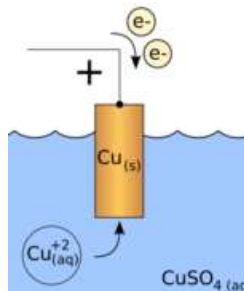
pengukuran secara potensiometrik yaitu elektroda logam dan elektroda membran.

- **Elektroda pembanding**

Elektroda pembanding (*reference*) adalah sebuah elektroda tunggal yang potensialnya diketahui dengan tepat dan dapat digunakan untuk mengukur potensial sebuah elektroda lain.

- **Anoda**

Anoda adalah elektroda, bisa berupa logam maupun penghantar listrik lain, pada sel elektrokimia yang terpolarisasi jika arus listrik mengalir ke dalamnya. Arus listrik mengalir berlawanan dengan arah pergerakan elektron. Pada proses elektrokimia, baik sel galvanik (baterai) maupun sel elektrolisis, anoda mengalami oksidasi (Pelepasan elektron oleh sebuah molekul, atom, atau ion).

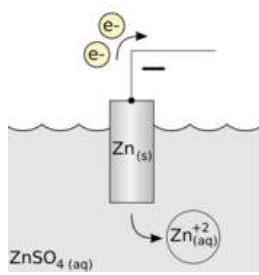


Gambar 2.2. Proses Oksidasi Pada Anoda [16].

Pada sel galvanik atau pembangkit listrik (baterai), anoda adalah kutub negatif. Elektroda akan melepaskan elektron menuju ke sirkuit dan karenanya arus listrik mengalir ke dalam elektroda ini dan menjadikannya anoda dan berkutub negatif. Dalam sel galvanik, reaksi oksidasi terjadi secara spontan. Karena terus menerus melepaskan elektron anoda cenderung menjadi bermuatan positif dan menarik anion dari larutan (elektrolit) serta menjauhkan kation.

- **Katoda**

Kebalikan dari anoda, katoda adalah elektroda dalam sel elektrokimia yang terpolarisasi jika arus listrik mengalir keluar darinya. Katoda merupakan elektroda dengan potensial negatif terhadap anoda. Katoda dalam hal ini mengalami reduksi (penambahan elektron oleh sebuah molekul, atom atau ion)



Gambar 2.3. Proses Reduksi Pada Katoda [16].

pH merupakan cerminan derajat keasaman yang diukur dari jumlah ion hidrogen menggunakan rumus $\text{pH} = -\log(\text{H}^+)$. Air murni terdiri dari ion H⁺ dan ion OH⁻ dalam jumlah berimbang pH air murni adalah 7. Makin banyak ion OH⁻ dalam cairan makin rendah ion H⁺ dan makin tinggi pH dan cairan demikian bersifat basa. Sebaliknya, makin banyak H⁺ makin rendah OH⁻ dan cairan tersebut bersifat asam [16].



Gambar 2.4. Pengukuran pH [16].

Pada gambar 3.2.1.1.d. menunjukkan perbandingan 2 elektroda. Elektroda indikator dengan nilai muatan H⁺ tetap dibandingkan nilai muatan larutan yang berada pada pembanding yang berubah-ubah nilai muatannya. Secara umum ada 3 bagian yang terdapat pada peralatan yang digunakan dalam sensor potensiometri. Ketiga bagian tersebut adalah:

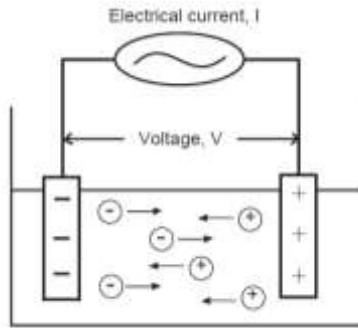
- Elektroda Pembanding (*Reference Electrode*)
- Elektroda Indikator (*Indicator Electrode*)
- Alat Ukur Potensial (*Potensiostat*)

2.6.2. Sensor Konduktivitas Listrik

Konduktivitas adalah ukuran dari kemampuan suatu bahan untuk membangkitkan arus listrik. Kemampuan dalam membangkitkan arus listrik tergantung dari beberapa faktor yakni :

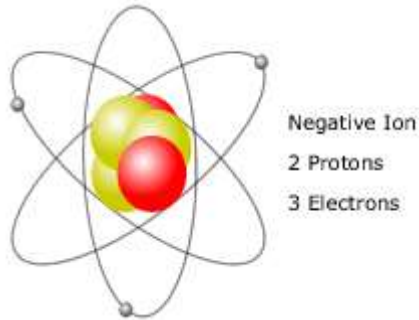
- Konsentrasi
- Mobilitas ion
- Valensi ion
- Temperatur

Konduktivitas dapat diukur dengan menerapkan arus listrik bolak balik (I) ke kedua elektroda yang direndam dalam larutan dan mengukur nilai tegangan (V). Selama proses ini, kation (ion positif) bermigrasi ke negatif elektroda, anion (ion negatif) ke elektroda positif dan larutan bertindak sebagai konduktor listrik



Gambar 2.5. Pergerakan Ion Dalam Larutan [17].

Konduktivitas biasanya diukur dalam larutan elektrolit. Elektrolit adalah zat yang mengandung larutan ion, yaitu larutan garam ionic atau senyawa yang terionisasi dalam larutan. Ion yang terbentuk didalam larutan bertanggungjawab sebagai pengantar arus listrik. Elektrolit yang kuat adalah zat yang terionisasi penuh dalam larutan sedangkan elektrolit yang lemah adalah zat yang tidak terionisasi sepenuhnya dalam larutan. Ion merupakan atom atau gabungan beberapa atom yang mempunyai muatan listrik positif atau negatif. Atom atau kumpulan atom yang memiliki muatan listrik positif disebut ion positif atau kation. Sedangkan, yang bermuatan listrik negatif disebut ion negatif atau anion. asam merupakan suatu zat yang dapat memberi proton (ion H⁺) kepada zat lain.



Gambar 2.6. Struktur Ion [17].

Pada gambar 2.6. Elektron yang mengelilingi inti atom terus bergerak sambil berputar pada sumbu. Akan tetapi, elektron dapat meninggalkan atom karena sesuatu hal, seperti pemanasan, medan listrik, dan medan magnet. Elektron yang keluar dari suatu atom dapat masuk ke atom lainnya. Akibatnya, atom yang kehilangan elektron akan menjadi atom yang bermuatan listrik positif karena jumlah proton menjadi lebih besar daripada jumlah elektronnya, sedangkan atom yang kedatangan elektron menjadi atom bermuatan listrik negatif karena jumlah elektronnya melebihi jumlah protonnya [17].

Salah satu faktor penting dalam menentukan kelayakan minuman untuk dikonsumsi manusia adalah kandungan Total zat padat terlarut (*Total Dissolved Solids*, sering disingkat dengan TDS) yaitu ukuran zat terlarut (baik itu zat organik maupun anorganik, mis : garam, dll) yang terdapat pada sebuah larutan. TDS meter menggambarkan jumlah zat terlarut dalam *Part Per Million* (PPM) atau sama dengan milligram per Liter (mg/L). Konduktivitas listrik dari minuman terkait secara langsung dengan konsentrasi zat-zat padat terionisasi terlarut dalam air. Ion-ion dari zat-zat padat terlarut dalam air memberikan kemampuan bagi air untuk menghantarkan arus listrik, yang dapat diukur menggunakan *conductivity meter* konvensional atau TDS meter. Konsentrasi TDS yang terionisasi dalam suatu zat cair mempengaruhi konduktivitas listrik zat cair tersebut. Makin tinggi konsentrasi TDS yang terionisasi dalam air, makin besar konduktivitas listrik larutan tersebut. Prinsip kerjanya dengan menghubungkan 2 buah probe ke larutan yang diukur, kemudian dengan rangkaian pemrosesan sinyal akan mengeluarkan *output* yang menunjukkan besar konduktivitas/daya hantar listrik sampel air tersebut [18]. Nilai konduktivitas listrik sebuah zat cair menjadi referensi atas jumlah ion serta konsentrasi padatan (*Total Dissolved Solids* / TDS) yang

terlarut di dalamnya. Konsentrasi ion di dalam larutan berbanding lurus dengan daya hantar listriknya. Semakin banyak ion mineral yang terlarut, maka akan semakin besar kemampuan larutan tersebut untuk menghantarkan listrik. Sifat kimia inilah yang digunakan sebagai prinsip kerja *conductivity meter*. Teori tentang konduktometri adalah merupakan kebalikan dari teori hukum ohm. Maka disusunlah teori tentang konduktivitas yang merupakan kebalikan dari resistivitas.

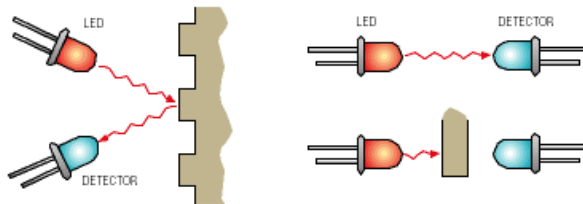
$$G = \frac{I}{R} \quad (2.3)$$

Dimana : G adalah konduktivitas (mho) atau (S), I adalah panjang material (meter), R adalah resistansi (ohm).

Sebuah sistem *conductivity meter* tersusun atas dua elektrode, yang dirangkaikan dengan sumber tegangan serta sebuah ampere meter. Elektroda-elektroda tersebut diatur sehingga memiliki jarak tertentu antara keduanya (biasanya 1 cm). Pada saat pengukuran, kedua elektroda ini dicelupkan ke dalam sampel larutan dan diberi tegangan dengan besar tertentu. Nilai arus listrik yang dibaca oleh amperemeter, digunakan lebih lanjut untuk menghitung nilai konduktivitas listrik larutan [19].

2.6.3. Sensor Optik

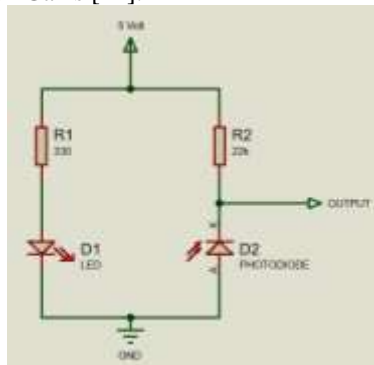
Sensor Optik adalah piranti masukan suatu sistem kendali otomatis yang dibuat dengan komponen optikal yang berfungsi untuk menangkap/mengumpulkan informasi mengenai kondisi lingkungan di sekitar sensor dengan bantuan cahaya.



Gambar 2.7. Cara Kerja Sensor Optik [21].

Pada sensor *optic* diatas mengadopsi prinsip kerja fotodiode, yaitu hanya mengalirkan 1 arah saja. Proses pemantulan akan merubah besaran konduktivitas. Berdasarkan banyaknya cahaya yang di terima, semakin besar intensitas cahaya yang dia terima maka akan semakin besar pula nilai konduktasinya [20]. Fotodiode adalah suatu jenis diode yang resistansinya berubah-ubah kalau cahaya yang jatuh pada diode berubahubah intensitasnya. Dalam gelap nilai tahanannya sangat besar

hingga praktis tidak ada arus yang mengalir. Semakin kuat cahaya yang jatuh pada dioda maka makin kecil nilai tahanannya, sehingga arus yang mengalir semakin besar. Jika fotodioda persambungan p-n bertegangan balik disinari, maka arus akan berubah secara linier dengan kenaikan fluks cahaya yang dikenakan pada persambungan tersebut. Fotodioda terbuat dari bahan semikonduktor. Biasanya yang dipakai adalah *silicon* (Si) atau *gallium arsenide* (GaAs), dan lain-lain termasuk *indium antimonide* (InSb), *indium arsenide* (InAs), *lead selenide* (PbSe), dan timah *sulfide* (PBS). Bahan-bahan ini menyerap cahaya melalui karakteristik jangkauan panjang gelombang, misalnya: 250 nm ke 1100 untuk nm silicon, dan 800 nm ke 2,0 µm untuk GaAs [21].



Gambar 2.8. Rangkaian Elektrik Sensor Fotodioda

Seperti yang terlihat pada gambar 2.8. merupakan dasar dari sensor fotodioda, Pada kondisi awal LED sebagai *Transmitter*, cahaya akan menyinari (tanpa penghalang) fotodioda sebagai *receiver* sehingga nilai resistansi pada sensor fotodioda akan minimum dengan kata lain nilai V_{out} akan mendekati nilai rendah. Sedangkan pada kondisi terhalang, cahaya dari LED tidak dapat diterima oleh fotodioda (*receiver*) sehingga nilai resistansi R1 maksimum, sehinggann nilai dari V_{out} akan mendekati V_{cc} . Adapun rumus perhitungan nilai resistansi dari fotodioda tersebut yaitu :

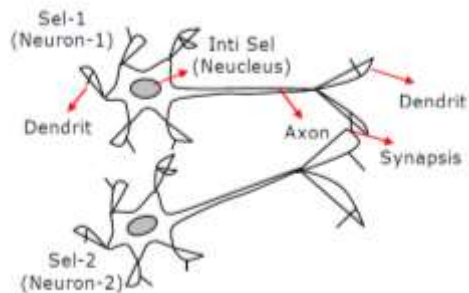
$$V_{out} = \frac{R_{fotodioda}}{R_{fotodioda} + R_2} \times V_{in} \quad (2.4)$$

Dimana : V_{in} adalah tegangan masukan pada rangkaian sensor fotodioda, V_{out} adalah tegangan keluaran pada rangkaian sensor fotodioda,

Rfotodiode adalah resistansi dari fotodiode, R2 adalah resistansi resistor pada rangkaian sensor fotodiode.

2.7. Metode Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran.



Gambar 2.9. Susunan Syaraf Manusia [22].

Gambar 2.9. menunjukkan susunan syaraf pada manusia. Setiap sel syaraf (*Neuron*) akan memiliki satu inti sel, inti sel ini nanti yang akan melakukan pemrosesan informasi. Informasi yang datang akan diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai *axon* sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini akan menjadi masukan bagi *Neuron* lain yang mana antar dendrit kedua sel tersebut dipertemukan dengan sinapsis. Informasi yang dikirimkan antar *Neuron* ini berupa rangsangan yang dilewatkan melalui dendrit. Informasi yang datang dan diterima oleh dendrit akan dijumlahkan dan dikirim melalui *axon* ke dendrit akhir yang bersentuhan dengan dendrit dari *Neuron* yang lain. Informasi ini akan diterima oleh *Neuron* lain jika memenuhi batasan tertentu, yang sering dikenal dengan nama nilai ambang (*threshold*). Pada kasus ini, *Neuron* tersebut dikatakan teraktivasi. Hubungan antar *Neuron* terjadi secara adaptif, artinya struktur hubungan tersebut terjadi secara dinamis. Otak manusia selalu memiliki kemampuan untuk belajar dengan melakukan adaptasi [22].

2.8. Komponen Jaringan Syaraf Tiruan

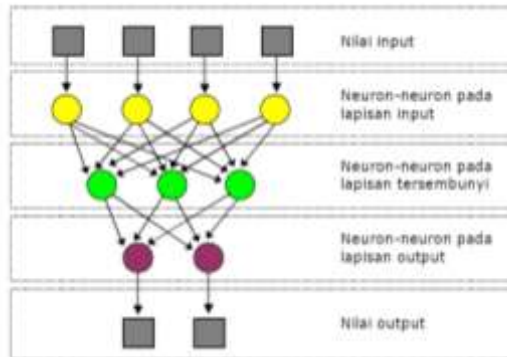
Ada beberapa tipe jaringan syaraf, namun demikian, hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri-dari beberapa neuron, dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Gambar 2.10. menunjukkan struktur neuron pada jaringan syaraf [22].



Gambar 2.10. Struktur Neuron Jaringan Syaraf Tiruan [22].

Jika kita lihat, neuron buatan ini sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron-neuron biologis. Informasi (disebut dengan: *input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya. Demikian seterusnya. Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (neuron *layers*). Biasanya neuron neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*). Tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan

secara mundur pada jaringan. Gambar 2.11. menunjukkan jaringan syaraf dengan 3 lapisan [22].



Gambar 2.11. Jaringan Syaraf Dengan 3 Lapisan [22].

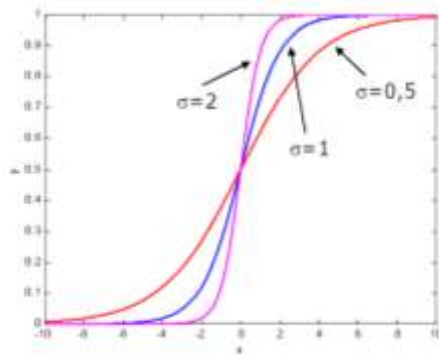
2.9. Fungsi Aktivasi

- **Fungsi Sigmoid Biner.**

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation*. Fungsi sigmoid biner memiliki nilai pada *range* 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1 [22]. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai *output*nya 0 atau 1 (Gambar 2.3.1). Fungsi sigmoid biner dirumuskan sebagai:

$$y = f(x) = \frac{1}{1+e^{-\sigma x}} \quad (2.5)$$

Dengan $f'(x) = \sigma f(x)[1 - f(x)]$ (2.6)



Gambar 2.12. Fungsi Aktivasi : Sigmoid Biner [22].

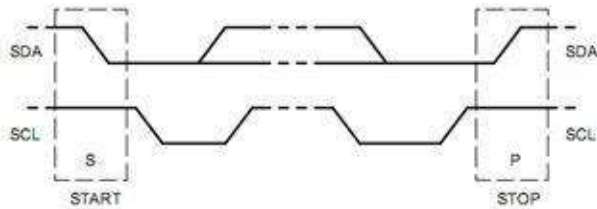
2.10. I2C (*Inter Integrated Circuit*)

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya.



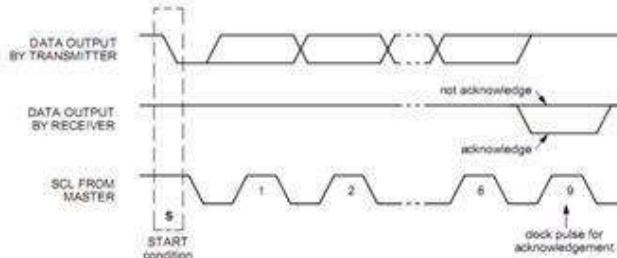
Gambar 2.13. Modul *Inter Integrated Circuit I* [23].

Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai *Transfer* data pada I2C Bus dengan membentuk sinyal *Start*, mengakhiri *Transfer* data dengan membentuk sinyal *Stop*, dan membangkitkan sinyal clock. *Slave* adalah piranti yang dialamati *Master*. Sinyal *Start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “1” menjadi “0” pada saat SCL “1”. Sinyal *Stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1” [23]. Kondisi sinyal *Start* dan sinyal *Stop* seperti tampak pada Gambar 2.14.



Gambar 2.14. Kondisi sinyal *Start* dan *Stop* [23].

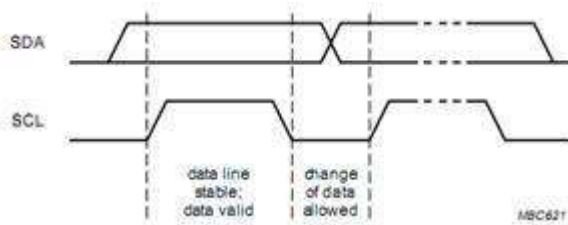
Sinyal dasar yang lain dalam I2C Bus adalah sinyal *acknowledge* yang disimbolkan dengan ACK. Setelah *Transfer* data oleh *Master* berhasil diterima *slave*, *slave* akan menjawabnya dengan mengirim sinyal *acknowledge*, yaitu dengan membuat SDA menjadi “0” selama siklus *clock* ke 9. Ini menunjukkan bahwa *Slave* telah menerima 8 bit data dari *Master*. Kondisi sinyal *acknowledge* seperti tampak pada Gambar 2.15.



Gambar 2.15. Sinyal ACK dan NACK [23].

Dalam melakukan *Transfer* data pada I2C Bus, kita harus mengikuti tata cara yang telah ditetapkan yaitu:

- *Transfer* data hanya dapat dilakukan ketika Bus tidak dalam keadaan sibuk.
- Selama proses *Transfer* data, keadaan data pada SDA harus stabil selama SCL dalam keadaan tinggi. Keadaan perubahan “1” atau “0” pada SDA hanya dapat dilakukan selama SCL dalam keadaan rendah. Jika terjadi perubahan keadaan SDA pada saat SCL dalam keadaan tinggi, maka perubahan itu dianggap sebagai sinyal *Start* atau sinyal *Stop*.



Gambar 2.16. Trasfer Bit pada I2C bus [23].

2.11. LCD (*Liquid Crystal Display*)

LCD (*Liquid Crystal Display*) adalah salah satu komponen elektronika yang berfungsi sebagai tampilan suati data, baik karakter huruf ataupun grafik. Dipasaran tampilan LCD sudah tersedia dalam bentuk modul ataupun tampilan LCD beserta rangkaian pendukungnya.

LCD mempunyai pin data, *control* catu daya, dan pengatur kontras tampilan. LCD juga merupakan perangkat *display* yang paling umum dipasangkan pada mikrokontroller, mengingat ukurannya yang kecil dan kemampuannya menampilkan karakter atau grafik yang lebih jika dibandingkan dengan seven segment. Pada pengembangan *system embedded*, LCD merupakan komponen penting dalam pengaplikasiannya yang diperlukan sebagai penampil informasi utama, misalnya alat pengukur pH meter, penampil waktu jam, menit dan detik dll. Register yang terdapat pada LCD adalah sebagai berikut :

- IR (*Intruccion register*)

Digunakan untuk menentukan fungsi yang harus dikerjakan oleh LCD serta pengalamatan DDRAM atau CGRAM

- DR (*Data Register*)

Digunakan sebagai tempat data DDRAM atau CGRAM yang akan ditulis atau dibaca oleh *computer* atau sistem minimum. Saat dibaca DR menyimpan data DDRAM atau CGRAM, setelah itu data alamatnya secara otomatis masuk ke DR. pada waktu menulis, cukup lakukan inialisasi DDRAM atau CGRAM, kemudian untuk selanjutnya data dituliskan ke DDRAM atau CGRAM sejak awal alamat tersebut.

- BF (*Busy Flag*)

Digunakan untuk mengetahui apakah LCD dalam keadaan siap atau sibuk. Apabila LCD sedang melakukan operasi internal, BF diset menjadi 1 sehingga tidak akan menerima perintah dari luar. Jadi, BF harus dicek apakah telah *dirreset* menjadi 0 ketika akan menulis LCD (memberi data pada LCD). Cara untuk menulis LCD adalah dengan mengeset RS menjadi 0 dan mengeset R/W menjadi 1

- AC (Address Counter)

Digunakan untuk menunjukkan alamat pada DDRAM atau CGRAM dibaca atau ditulis, maka AC secara otomatis menunjukkan alamat berikutnya. Alamat yang disimpan AC dapat dibaca bersamaan dengan BF

- DDRAM (*Display Data Random Access Memory*)

Digunakan sebagai tempat penyimpanan data yang sebesar 80 byte atau 80 karakter. AC menunjukkan alamat karakter yang sedang ditampilkan.

- CGROM (*Character Generator Read Only Memory*)

Pada LCD terdapat ROM untuk menyimpan karakter-karakter ASCII (*American Standard Code for Information Interchange*) sehingga cukup memasukkan kode ASCII untuk menampilkannya.

- CGRAM (*Character Generator Random Access Memory*)

Sebagai data *storage* untuk merancang karakter yang dikehendaki. Untuk CGRAM terdapat kode ASCII dari 00h sampai 0Fh, tetapi hanya tinggi karakter dan 3 bit tinggi menjadi 3 bit rendah DDRAM yang menunjukkan karakter, sedangkan 3 bit rendah sebagai posisi data CGRAM untuk membuat tampilan baris dalam dotmatriks 5x7 karakter tersebut, dimulai dari atas. Sehingga karakter untuk kode ASCII 00h sama dengan 09h sampai 07h dengan 0Fh. Dengan demikian untuk perancangan 1 karakter memerlukan penulisan data ke CGRAM sampai 8 kali.

- *Cursor dan Blink Control Circuit*

Merupakan rangkaian yang menghasilkan tampilan kursor dan kondisi *blink* [24]



Gambar 2.17. Liquid Crystal Display 16x4 [24].

Tabel 2.2. Konfigurasi Pin LCD

No. Pin	Symbol	Function
1	VSS	Ground
2	VDD	Power Supply
3	V0	Power Supply for LDC
4	RS	Select Display Data ("H") or Instruction ("L")
5	R/W	Read or Write Select Signal
6	E	Read/Write Enable Signal
7	DB0	Display Data Signal
8	...	
9	...	
10	DB7	
11	LED - (K)	Please also refer to 6.1 PCB drawing and description
12	LED + (A)	Please also refer to 6.1 PCB drawing and description

2.12. Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor Atmel AVR dan *software*nya memiliki bahasa pemrograman sendiri. Tampilan Arduino dapat dilihat pada gambar 2.18.



Gambar 2.18. Arduino Uno [25].

Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino. Arduino juga menyederhanakan proses

bekerja dengan mikrokontroler, sekaligus menawarkan berbagai macam kelebihan antara lain:

- Perangkat lunak Arduino IDE dipublikasikan sebagai *Open Source*, Bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada Bahasa C untuk AVR.
- Perangkat keras Arduino berbasis mikrokontroler ATMEGA8, ATMEGA168, ATMEGA328 dan ATMEGA1280 (yang terbaru ATMEGA2560).
- Tidak perlu perangkat *chip programmer* karena di dalamnya sudah ada *bootloader* yang akan menangani upload program dari komputer.
- Sudah memiliki sarana komunikasi USB, sehingga pengguna Laptop yang tidak memiliki port serial/RS323 bisa menggunakannya.
- Bahasa pemrograman relatif mudah karena *software* Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
- Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* Arduino. Misalnya *shield* GPS, Ethernet, SD Card, dll.

Untuk membuat program Arduino dan mengupload ke dalam *board* Arduino, dengan menggunakan *software* Arduino IDE (*Integrated Development Enviroment*) yang bisa di *download* gratis [25].

-Halaman Ini Sengaja Dikosongkan-

BAB 3

PERANCANGAN SISTEM

Perancangan *system* meliputi perancangan perangkat keras dan perangkat lunak. Perancangan keras meliputi perancangan mekanik dan elektrik. Perangkat keras berguna mendapatkan data sensor untuk diolah dengan Jaringan Syaraf Tiruan. Sensor yang digunakan merupakan sensor modul dari dfrobot.com. ketiga sensor ini digunakan untuk mengetahui kondisi fisik dan kimiawi dari sebuah minuman. Sampel yang digunakan yakni, Susu kedelai, Susu UHT, Jus Jeruk, *Bear Brand*, dan Air Kelapa merupakan minuman kemasan yang banyak dijual diberbagai *supermarket* amupun pada pedagang kaki lima.

Perangkat lunak meliputi program Arduino IDE dan Jaringan Syaraf Tiruan menggunakan *Software DevC++*. Jaringan syaraf tiruan digunakan untuk menentukan apakah sebuah minuman layak dan tidak layak dari untuk diminum setelah kemasan dibuka beberapa jam. Bab ini menjelaskan secara keseluruhan dan desain tiap-tiap blok yang ada.

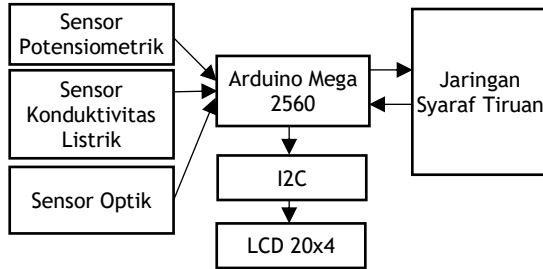
3.1. Diagram Blok Sistem

Secara umum sisitem ini terdiri dari perangkat keras dan perangkat lunak. Perangkat keras meliputi perancangan perangkat elektrik dan perangkat mekanik. Perangkat lunak meliputi program Arduino dan Jaringan Syaraf Tiruan menggunakan *software DevC++*. Sedangkan perangkat meknik meliputi tata letak Sensor, LCD, Arduino, I2C dan kotak dari alat ini.

Sistem kerja dari alat ini adalah ketiga sensor akan mengukur nilai pH, TDS (Total Zat Padat terlarut) dan NTU (*Nephelometric Turbdity Unit*) atau Kekeruhan dan dibaca oleh Arduino Mega kemudian ditampilkan ke layar LCD berukuran 20x4. Dari data pengukuran yang didapat akan diolah menggunakan Jaringan Syaraf Tiruan untuk mengklasifikasikan kondisi minuman. Klasifikasi minuman yang ditentukan adalah Segar, Kurang Segar dan Tidak Segar agar dan kemudian klasifikasi tersebut akan ditampilkan pada LCD 20x4.

3.2. Perancangan Perangkat Keras

Perancangan ini terdiri dari bagian elektrik. Dibagian elektrik terdiri dari Sensor SEN0161, Sensor SEN0244, Sensor SEN0189, Arduino Mega, Modul I2C, LCD 20x4. Diagram blok perancangan elektrik ditunjukkan pada gambar 3.1.



Gambar 3.1. Diagram Blok Perancangan Elektrik

Sedangkan kotak yang digunakan adalah kotak elektronik berwarna hitam yang dilubangi sehingga port dari masing-masing sensor bisa masuk. Gambar kotak ditunjukkan pada gambar 3.2.



Gambar 3.2. Kotak Wadah Rangkaian

3.2.1. Perangkat Elektrik

Perangkat elektrik terdiri dari Sensor SEN0161, Sensor SEN0244, Sensor SEN0189, Arduino Mega, Modul I2C, LCD 20x4.

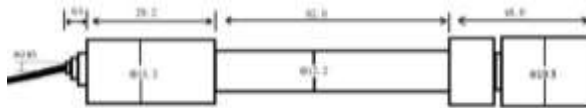
3.2.1.1. Sensor SEN0161

Jenis sensor yang digunakan untuk mengukur nilai pH adalah pH Meter SKU SEN0161.



Gambar 3.3. SEN0161 Sensor Potensiometrik

Sensor SEN0161 adalah salah satu komponen untuk mengukur nilai pH (Derajat Keasaman) suatu larutan. Sensor SEN0161 dirancang khusus untuk pengontrol Arduino yang memiliki koneksi serta fitur yang sederhana. Memiliki LED yang berfungsi sebagai indikator daya, konektor BNC atau pH *probe*. Sensor SEN0161 memiliki dimensi yang ditunjukkan pada gambar 3.4. :



Gambar 3.4. Dimensi SEN0161

Setiap sensor memiliki nilai spesifikasi masing-masing, termasuk Sensor SEN0161. Berikut adalah nilai spesifikasi dari Sensor SEN0161 :

- Spesifikasi
 - Tegangan Modul : 5V
 - Ukuran Modul : 43mm x 32mm
 - Pengukuran : 0-14 pH
 - Pengukuran Suhu : 0-60 °C
 - Akurasi : ± 0.1 pH (25 °C)
 - Respon Waktu : ≤ 1 menit



Gambar 3.5. Modul Sensor SEN0161

Pada gambar 3.5. menunjukkan tampilan modul dari Sensor SEN0161 yang memiliki *port*. Fungsi dari setiap *port* ditunjukkan pada tabel 3.1
Tabel 3.1. Fungsi Dari Masing-Masing Port

<i>Num</i>	<i>Label</i>	<i>Description</i>
1	-	<i>Power GND (0V)</i>
2	+	<i>Power VCC (3.3~5.5V)</i>
3	A	<i>Analog Signal Output(0~3.0V)</i>
4	BNC	<i>pH Probe Connector</i>

3.2.1.2. Sensor SEN0244

Jenis sensor yang digunakan untuk mengukur nilai TDS adalah *Gravity Analog TDS Sensor Meter* SKU SEN0244.

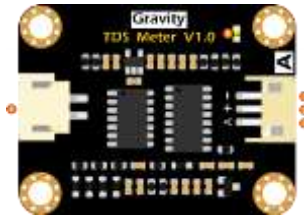


Gambar 3.6. *Gravity Analog TDS Sensor Meter* SKU SEN0244.

TDS sensor adalah peralatan yang banyak digunakan untuk mengukur nilai TDS. Sensor ini mendukung *input* tegangan lebar 3.3 ~ 5.5V, dan *output* tegangan analog 0 ~ 2.3V, yang membuatnya kompatibel dengan sistem atau papan kontrol 5V atau 3.3V. Probe TDS tahan air, dapat direndam dalam air untuk pengukuran waktu yang lama.

- Spesifikasi
 - *Signal Transmitter Board*
 - *Input Voltage*: 3.3 ~ 5.5V
 - *Output Voltage*: 0 ~ 2.3V
 - *Working Current*: 3 ~ 6mA
 - *TDS Measurement Range*: 0 ~ 1000ppm
 - *TDS Measurement Accuracy*: ± 10% F.S. (25 °C)
 - *Module Size*: 42 * 32mm
 - *Module Interface*: PH2.0-3P
 - *Electrode Interface*: XH2.54-2P
 - TDS probe
 - *Number of Needle*: 2
 - *Total Length*: 83cm
 - *Connection Interface*: XH2.54-2P
 - *Colour*: Black
 - *Other*: Waterproof Probe

➤ *Board Overview*



Gambar 3.7. Modul Sensor SEN0244

Tabel 3.2. Konfigurasi Pin Modul Sensor SEN0244

Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.3 ~ 5.5V)
3	A	Analog Signal Output(0 ~ 2.3V)
4	TDS	TDS Probe Connector
5	LED	Power Indicator

3.2.1.3. Sensor SEN0189

Jenis sensor yang digunakan untuk mengukur nilai kekerhan adalah Turbidity Sensor SKU SEN0189.



Gambar 3.8. Turbidity Sensor SKU SEN0189.

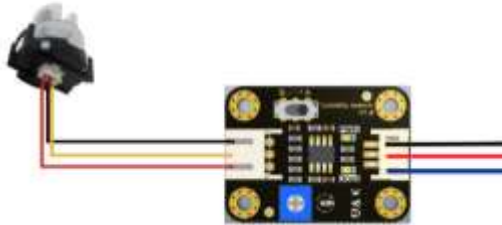
Sensor kekeruhan arduino gravitasi mendeteksi kualitas air dengan mengukur tingkat kekeruhan. Menggunakan cahaya untuk mendeteksi partikel tersuspensi dalam air dengan mengukur transmisi cahaya dan tingkat hamburan, yang berubah dengan jumlah total padatan tersuspensi (TSS) dalam air. Dengan meningkatnya TTS, tingkat kekeruhan cairan meningkat. Sensor cair ini menyediakan mode keluaran

sinyal analog dan digital. Ambang disesuaikan ketika dalam mode sinyal digital. Anda dapat memilih mode sesuai dengan MCU Anda.

➤ Spesifikasi

- *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



Gambar 3.9. Connection Diagram

3.2.1.4. Arduino Mega 2560

Pada tugas akhir ini menggunakan Arduino Mega 2560. Pembacaan hasil pengukuran yang berupa tegangan analog kemudian dikonversi menjadi sinyal digital dengan memanfaatkan ADC *converter* internal milik Arduino Mega 2560 dan diolah menjadi nilai dari masing-masing sensor untuk kemudian dijadikan sebagai *input* dari Jaringan Syaraf Tiruan.

Arduino Mega 2560 adalah papan mikrokontroler berdasarkan ATmega2560. Ini memiliki 54 pin *input / output* digital (dimana 14 dapat digunakan sebagai *output* PWM), 16 *input* analog, 4 UART (*port serial*

perangkat keras), osilator kristal 16 MHz, koneksi USB, colokan listrik, *header ICSP*, Dan tombol *reset*. Ini berisi semua yang dibutuhkan untuk mendukung mikrokontroler; Cukup hubungkan ke komputer dengan kabel USB atau nyalakan dengan adaptor AC-ke-DC atau baterai untuk memulai.



Gambar 3.10. Tampilan Arduino Mega 2560 [25].

Arduino Mega 2560 memiliki spesifikasi sebagai berikut :

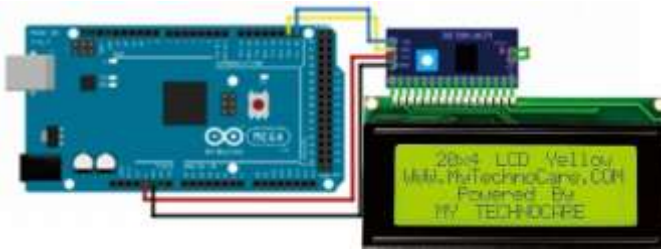
- Mikrokontroler ATmega2560
- Tegangan Operasi 5V
- Tegangan Masukan (disarankan) 7-12V
- Tegangan (batas) 6-20V
- Pin I/O digital 54
- Pin *Input Analog* 16
- Arus DC per I/O Pin 40mA
- Arus DC untuk Pin 3.3 V 50mA
- *Flash Memory* 256 KB
- SRAM 8 KB EEPROM 4 KB
- Kecepatan Jam 16 MHz

Pin daya yang digunakan adalah 5V. Catu daya yang diatur digunakan untuk menyalakan mikrokontroler dan komponen lainnya di papan tulis. Ini bisa datang baik dari VIN melalui *regulator on-board*, atau disediakan oleh USB atau suplai 5V yang diatur lainnya.

ATmega2560 memiliki memori *flash* 256 KB untuk menyimpan kode (dimana 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB dari EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM). I2C: 20 (SDA) dan 21 (SCL). Dukung komunikasi I2C (TWI) menggunakan perpustakaan *Wire* untuk menyederhanakan penggunaan bus I2C

3.2.1.5. Modul Serial I2C LCD 20x4

Mengingat jumlah kaki LCD yang banyak, maka untuk menghemat pemakaian kaki Arduino bisa menggunakan adapter dengan komunikasi I2C. Komunikasi I2C (*Inter-Integrated Circuit*) adalah komunikasi yang hanya membutuhkan 2 jalur komunikasi yang disebut dengan SDA (*serial data*) dan SCL (*serial clock*). Komunikasi I2C juga dikenal dengan istilah *Two-Wire Serial Interface* (TWI). Jadi, jika ada modul yang menggunakan komunikasi TWI, berarti bisa diakses menggunakan I2C.



Gambar 3.11. Arduino Mega 2560 I2C-LCD [25].

LCD 20x4 berfungsi untuk menampilkan data dari ketiga sensor. Pada baris pertama digunakan untuk menampilkan nilai dari pH, baris kedua digunakan untuk menampilkan nilai NTU, baris ketiga digunakan untuk menampilkan nilai TDS, dan baris keempat digunakan untuk menampilkan target dari olahan Jaringan Syaraf Tiruan pada suatu minuman apakah masih Segar, Kurang Segar, dan Tidak Segar. *Supply* tegangan untuk LCD 20x4 sebesar 5 volt yang berasal dari Arduino Mega 2560.

- Spesifikasi
 - LCD 20 karakter dengan 4 baris
 - I2C *Address range* = 0x20 ro 0x27
 - *Operating voltage* = 5Vdc
 - *Contrast* = *adjustable by potentiometer on I2c interface*

3.2.2. Perancangan Letak Posisi Sensor

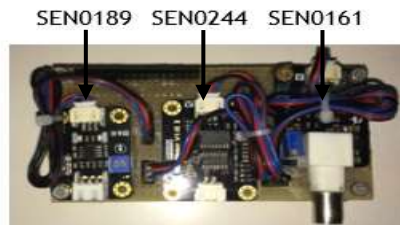
Posisi ketiga sensor ini akan diletakkan pada kotak yang berukuran 12.5mm x 8.5mm x 5mm berwarna hitam. Tujuan dari peletakan sensor

ini adalah untuk mengatur agar supaya terlihat lebih tertata dan dapat melindungi sensor dari kerusakan akibat terjatuh ataupun terkena cairan dari sampel yang diuji. Pada posisi *input* sensor, kotak diberi 3 lubang yang berfungsi sebagai akses jalur *port* dari sensor. Seperti yang terlihat pada gambar 3.12.



Gambar 3.12. Posisi lubang sensor

Untuk posisi sensor sendiri, dipasang secara berurutan seperti yang ditunjukkan pada gambar 3.13.

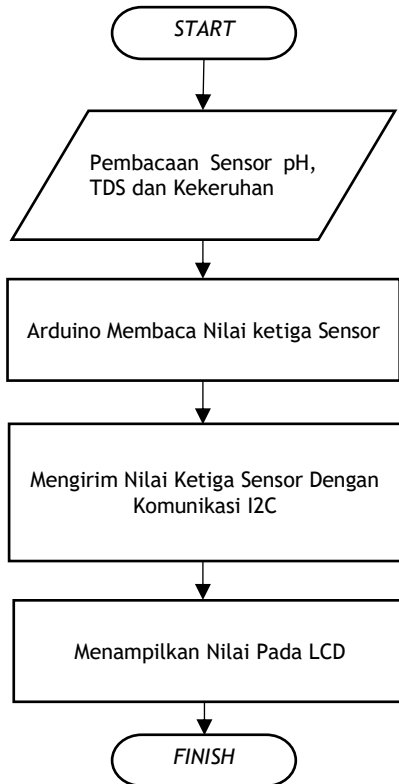


Gambar 3.13. Tata Letak Posisi Sensor

3.3. Perangkat Lunak

Perangkat lunak yang dirancang terdiri atas hanya perangkat lunak pada Arduino digunakan untuk mengukur nilai dari ketiga sensor, melakukan proses *learning* pada Jaringan Syaraf Tiruan, mengolah Jaringan Syaraf Tiruan itu sendiri serta menampilkan hasil pembacaan sensor pada LCD dan klasifikasi target dari *output*.

- Pembacaan Sensor Pada Arduino

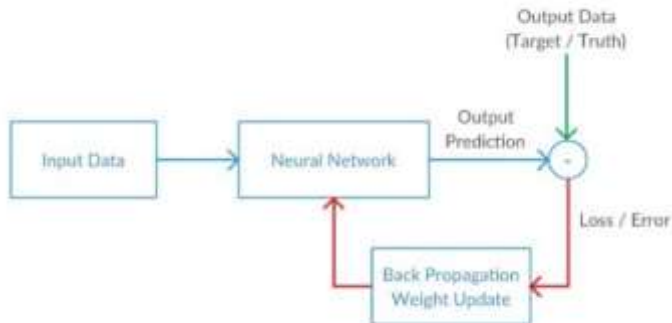


Gambar 3.14. *Flowchart* Pembacaan Nilai Ketiga Sensor

Pada gambar 3.14. Merupakan *flowchart* dari pembacaan ketiga sensor yang digunakan. Tujuannya adalah untuk mengetahui dan mengambil data dari sampel minuman pengukuran untuk kemudian diamati karakteristik dari masing-masing sensor

3.4. Proses Jaringan Syaraf Tiruan

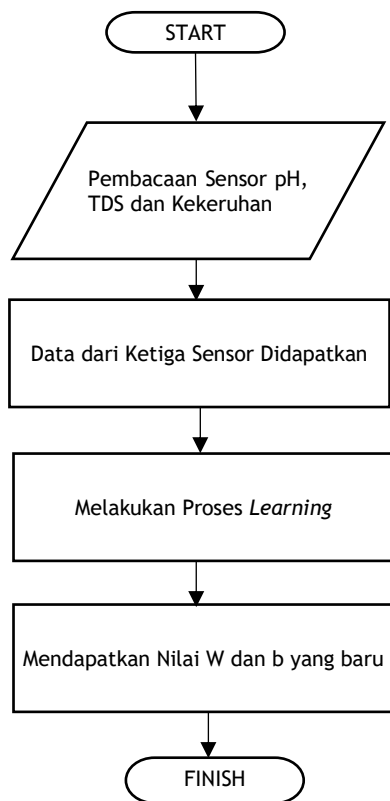
Hal yang penting untuk diperhatikan bahwa minuman yang kemasannya telah terbuka apalagi yang mengandung serat akan cepat basi hal ini dikarenakan adanya aktivitas mikroorganisme didalam minuman tersebut. Proses dari jaringan syaraf tiruan disini untuk menentukan kondisi suatu minuman apakah masih segar untuk diminum setelah beberapa jam kemasannya minuman telah terbuka, kurang segar atau bahkan sudah tidak segar dan tidak layak untuk diminum. Parameter basi sendiri selain ditentukan dari *output* data dari sampel, juga dapat dilihat dari bentuk visualisasinya. Sebuah minuman yang telah rusak atau basi akan menggumpal dan mengeluarkan bau busuk serta kondisi minuman yang agak keruh. Hal ini dapat dijadikan parameter pemilihan sensor yang digunakan. Setelah sampel data diambil maka proses selanjutnya akan dilakukan proses *learning* atau latihan secara matematis untuk menentukan W (bobot) masing-masing sensor dan B (bias). Proses *learning* menggunakan metode *backpropagation*



Gambar 3.15. Proses *Learning* Jaringan Syaraf Tiruan [22].

Pada sistem jaringan syaraf tiruan ini menggunakan menggunakan metode *forward propagation* dimana nilai bobot yang tepat telah didapatkan dari proses *learning* dari *backpropagation*. Metode ini menggunakan 3 *input layer* dengan masing-masing sensor, *hidden layer* atau layar tersembunyi menggunakan 8 neuron dan 3 *output* dengan kategori masing-masing.

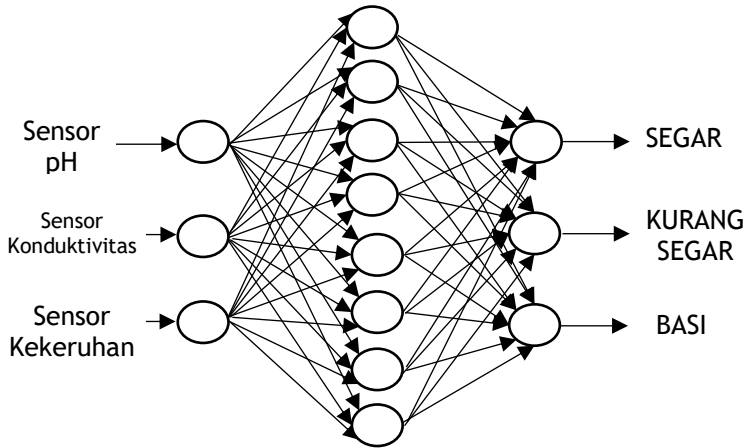
Untuk fungsi aktivasi digunakan fungsi aktivasi sigmoid pada *hidden layer*. Berikut adalah *flowchart* dari *Learning* jaringan syaraf tiruan.



Gambar 3.16. *Flowchart* Proses *Learning* Jaringan Syaraf Tiruan

Cara kerja dari penerapan jaringan syaraf tiruan ini adalah memprediksi kondisi sebuah minuman, jika nilai *output* [0 , 0] maka kondisi minuman masih segar, jika nilai *output* [0 , 1] maka kondisi minuman kurang segar dan jika nilai *output* [1 , 1] maka kondisi minuman basi. Proses identifikasi minuman ini menggunakan *forward pass* yang sebelumnya nilai bobot dan biasanya telah di *update* menggunakan *backpropagation*

Forward pass atau disebut juga forward propagation adalah proses dimana input melewati setiap *neuron* pada *hidden layer* sampai kepada output yang nantinya ditentukan targetnya



Gambar 3.17. Proses Jaringan Syaraf Tiruan

Proses identifikasi ini menggunakan *forward pass* dimana input menuju hidden layer (*dotj*) yang dilakukan dengan menggunakan perkalian (*dot product*) dan penjumlahan matriks antara input, bobot dan bias

$$dotj = \sum_i^3 WjiXi + bj \quad (3.1)$$

$$out = \frac{1}{1+e^{-dotj}} \quad (3.2)$$

Persamaan diatas adalah forward pass dengan menggunakan sigmoid sebagai fungsi aktivasi. Dimana *I* adalah *node* pada *input layer* (3 input), *j* adalah *node* pada hidden layer.

-Halaman Ini Sengaja Dikosongkan-

BAB 4

PENGUJIAN DAN ANALISIS

Pengujian kualitas minuman ini dibagi menjadi beberapa tahap bagian, dimulai dari pengujian perangkat keras dan perangkat lunak. Tujuan dari pengujian keseluruhan sistem ini untuk mendapatkan data nilai dari pH, TDS (*Total Dissolved Solid*) dan NTU (*Nephelometric Turbidity Unit*) kemudian mengklasifikasikan kondisi minuman dengan menggunakan metode jaringan syaraf tiruan.

4.1. Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan dengan menguji suatu sampel minuman dengan menggunakan sensor dan menampilkan hasil pengujian tersebut pada *display* LCD. Beberapa pengujian sensor dapat dilihat pada uraian berikut :

4.1.1. Sensor SEN0161

Sebelum melakukan pengujian pada Sensor SEN0161 hal yang perlu diperhatikan adalah melakukan proses pengkalibrasian pada sensor sebelum dan setelah pengukuran hal ini agar menjaga hasil pengukuran tetap optimal. Alat ukur akan mengalami pergeseran hasil pengukuran dalam selang waktu tertentu dikarenakan penggunaan dan kondisi lingkungan atau bahkan internal alat ukur. Hal ini dapat berpengaruh terhadap hasil pengukuran sensor. Dalam melakukan kalibrasi bahan yang digunakan adalah pH *Buffer* dengan nilai 7 (netral) dan pH *Buffer* dengan nilai 4 (asam) serta aquades sebagai cairan pembersih elektroda sensor. Dari data hasil kalibrasi didapatkan nilai yang ditunjukkan pada tabel 4.1. dan tabel 4.2.



Gambar 4.1. Kalibrasi Dengan pH *Buffer* 7

Tabel 4.1. Kalibrasi dengan pH Buffer 7

pH Buffer	Volt	pH	Volt	Error (pH)	Error (Volt)	Akurasi (%)
7	2	6,89	1,97	0,11	0,03	98,43
7	2	6,89	1,97	0,11	0,03	98,43
7	2	6,89	1,97	0,11	0,03	98,43
7	2	6,9	1,97	0,1	0,03	98,57
7	2	6,9	1,97	0,1	0,03	98,57
7	2	6,9	1,97	0,1	0,03	98,57
7	2	6,9	1,97	0,1	0,03	98,57
7	2	6,91	1,97	0,09	0,03	98,71



Gambar 4.2. Kalibrasi Dengan pH Buffer 4

Tabel 4.2. Kalibrasi dengan pH Buffer 4

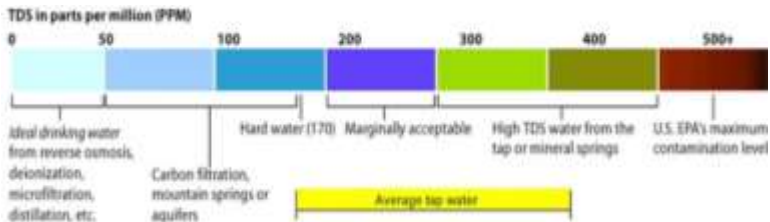
pH Buffer	Volt	pH	Volt	Error (pH)	Error (Volt)	Akurasi (%)
4	1,14	6,32	1,8	2,32	0,66	63,29
4	1,14	6,32	1,8	2,32	0,66	63,29
4	1,14	6,32	1,8	2,32	0,66	63,29
4	1,14	6,32	1,8	2,32	0,66	63,29
4	1,14	6,33	1,81	2,33	0,67	63,19
4	1,14	6,34	1,81	2,34	0,67	63,09
4	1,14	6,34	1,81	2,34	0,67	63,09
4	1,14	6,35	1,81	2,35	0,67	62,99
4	1,14	6,35	1,81	2,35	0,67	62,99

Kalibrasi pada dasarnya membandingkan *output* hasil pengukuran yang dilakukan oleh sensor dengan cairan *Buffer* yang hasilnya sangat mendekati nilainya. Pada pengujian pH *Buffer* 7 didapatkan hasil pengukuran nilai yang mendekati dengan nilai pH *Buffernya* yakni dengan *range* 6,89-6,91 dengan tingkat keakurasian mencapai nilai 98% hal berbeda yang terjadi ketika mengukur pada pH *Buffer* 4. Pada pengujian pH *Buffer* 4 terdapat selisih yang sangat jauh bila dibandingkan dengan pH *Buffer* 7.

Pada pH *Buffer* 4 didapatkan nilai dengan *range* 6,32-6,35 dan dengan tingkat keakurasian mencapai 62%. Sehingga perlu dilakukan proses kalibrasi ulang pada modul sensor SEN0161. Sehingga diputuskan untuk mengambil acuan pada pH *Buffer* 7 sebagai bahan untuk mengkalibrasi pengukuran.

4.1.2. Sensor SEN0244

Hal yang perlu dilakukan pada Sensor SEN0244 adalah menentukan nilai acuan atau referensi dari pengukuran zat padat terlarut. Pada pengujian Sensor SEN0244 hal yang menjadi acuan dalam pengukuran minuman adalah air putih. Dimana air putih memiliki nilai TDS yang rendah. Hal ini mengacu pada gambar 4.3.



Gambar 4.3. Kandungan TDS Dalam Air

Menurut WHO (*World Health Organization*) standar kandungan padatan terlarut dalam air minum terbagi menjadi beberapa kriteria level. Level yang bagus sekali berada pada nilai yang kurang dari 300, level yang baik berada pada nilai 300 sampai dengan 600, level yang bisa diminum berada pada nilai 600 sampai dengan 900, level buruk dan berbahaya berada pada nilai 900 sampai dengan 1200 dan lebih dari 1200 [26]. Untuk lebih jelasnya ditunjukkan pada tabel 4.3

Tabel 4.3. Kandungan TDS dalam Air Menurut WHO

Kandungan TDS (mg/l)	Penilaian
< 300	Bagus Sekali
300-600	Baik
600-900	Bisa diminum
900-1200	Buruk
>1200	Bebahaya

Maka dari tabel 4.3. dijadikan acuan pengukuran pada Sensor SEN0244.



Gambar 4.4. Pengujian Sensor Konduktivitas Listrik

Pada pengukuran konduktivitas listrik didapatkan hasil pengukuran air yang diambil dari berbagai sumber. Maka hasil yang didapat dapat dilihat pada tabel 4.4.

Tabel 4.4. Data Pengukuran Kandungan TDS dalam Air

Jenis Air	TDS	mVolt
Galon (Minum)	119,9	302
Keran	251,19	649
Aquades	0	0
Addes (Minum)	32,74	78

Dari data yang didapat, data pengukuran aquades menunjukkan angka 0 ini dikarenakan kandungan aquades merupakan air hasil destilasi atau penyulingan sama dengan air murni atau H₂O, karena H₂O hampir tidak mengandung mineral. Pada pengukuran air keran, hasil pengukuran menunjukkan angka 251,19. Sedikit lebih tinggi bila dibandingkan dengan

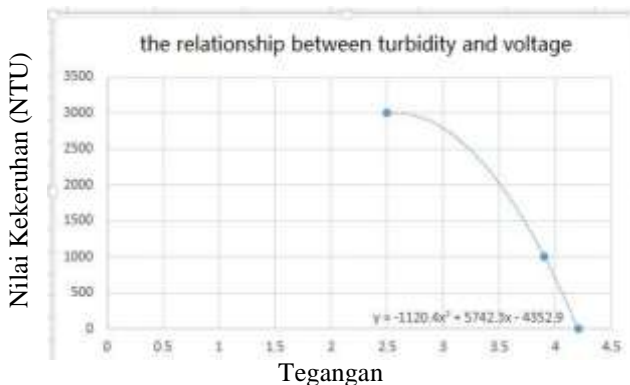
sampel yang lain, akan tetapi masih dalam batas yang sangat aman. Pada pengukuran air minum (galon dan addes) didapatkan hasil pengukuran dengan angka 119,9 dan 32,74. Dari data tersebut terdapat perbedaan dikarenakan proses produksinya yang berbeda, air addes merupakan hasil produksi industri air minum yang dijaga kandungannya namun keduanya masih dalam batas yang sangat aman.

4.1.3. Sensor SEN0198

Pengukuran kekeruhan (turbiditas) didasarkan pada pengukuran intensitas cahaya yang dihamburkan oleh zat-zat tersuspensi dalam air dengan melewati sejumlah cahaya ke dalam air dengan ketebalan tertentu. Banyaknya sinar yang dihamburkan oleh partikel-partikel tersuspensi di ukur dan dinyatakan sebagai kekeruhan dalam air. Semakin tinggi kekeruhan, semakin banyak sinar yang dihamburkan. Sesuai dengan karakteristik Sensor SEN0189, Pengukuran turbiditas mengacu pada persamaan :

$$y = -1120x^2 + 5742,3x - 4352,9 \quad (4. 1)$$

Dari persamaan diatas didapatkan persamaan garis seperti pada gambar 4.1.3.a.



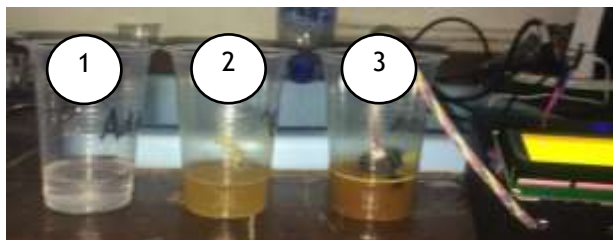
Gambar 4.5. Perbandingan Antara Kekeruhan dan Tegangan
Pengukuran turbiditas mengacu pada nilai tegangan yang terukur, nilai tegangan yang terukur sangatlah berpengaruh pada hasil turbiditasnya. tingkat kekeruhan air disebabkan oleh adanya endapan, partikel organik yang koloid, plankton, dan organisme mikroskopik lainnya [27]. Pada

Peraturan Menteri Kesehatan Republik Indonesia, tingkat kekeruhan pada air maksimal 5 NTU [27].

Tabel 4.5. Pengujian Kekeruhan Air

Sampel	Nilai Kekeruhan (NTU)	Volt
1	-1003.75	4.45
2	497	4.06
3	2463	3.27

Pada data pengujian diatas, terlihat bahwa pada gelas 1 menunjukkan angka -1000.75 hal ini dikarenakan nilai tegangan yang sangat mendekati nilai batas titik potong di sumbu x



Gambar 4.6. Pengujian Sensor Optik

4.2. Pengujian Sampel Minuman

Pengujian sampel minuman yang diambil merupakan minuman yang banyak dijual baik di *supermarket* maupun pada pedagang kaki lima. Guna mempercepat proses pembusukan, maka sampel minuman yang diambil merupakan sampel yang tidak mengandung pengawet makanan atau bahkan mengandung pengawet tetapi dalam jumlah yang sangat sedikit. Sampel minuman antara lain:

1. Susu Kedelai

Tabel 4.6. Pengujian Susu Kedelai

Kondisi	pH			TDS		NTU	
	Kalibrasi	Nilai	Volt	Nilai	Volt	Nilai	volt
Segar	6,95	6,85	2	1422	2.577	2822	2.16
	6,95	6,75	2	1419	2.578	2822	2.16
	6,91	6,7	1,96	1408	2.568	2799	2.13

Kurang Segar	6,69	6,69	1,87	1245	2.412	2822	2.16
	6,97	6,68	1,87	1207	1.382	2817	2.15
	6,95	6,65	1,86	1207	2.387	2817	2.15
Basi	6,95	6,6	1,85	1197	2.382	2817	2.15
	6,94	6,5	1,84	1188	2.373	2817	2.15
	6,89	6,4	1,83	1245	2.407	2813	2.15
	6,92	6,3	1,83	1197	2.368	2813	2.15
	6,83	6,25	1,82	1193	2.373	2817	2.16

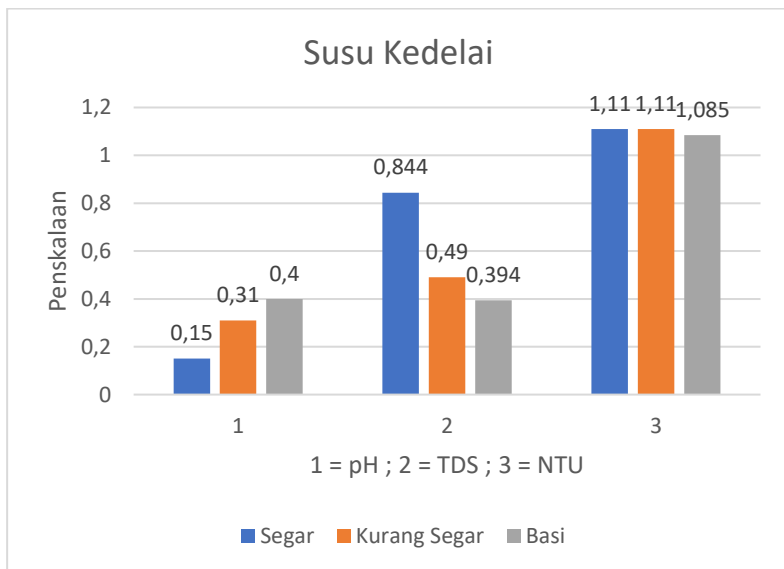
Grafik nilai pembacaan setiap parameter diskalakan, untuk memudahkan dalam melihat grafik pertumbuhan tingkat pembusukan. Dengan persamaan

$$pH = (7 - \text{Nilai pH}) \quad (4.2)$$

$$TDS = ((\text{Nilai TDS}/1000)-1) \times 2 \quad (4.3)$$

$$NTU = (\text{Nilai NTU}/200) - 13 \quad (4.4)$$

Persamaan diatas memiliki *output* dengan selisih yang kecil. Sehingga pembacaan pada proses *learning* dapat lebih mudah dilakukan. Nilai *output* dari persamaan (4.3),(4.4),(4.5) dapat dilihat pada gambar 4.7



Gambar 4.7. Grafik Pengujian Susu Kedelai

Grafik menunjukkan perubahan karakteristik susu kedelai berlangsung cepat, utamanya pada nilai TDS. Pada 13 jam setelah kemasan terbuka, susu kedelai mulai basi dan pada 17 jam setelah kemasan terbuka, susu kedelai telah basi yang ditandai timbulnya bau busuk dan terjadi penggumpalan pada susu kedelai.

2. Susu UHT

Tabel 4.7. Pengujian Susu UHT

Kondisi	pH			TDS		NTU	
	Kalibrasi	Nilai	Volt	Nilai	Volt	Nilai	volt
Segar	6,93	6,95	1,99	1382	2.578	2831	2.17
	6,95	7,03	2,01	1382	2.578	2831	2.17
	6,9	6,94	1,98	1382	2.573	2831	2.16
	6,73	6,79	1,94	1284	2.450	2831	2.17
	6,95	6,82	1,95	1202	2.387	2831	2.17
Kurang Segar	6,94	6,78	1,93	1202	2.387	2835	2.17
	6,96	6,72	1,9	1183	2.373	2835	2.17
	6,93	6,7	1,89	1188	2.368	2835	2.17
Basi	6,9	6,65	1,86	1221	2.407	2831	2.16
	6,93	6,6	1,83	1197	2.377	2831	2.156
	7,15	6,45	1,84	1197	2.373	2886	2.24

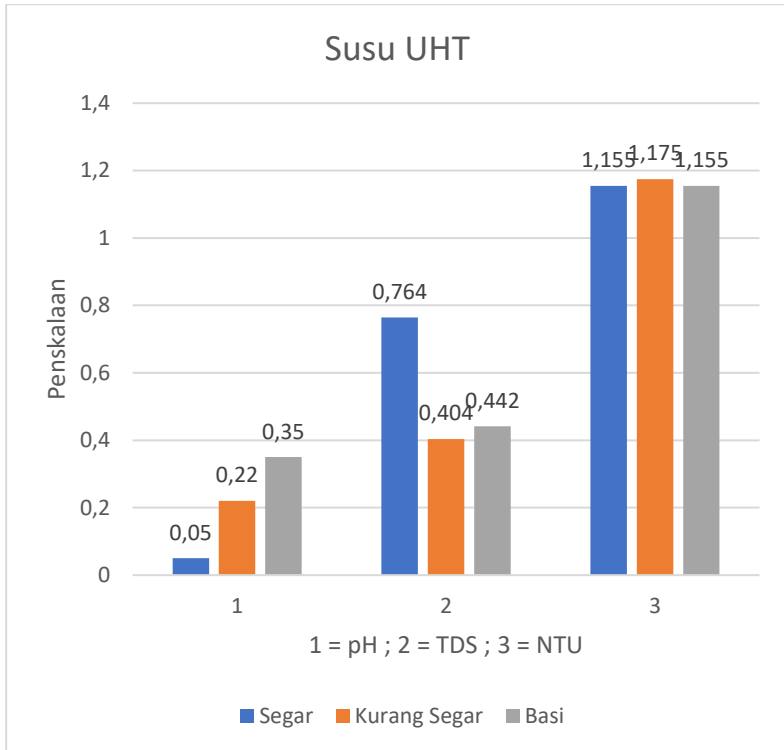
Grafik nilai pembacaan setiap parameter diskalakan, untuk memudahkan dalam melihat grafik pertumbuhan tingkat pembusukan. Dengan persamaan masing masing pembacaan

$$\text{pH} = (7 - \text{Nilai pH}) \quad (4.5)$$

$$\text{TDS} = ((\text{Nilai TDS}/1000)-1) \times 2 \quad (4.6)$$

$$\text{NTU} = (\text{Nilai NTU}/200)-13 \quad (4.7)$$

Persamaan diatas memiliki *output* dengan selisih yang kecil. Sehingga pembacaan pada proses *learning* dapat lebih mudah dilakukan. Nilai *output* dari persamaan (4.6),(4.7),(4.8) dapat dilihat pada gambar 4.8



Gambar 4.8. Grafik Pengujian Susu UHT

Grafik menunjukkan perubahan karakteristik susu uht berlangsung cepat, utamanya pada nilai TDS. Pada 13 jam setelah kemasan terbuka, susu uht mulai basi dan pada 17 jam setelah kemasan terbuka, susu uht telah basi yang ditandai timbulnya bau busuk dan terjadi penggumpalan pada susu uht.

3. Jus Jeruk

Tabel 4.8. Pengujian Jus Jeruk

Kondisi	pH			TDS		NTU	
	Kalibrasi	Nilai	Volt	Nilai	Volt	Nilai	volt
Segar	6,83	5,78	1,78	316	1.190	-4191	5,1
	6,8	5,5	1,79	439	2.220	-3634	4,9
	6,75	5,6	1,79	432	1.170	-4191	5,1
Kurang Segar	7,15	5,9	1,84	432	1.170	-419	5,1
	6,55	6	1,72	533	1.396	-4771	5,2
	6,64	6	1,8	515	1.362	-6645	5,5
	6,64	6,06	1,87	589	1.508	-6645	5,5
	6,62	6,1	1,71	551	1.435	2412	3,1
Basi	6,58	6,2	1,55	521	1.362	2412	3,3
	6,29	6,25	1,59	577	1.469	2412	3,3
	6,26	6,26	1,59	570	1.469	2412	3,3
	6,24	6,27	1,58	497	1.323	2412	3,3
	6,38	6,28	1,6	624	1.591	577	4,02
	6,24	6,3	1,58	587	1.508	689	4
	6,3	6,35	1,56	558	1.455	480	4,08

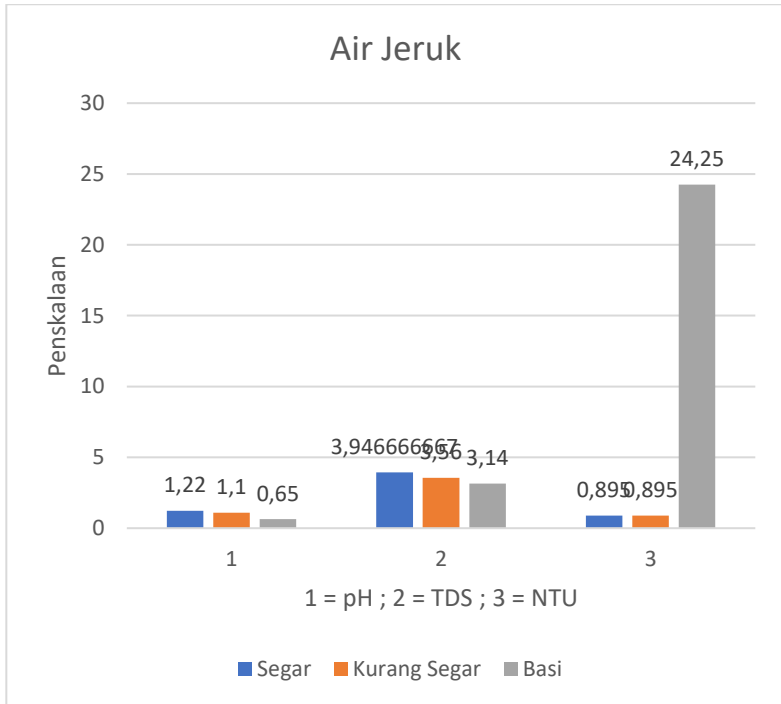
Grafik nilai pembacaan setiap parameter diskalakan, untuk memudahkan dalam melihat grafik pertumbuhan tingkat pembusukan. Dengan persamaan masing masing pembacaan :

$$\text{pH} = (7 - \text{Nilai pH}) \quad (4.8)$$

$$\text{TDS} = (1500 - \text{Nilai TDS})/300 \quad (4.9)$$

$$\text{NTU} = (\text{Nilai NTU} + 2770/200)+8 \quad (4.10)$$

Persamaan diatas memiliki *output* dengan selisih yang kecil. Sehingga pembacaan pada proses *learning* dapat lebih mudah dilakukan. Nilai *output* dari persamaan (4.9),(4.10),(4.11) dapat dilihat pada gambar 4.9



Gambar 4.9. Grafik Pengujian Jus Jeruk

Grafik menunjukkan perubahan pembusukan Jus Jeruk berlangsung lebih cepat. Pada 6 jam setelah kemasan terbuka, jeruk mulai basi namun untuk menjadi basi terjadi di 23 jam setelah kemasan terbuka.

4. Bear Brand

Tabel 4.9. Bear Brand

Kondisi	pH			TDS		NTU	
	Kalibrasi	Nilai	Volt	Nilai	Volt	Nilai	volt
Segar	6,82	6,81	1,95	1221	2.390	2807	3
	6,86	6,83	1,95	1216	2.400	2894	2,9
	6,86	6,83	1,95	1216	2.400	2894	2,9
Mulai Basi	6,92	7,05	2,01	1216	2.400	2894	2,9
	6,55	6,47	1,84	1235	2.421	2958	2,8
	6,6	5,96	1,7	1235	2.421	2412	3,3
	6,61	6,17	1,76	1259	2.446	2566	3,3
	6,69	6,78	1,94	1116	2.284	2566	3,2
Basi	6,6	6,69	1,91	1188	2.373	2236	3,4
	6,28	5,99	1,71	1174	2.133	2236	3,4
	6,26	5,95	1,7	1169	2.353	2566	3,2
	6,23	5,85	1,67	1179	2.353	2566	3,2
	6,36	5,96	1,71	1202	2.377	2840	2,95
	6,24	6,02	1,72	1279	2.465	2893	2,25
	6,29	5,86	1,67	1179	2.348	2808	2,14

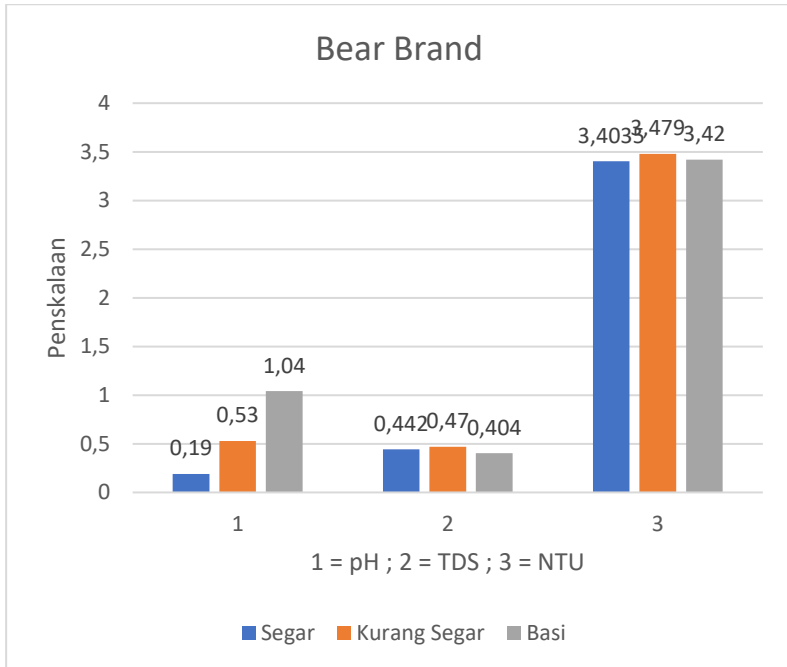
Grafik nilai pembacaan setiap parameter diskalakan, untuk memudahkan dalam melihat grafik pertumbuhan tingkat pembusukan. Dengan persamaan masing masing pembacaan

$$\text{pH} = (7 - \text{Nilai pH}) \quad (4.11)$$

$$\text{TDS} = ((\text{Nilai TDS}/1000)-1) \times 2 \quad (4.12)$$

$$\text{NTU} = (\text{Nilai NTU}/200)-13 \quad (4.13)$$

Persamaan diatas memiliki *output* dengan selisih yang kecil. Sehingga pembacaan pada proses *learning* dapat lebih mudah dilakukan. Nilai *output* dari persamaan (4.12),(4.13),(4.14) dapat dilihat pada gambar 4.10



Gambar 4.10. Grafik Pengujian Bear Brand

Grafik menunjukkan perubahan pembusukan Susu Bear Brand berlangsung lebih cepat. Pada 6 jam setelah kemasan terbuka, Susu Bear Brand mulai basi namun untuk menjadi basi terjadi di 23 jam setelah kemasan terbuka. Hal ini ditandai dengan munculnya bau busuk dan pengumpalan pada Susu Bear Brand.

5. Air Kelapa

Tabel 4.10. Pengujian Air Kelapa

Kondisi	pH			TDS		NTU	
	Kal	Nilai	Volt	Nilai	Volt	Nilai	volt
Segar	6,87	6,76	1,93	1221	2.390	2807	3
	6,85	6,81	1,95	1216	2.390	2958	2,8
		6,81	1,95	1211	2.390	2894	2,9
Kurang Segar	6,92	6,88	1,97	1211	2.390	2894	2,9
	6,67	6,43	1,84	1211	2.407	-4191	5,1
	6,62	5,85	1,67	1225	2.431	-4771	5,2
	6,65	5,66	1,62	1264	2.456	-4191	5,1
	6,62	5,12	1,46	1160	2.324	-6645	5,5
Basi	6,68	5,61	1,6	1197	2.143	1306	3,8
	6,27	5,54	1,58	1179	2.358	2236	3,4
	6,25	5,54	1,59	1183	2.363	2236	3,3
	6,22	5,68	1,63	1179	2.353	2412	3,3
	6,34	5,24	1,59	1188	2.368	-1364	4,54
	6,23	4,82	1,58	1279	2.465	1248	3,8
	6,28	4,7	1,56	1183	2.353	1221	3,82

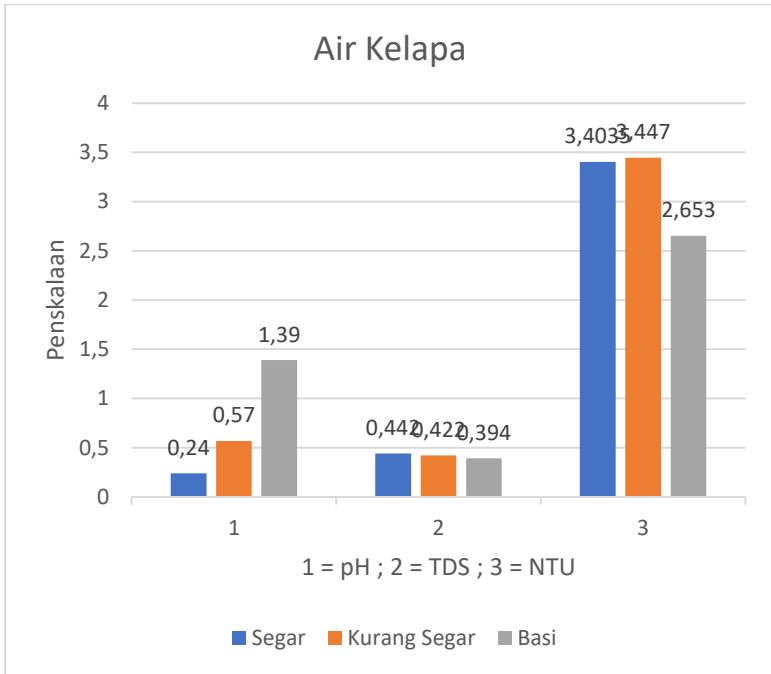
Grafik nilai pembacaan setiap parameter diskalakan, untuk memudahkan dalam melihat grafik pertumbuhan tingkat pembusukan. Dengan persamaan masing masing pembacaan

$$\text{pH} = (7 - \text{Nilai pH}) \quad (4.14)$$

$$\text{TDS} = ((\text{Nilai TDS}/1000)-1) \times 2 \quad (4.15)$$

$$\text{NTU} = (\text{Nilai NTU}/200)-13 \quad (4.16)$$

Persamaan diatas memiliki *output* dengan selisih yang kecil. Sehingga pembacaan pada proses *learning* dapat lebih mudah dilakukan. Nilai *output* dari persamaan (4.15),(4.16),(4.17) dapat dilihat pada gambar 4.11

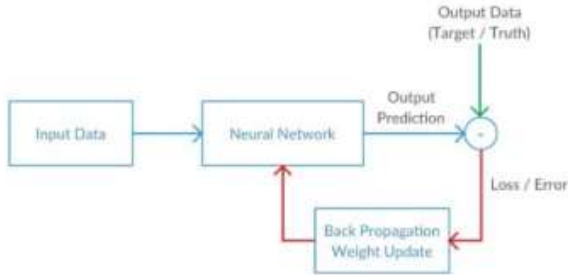


Gambar 4.11. Grafik Pengujian Air Kelapa

Grafik menunjukkan perubahan pembusukan Air Kelapa berlangsung lebih cepat. Pada 6 jam setelah kemasan terbuka, Air Kelapa mulai basi namun untuk menjadi basi terjadi di 23 jam setelah kemasan terbuka. Hal ini ditandai dengan munculnya bau busuk dan perubahan tekstur yang lebih mengental.

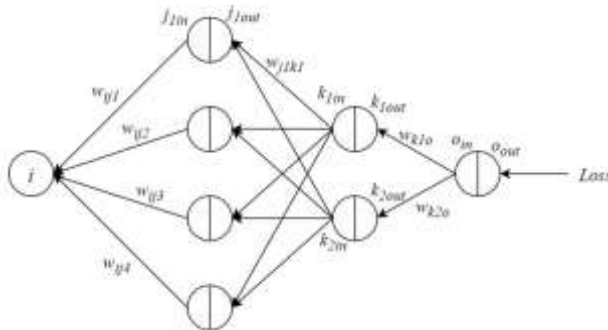
4.3. Pengujian Sistem Jaringan Syaraf Tiruan

Dalam proses pengujian, diperlukan sebuah data untuk diproses sebagai data *learning* pada Jaringan Syaraf Tiruan, untuk menentukan *update* bias dan bobotnya. Pada proses *learning* menggunakan proses *backpropagation* dimana *output* dari *backpropagation* digunakan untuk menentukan bias dan bobot yang sesuai. Berikut adalah proses dan hasil dari *learning* menggunakan *backpropagation*



Gambar 4.12. *Training* Jaringan Syaraf Tiruan [22].

Dalam proses training terdiri dari 2 bagian utama yaitu forward pass dan backward pass. Panah biru pada gambar 4.12. adalah forward pass dan panah merah pada gambar 4.12. adalah backward pass. Dalam supervised learning, training data terdiri dari input dan output atau target. Pada saat forward pass, input akan dipropagasikan menuju output layer dan hasil prediksi output akan dibandingkan dengan target menggunakan fungsi aktivasi, dimana hal ini menggunakan fungsi sigmoid. Pada dasarnya training dilakukan secara terus menerus sehingga loss atau error yang didapatkan memiliki nilai yang relatif kecil.



Gambar 4.13. Backpropagation [22].

4.3.1. Proses *Learning* Dengan Menggunakan Dev++

Pada proses *learning* dilakukan dengan menggunakan software dev++ dengan bahasa pemrograman c. Proses ini mengaplikasikan metode *backpropagation* yang bertujuan untuk menentukan nilai bobot dan bias yang sesuai dengan nilai error sekecil mungkin.

Tabel 4.11. Hasil Identifikasi Minuman Menggunakan Dev++

pH	TDS	NTU	Target Output		Kondisi
0.15	0.8	1.1	0.00000	0.000090	Segar
0.31	0.49	1.1	0.00000	0.805322	Kurang Segar
0.35	0.414	1.085	0.00000	0.991714	Kurang Segar
0.75	0.386	1.085	0.99951	0.999998	Basi
0.7	0.394	1.065	0.99772	0.999997	Basi

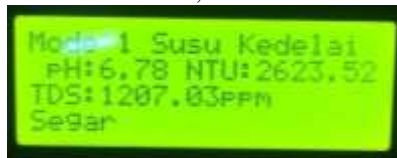
4.4. Pengujian Sistem

Pengujian sampel yang digunakan adalah Susu Kedelai, Susu UHT, Air Jeruk, Bear Brand dan Air kelapa dilakukan dengan pengujian secara langsung mencelupkan semua sensor ke wadah yang telah terisi sampel. Dari pengujian sistem, didapatkan hasil :

1. Susu Kedelai

- Kondisi Segar

Pada kondisi segar nilai pH = 6.78, nilai TDS = 1207,03 ppm dan nilai NTU = 2623,52



Gambar 4.16. *Display* Pembacaan Susu Kedelai Segar

- Kondisi Kurang Segar

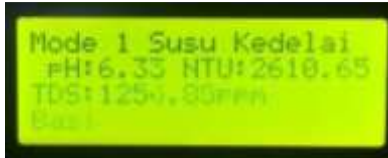
Pada kondisi kurang segar nilai pH = 6.78, nilai TDS = 1193,01 ppm dan nilai NTU = 2813,45



Gambar 4.17. *Display* Pembacaan Susu Kedelai Kurang Segar

- Kondisi Basi

Pada kondisi basi nilai pH = 6.33, nilai TDS = 1254,85 ppm dan nilai NTU = 2610,65

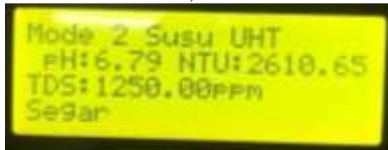


Gambar 4.18. *Display* Pembacaan Susu Kedelai Basi

2. Susu UHT

- Kondisi Segar

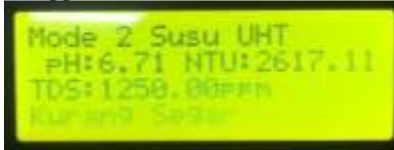
Pada kondisi segar nilai pH = 6.79, nilai TDS = 1250,00 ppm dan nilai NTU = 2610,65



Gambar 4.19. *Display* Pembacaan Susu UHT Segar

- Kondisi Kurang Segar

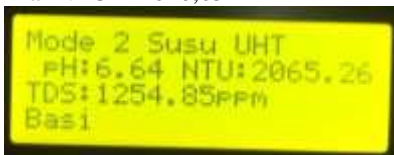
Pada kondisi kurang segar nilai pH = 6.71, nilai TDS = 1250,00 ppm dan nilai NTU = 2617,11



Gambar 4.20. *Display* Pembacaan Susu UHT Kurang Segar

- Kondisi Basi

Pada kondisi basi nilai pH = 6.33, nilai TDS = 1254,85 ppm dan nilai NTU = 2610,65

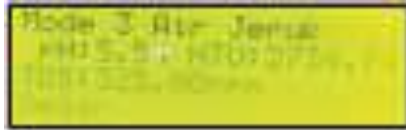


Gambar 4.21. *Display* Pembacaan Susu UHT Basi

3. Air Jeruk

- Kondisi Segar

Pada kondisi segar nilai pH = 5.54, nilai TDS = 325 ppm dan nilai NTU = 2734



Gambar 4.22. *Display* Pembacaan Air Jeruk Segar

- Kondisi Kurang Segar

Pada kondisi kurang segar nilai pH = 5.88, nilai TDS = 439.99 ppm dan nilai NTU = 2780.92



Gambar 4.23. *Display* Pembacaan Air Jeruk Kurang Segar

- Kondisi Basi

Pada kondisi basi nilai pH = 6.10, nilai TDS = 439.99 ppm dan nilai NTU = 718.75



Gambar 4.24. *Display* Pembacaan Air Jeruk Basi

4. Susu Bear Brand

- Kondisi Segar

- Pada kondisi basi nilai pH = 6.85, nilai TDS = 1250 ppm dan nilai NTU = 2678.36



Gambar 4.25. *Display* Pembacaan Bear Brand Segar

- Kondisi Kurang Segar

Pada kondisi basi nilai pH = 6.54, nilai TDS = 1254 ppm dan nilai NTU = 2951.76



Gambar 4.26. *Display* Pembacaan Bear Brand Kurang Segar

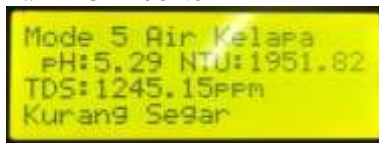
- Kondisi Basi
Pada kondisi basi nilai pH = 4.61, nilai TDS = 1250 ppm dan nilai NTU = -1302.1



Gambar 4.27. *Display* Pembacaan Bear Brand Kurang Segar

5. Air Kelapa

- Kondisi Segar
Pada kondisi basi nilai pH = 5.29, nilai TDS = 1245.15 ppm dan nilai NTU = 1951.82



Gambar 4.28. *Display* Pembacaan Air Kelapa Segar

- Kondisi Kurang Segar
Pada kondisi basi nilai pH = 5.23, nilai TDS = 1245.15ppm dan nilai NTU = 1433.41



Gambar 4.29. *Display* Pembacaan Air Kelapa Kurang Segar

- Kondisi Basi

Pada kondisi basi nilai pH = 5.14, nilai TDS = 1250 ppm dan nilai NTU = 1484.81



Gambar 4.30. *Display* Pembacaan Air Kelapa Basi

Dari 15 kali percobaan, terdapat 4 kesalahan dalam membaca sampel minuman yaitu sampel susu bear brand pada kondisi segar dan 3 diantaranya pada sampel air kelapa. Sehingga hasil keakuratan dalam mengidentifikasi minuman mencapai 73,3%.

BAB 5

PENUTUP

5.1. Kesimpulan

Pada penelitian tugas akhir ini, mengangkat tema pengujian minuman dengan judul Uji Kualitas Minuman Menggunakan Sensor Potensiometri, Konduktivitas Listrik, Optik dan Metode Jaringan Syaraf Tiruan. Jenis sampel yang digunakan adalah susu kedelai, susu uht, air jeruk, susu sapi dengan merek *bear brand* dan air kelapa. Dari masing-masing sampel didapatkan bahwa grafik pada susu kedelai semakin naik seiring pembusukannya, pada grafik susu uht semakin naik seiring pembusukannya, pada grafik air jeruk semakin turun seiring pembusukannya, pada grafik susu sapi "*bear brand*" semakin naik seiring pembusukannya, dan grafik air kelapa semakin naik seiring pembusukannya. Penelitian ini menggunakan metode jaringan syaraf tiruan. Hasil dari pengujian menunjukkan bahwa tingkat keakurasian dari alat ini mencapai 73.34%

5.2. Saran

Dalam pengembangan uji kualitas minuman ini, terdapat beberapa saran yang dapat dipertimbangkan:

1. Penambahan sensor gas untuk untuk mengetahui kondisi minuman melalui bau (gas).
2. Menambahkan neuron serta *hidden layer* agar keluaran yang dihasilkan oleh jaringan syaraf tiruan dapat lebih akurat.
3. Penambahan Sampel pada pengujian, agar cakupan minuman dapat diidentifikasi lebih luas.
4. Penambahan data pengukuran dilakukan sebaiknya setiap 1 jam sekali sampai minuman benar-benar basi agar hasil identifikasi pada minuman dapat lebih tepat.

-Halaman Ini Sengaja Dikosongkan-

DAFTAR PUSTAKA

- [1]. [BSN] Badan Standarisasi Nasional. 1998. Standar Nasional Indonesia (SNI) 013950-1998. Susu UHT.
- [2]. Muhammad Syaif Ramadhan, Muhammad Rivai. 2018, *Sistem Kontrol Tingkat Kekeruhan Pada Aquarium Menggunakan Arduino Uno*. Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember. Surabaya
- [3]. [http : // staffnew. uny .ac .id/ upload/ 132231727/ pendidikan/ BBEVERAG.pdf](http://staffnew.uny.ac.id/upload/132231727/pendidikan/BBEVERAG.pdf) diakses pada tanggal 22 Juni 2019
- [4]. [http : // eprints. ung. ac. id/ 7023/ 5/ 2013 -2- 13201 -811409095-bab2-09012014080923.pdf](http://eprints.ung.ac.id/7023/5/2013-2-13201-811409095-bab2-09012014080923.pdf) diakses pada tanggal 22 Juni 2019
- [5]. Irawan, Djoko Windu P. 2016. "*Prinsip-Prinsip Hygiene Sanitasi Makanan Minuman di Rumah Sakit*". Forum Ilmiah Kesehatan (FORIKES).
- [6]. [https:// www. zenius. net/ blog/ 8870/ mengapa-makanan-basi-cara-awet](https://www.zenius.net/blog/8870/mengapa-makanan-basi-cara-awet) diakses pada tanggal 22 Juni 2019
- [7]. Ramihah, souvia. 2011. Mikrobiologi Susu. [http: // blogs. unpad .ac. id/ souvia/ files/ 2011/ 03/ MIKROBIOLOGI- SUSU. pdf](http://blogs.unpad.ac.id/souvia/files/2011/03/MIKROBIOLOGI-SUSU.pdf).
- [8]. Anggraeni, Dwi Frida & Prihandarini, Ririen. 2016. *Pengaruh jenis komoditi kedelai (organic dan suhu penyimpanan terhadap umur simpan susu kedelai)*. Artikel Publikasi Universitas Widyagama Malang.
- [9]. Aminudin, Muhamad., & Habib, Inayati. 2009. *Pengaruh lamanya penyimpanan terhadap pertumbuhan bakteri pada nasi yang dimasak di Rice cooker dengan nasi yang dikukus*. Jurnal Medika. Vol 9 (2).
- [10]. [https : // www. academia. edu/ 5030062/ KERUSAKAN_ MIKRO BIOLOGIS_ PADA_ JERUK](https://www.academia.edu/5030062/KERUSAKAN_MIKROBIOLOGIS_PADA_JERUK) diakses pada tanggal 22 Juni 2019
- [11]. Santoso, Budi, Hieronymus. 2003. *Air Kelapa , Limbah Penuh Khasiat*. Diakses : pada tanggal 22 Juni 2019. <http://www.kompas.com> (Kompas Cyber Media).
- [12]. Santoso, U., Kubo, K., Ota, T., Tadokoro, T., Maekawa, A. 1996. *Nutrient Composition of Kopyor Coconuts (Cocos nucifera L.)*. *Food Chem.* 57: 299–304.
- [13]. Kiswanto, Y dan Saryanto, S. 2004. *Pengaruh Suhu Lama Penyimpanan Air Kelapa Terhadap Produksi Nata De Coco*. Intitusi Pertanian INTAN Yogyakarta.
- [14]. Drs. Jong Jek Siang, M.sc. 2005. *Jaringan Syaraf Tiruan dan Pemogramannya Menggunakan Matlab*. Penerbit ANDI Yogyakarta.

- [15]. Wahab, Abd. W. Dan La Nafie, N., 2014, Metode Pemisahan dan Pengukuran 2 (Elektrometri dan Spektrometri), Universitas Hasanuddin.
- [16]. Putu Raisha Vishkariana Dewi, 2017 Keseimbangan Cairan dan Elektrolit. Fakultas Kedokteran, Universitas Udayana
- [17]. wikibooks.org Kimia /Materi Asam Basa Garam diakses pada tanggal 24 mei 2019
- [18]. Conductivity Theory and Practice. Radiometer Analytical.
- [19]. Yusdar Zakaria, Yurliasni, Mira Delima, dan Ely Diana. 2013, Analisa Keasaman dan Total Bakteri Asam laktat Yogurt Akibat Bahan Baku dan Presentase *Lactobacillus Casei* yang Berbeda. Jurusan Peternakan, Fakultas Pertanian, Universitas Syiah Kuala
- [20]. Yulia Ningsih. 2014. Analisis Zat Padat. Jurusan Kimia Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung
- [21]. henrysfork.org /hff-launches -new- investigation -turbidity diakses pada tanggal 9 mei 2019
- [22]. <https://medium.com/@samuelsena/pengenalan-deep-learning-8fbb7d8028ac> diakses pada 9 mei 2019
- [23]. <https://purnomosejati.wordpress.com/2011/08/25/mengenal-komunikasi-i2cinter-integrated-circuit/> diakses pada tanggal 9 mei 2019
- [24]. Hamzah Amir. 2013. LCD *Liquid Crystal Display*. www.scribd.com/doc/185920131/LCD-20X4 diakses pada tanggal 07 Februari 2019
- [25]. <https://kelasrobot.com/apa-itu-arduino-arduino-adalah/> diakses pada tanggal 9 mei 2019
- [26]. NN.1998. Guidelines for Drinking Water Quality(Vol.2). Belgium World Health Organization
- [27]. Saleh E. Dasar Pengolahan Susu dan Hasil ikutan Ternak. Program Studi Produksi Ternak Fakultas pertanian Universitas Sumatra Utara.

LAMPIRAN A

- *Source Code*

```
#define SensorPin A0
#define Offset 0.00
#define LED 13
#define samplingInterval 20
#define ArrayLenth 40
#define wire.h
#include <EEPROM.h>
#include "GravityTDS.h"
#define TdsSensorPin A1
GravityTDS gravityTds;
#include <OneWire.h>
#include <LiquidCrystal_I2C.h>
int pHArray[ArrayLenth];
int pHArrayIndex=0;
float volt;
float ntu;
float temperature = 25,tdsValue = 0;
const byte numReadings = 20;
byte ECsensorPin = A1;

unsigned int
AnalogSampleInterval=25,printInterval=700,tempSampleInterval=850;
unsigned int readings[numReadings];
byte index = 0;
unsigned long AnalogValueTotal = 0;
unsigned int AnalogAverage = 0,averageVoltage=0;
unsigned long AnalogSampleTime,printTime,tempSampleTime;
float pHValue,voltage;

LiquidCrystal_I2C lcd(0x3F,20,4);
int counter = 0;
int switchPin = 7;
void setup(){
Serial.begin(9600);
```

```

pinMode(2, INPUT);
pinMode(switchPin, INPUT);
gravityTds.setPin(TdsSensorPin);
gravityTds.setAref(5.0);
gravityTds.setAdcRange(1024);
gravityTds.begin();

for (byte thisReading = 0; thisReading < numReadings;
thisReading++)
    readings[thisReading] = 0;

AnalogSampleTime=millis();
printTime=millis();
tempSampleTime=millis();

}
float round_to_dp(float in_value, int decimal_place)
{
    float multiplier = powf (10.0f, decimal_place);
    in_value = roundf(in_value * multiplier)/multiplier;
    return in_value;
}
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){
        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;
    min=arr[i];
  }else {
    if(arr[i]>max){
      amount+=max;
      max=arr[i];
    }else{
      amount+=arr[i];
    }
  }
  avg = (double)amount/(number-2);
return avg;
}
void sensor(){
  int sensorValue = analogRead(A2);
  volt = sensorValue * (5.0 / 1024.0)+1.95;
  ntu = -1120.4*square(volt)+5742.3*volt-4352.9;
//pH measurement
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval)
  {
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
  }
  //if(millis() - printTime > printInterval)
  Serial.print("pH value: ");

```

```

Serial.print(pHValue);
Serial.print(" Voltage:");
Serial.println(voltage);
Serial.print("NTU:");
Serial.print(ntu);
Serial.print(" Volt:");
Serial.println(volt);
printTime=millis();

if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
  AnalogSampleTime=millis();
  AnalogValueTotal = AnalogValueTotal - readings[index];
  readings[index] = analogRead(ECsensorPin);
  AnalogValueTotal = AnalogValueTotal + readings[index];
  index = index + 1;
  if (index >= numReadings)
  index = 0;
  AnalogAverage = AnalogValueTotal / numReadings;
}
if(millis()-printTime>=printInterval)
{
printTime=millis();
averageVoltage=AnalogAverage*(float)5000/1024;
Serial.print(" AV:");
Serial.print(AnalogAverage);
Serial.print(" Volt:");
Serial.print(averageVoltage);
Serial.print("mV ");

gravityTds.setTemperature(temperature);
gravityTds.update();
tdsValue = gravityTds.getTdsValue();
Serial.print(" ppm:");
Serial.println(tdsValue,0);
}
lcd.setCursor(1,1);
lcd.print("pH:");

```

```

    lcd.print(pHValue);
    lcd.print(" NTU:");
    lcd.print(ntu);

    lcd.setCursor(0,2);
    lcd.print("TDS:");
    lcd.print(tdsValue);
    lcd.print("ppm");
}
void loop(){
    //Turbidity
    lcd.init();
    lcd.backlight();
    int i, j, p, q, r;
    float Accum, Error;
    const int PatternCount = 28;
    const int InputNodes = 3;
    const int HiddenNodes = 8;
    const int OutputNodes = 2;
    float Hidden[HiddenNodes];
    float Output[OutputNodes];

    int switchVal = digitalRead(switchPin);
    if(switchVal == HIGH)
    {
        counter ++;
        switchVal == LOW;
        if(counter == 6)
        {
            counter = 1;
        }
    }
    lcd.setCursor(0,0);
    lcd.print("Mode ");
    lcd.print(counter);

    int nn = 0;
    nn = digitalRead(2);

```

```

if(counter == 1){
  lcd.print(" Susu Kedelai");
  //lcd.print(nn);
  sensor();

//TDS
if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
  AnalogSampleTime=millis();
  AnalogValueTotal = AnalogValueTotal - readings[index];
  readings[index] = analogRead(ECsensorPin);
  AnalogValueTotal = AnalogValueTotal + readings[index];
  index = index + 1;
  if (index >= numReadings)
  index = 0;
  AnalogAverage = AnalogValueTotal / numReadings;
}
if(millis()-printTime>=printInterval)
{
  printTime=millis();
  averageVoltage=AnalogAverage*(float)5000/1024;
  gravityTds.setTemperature(temperature);
  gravityTds.update();
  tdsValue = gravityTds.getTdsValue();

//pH measurement
static unsigned long samplingTime = millis();
static unsigned long printTime = millis();
static float pHValue,voltage;
if(millis()-samplingTime > samplingInterval)
{
  pHArray[pHArrayIndex++]=analogRead(SensorPin);
  if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
  voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
  pHValue = 3.5*voltage+Offset;
  samplingTime=millis();
}

```



```

float HiddenWeights[InputNodes][HiddenNodes] = {
{-17.38, -7.58, 27.84, 11.23, 24.38, 18.77, 24.71, 5.46},
{6.61, 10.87, -11.82, -5.23, -9.89, -8.94, -10.61, -3.06},
{2.58, -2.03, -1.68, 0.04, -1.05, -1.05, -0.99, 0.18},
};
float OutputWeights[HiddenNodes][OutputNodes] = {
{-8.72, 0.09},
{-9.05, -13.98},
{8.87, 3.40},
{4.12, 2.24},
{7.91, 2.74},
{6.53, 3.01},
{8.09, 3.23},
{2.19, 2.14},
};
float HiddenBias[HiddenNodes] = {0.47, -0.40, -1.82, -1.13, -1.90, -
1.20, -1.88, -0.64};
float OutputBias[OutputNodes] = {-1.68, 0.97};
float x = 7.1 - pHValue ;
float y = ( ntu /200 )-13;
float z = ((tdsValue / 1000)-1)*2;
float Input[3] = { x, y, z };

for( i = 0 ; i < HiddenNodes ; i++ ) {
    Accum = HiddenWeights[InputNodes][i] ;
    for(j = 0 ; j < InputNodes ; j++ ) {
        Accum += ( Input[j] * HiddenWeights[j][i] ) + HiddenBias[i] ;
    }
    Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
}
for( i = 0 ; i < OutputNodes ; i++ ) {
    Accum = OutputWeights[HiddenNodes][i] ;
    for(j = 0 ; j < HiddenNodes ; j++ ) {
        Accum += (Hidden[j] * OutputWeights[j][i]) + OutputBias[i] ;
    }
    Output[i] = 1.0/(1.0 + exp(-Accum)) ;
}
float a = Output[0];

```

```

float b = Output[1];
if( a < 0.5 && b < 0.5){
    lcd.setCursor(0,3);
    lcd.print("Segar");
}
else if ( a < 0.5 && b > 0.5){
    lcd.setCursor(0,3);
    lcd.print("Kurang Segar");
}
else if ( a > 0.5 && b > 0.55 ){
    lcd.setCursor(0,3);
    lcd.print("Basi");
}
lcd.clear();
lcd.setCursor(0,1);
lcd.print(x);
lcd.setCursor(0,2);
lcd.print(y);
delay(5000);
}
}
else if(counter == 2){
    lcd.print(" Susu UHT  ");
    sensor();

if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
    AnalogSampleTime=millis();
    AnalogValueTotal = AnalogValueTotal - readings[index];
    readings[index] = analogRead(ECsensorPin);
    AnalogValueTotal = AnalogValueTotal + readings[index];
    index = index + 1;
    if (index >= numReadings)
        index = 0;
    AnalogAverage = AnalogValueTotal / numReadings;
}
if(millis()-printTime>=printInterval)
{

```

```

printTime=millis();
averageVoltage=AnalogAverage*(float)5000/1024;
gravityTds.setTemperature(temperature);
gravityTds.update();
tdsValue = gravityTds.getTdsValue();

//pH measurement
static unsigned long samplingTime = millis();
static unsigned long printTime = millis();
static float pHValue,voltage;
if(millis()-samplingTime > samplingInterval)
{
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
}

float HiddenWeights[InputNodes][HiddenNodes] = {
{9.55, -3.06, 16.45, 8.44, 17.97, 14.02, 19.83, 3.65},
{-4.54, 9.97, 3.04, -1.47, 4.69, -5.90, -3.92, -2.53},
{3.48, -2.39, -0.83, 1.60, -1.15, 2.78, 2.85, 1.49},
};

float OutputWeights[HiddenNodes][OutputNodes] = {
{-0.85, 6.42},
{5.32, -7.20},
{10.51, 7.39},
{2.23, 4.99},
{12.73, 7.29},
{1.35, 8.99},
{5.39, 12.41},
{-0.70, 2.82},
};

float HiddenBias[HiddenNodes] = {-1.62, -0.67, -1.31, -1.04, -1.69, -
1.00, -1.64, -0.59};
float OutputBias[OutputNodes] = {-2.64, -1.78};

```

```

float x = 7.2 - pHValue ;
float y = ( ntu /200 )-13;
float z = ((tdsValue / 1000)-1)*2;
float Input[3] = { x, y, z };
  for( i = 0 ; i < HiddenNodes ; i++ ) {
    Accum = HiddenWeights[InputNodes][i] ;
    for( j = 0 ; j < InputNodes ; j++ ) {
      Accum += ( Input[j] * HiddenWeights[j][i]) + HiddenBias[i] ;
    }
    Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
  }
  for( i = 0 ; i < OutputNodes ; i++ ) {
    Accum = OutputWeights[HiddenNodes][i] ;
    for( j = 0 ; j < HiddenNodes ; j++ ) {
      Accum += (Hidden[j] * OutputWeights[j][i]) + OutputBias[i] ;
    }
    Output[i] = 1.0/(1.0 + exp(-Accum)) ;
  }
float a = Output[0];
float b = Output[1];
if( a < 0.5 && b < 0.5){
  lcd.setCursor(0,3);
  lcd.print("Segar");
}
else if ( a < 0.5 && b > 0.5){
  lcd.setCursor(0,3);
  lcd.print("Kurang Segar");
}
else if ( a > 0.5 && b > 0.5 ){
  lcd.setCursor(0,3);
  lcd.print("Basi");
}
}
}
else if(counter == 3){
lcd.print(" Air Jeruk  ");
sensor();

```

```

if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
    AnalogSampleTime=millis();
    AnalogValueTotal = AnalogValueTotal - readings[index];
    readings[index] = analogRead(ECsensorPin);
    AnalogValueTotal = AnalogValueTotal + readings[index];
    index = index + 1;
    if (index >= numReadings)
        index = 0;
    AnalogAverage = AnalogValueTotal / numReadings;
}
if(millis()-printTime>=printInterval)
{
    printTime=millis();
    averageVoltage=AnalogAverage*(float)5000/1024;
    gravityTds.setTemperature(temperature);
    gravityTds.update();
    tdsValue = gravityTds.getTdsValue();

//pH measurement
static unsigned long samplingTime = millis();
static unsigned long printTime = millis();
static float pHValue,voltage;
if(millis()-samplingTime > samplingInterval)
{
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
}

    float HiddenWeights[InputNodes][HiddenNodes] = {
{8.76, -4.12, -11.15, -7.87, -2.65, -12.14, -4.65, -2.23},
{3.41, -0.11, 0.50, -0.91, 1.15, -0.43, 4.32, 0.85},
{-1.10, 1.74, -2.27, 1.85, 3.21, -1.76, 0.53, 0.12},
};
float OutputWeights[HiddenNodes][OutputNodes] = {
{-6.94, -0.19},

```

```

{2.81, 0.34},
{7.23, 7.50},
{5.84, 1.77},
{5.43, 1.68},
{7.93, 6.77},
{1.66, 7.25},
{1.27, 1.96},
};

```

```

float HiddenBias[HiddenNodes] = {-2.05, -0.62, 3.68, 0.41, -0.90, 3.72,
1.09, 0.05};
float OutputBias[OutputNodes] = {-1.69, -1.08};
float x = 7 - pHValue ;
float y = ( ntu /200 )-13;
float z = ((tdsValue / 1000)-1)*2;
float Input[3] = { x, y, z };

```

```

for( i = 0 ; i < HiddenNodes ; i++ ) {
    Accum = HiddenWeights[InputNodes][i] ;
    for( j = 0 ; j < InputNodes ; j++ ) {
        Accum += ( Input[j] * HiddenWeights[j][i]) + HiddenBias[i] ;
    }
    Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
}
for( i = 0 ; i < OutputNodes ; i++ ) {
    Accum = OutputWeights[HiddenNodes][i] ;
    for( j = 0 ; j < HiddenNodes ; j++ ) {
        Accum += (Hidden[j] * OutputWeights[j][i]) + OutputBias[i] ;
    }
    Output[i] = 1.0/(1.0 + exp(-Accum)) ;
}
float a = Output[0];
float b = Output[1];
if( a < 0.5 && b < 0.55){
    lcd.setCursor(0,3);
    lcd.print("Segar");
}
else if ( a < 0.5 && b > 0.5){

```

```

    lcd.setCursor(0,3);
    lcd.print("Kurang Segar");
  }
  else if ( a > 0.5 && b > 0.5 ){
    lcd.setCursor(0,3);
    lcd.print("Basi");
  }
}
}
else if(counter == 4){
  lcd.print(" Bear Brand ");
  sensor();

if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
  AnalogSampleTime=millis();
  AnalogValueTotal = AnalogValueTotal - readings[index];
  readings[index] = analogRead(ECsensorPin);
  AnalogValueTotal = AnalogValueTotal + readings[index];
  index = index + 1;
  if (index >= numReadings)
  index = 0;
  AnalogAverage = AnalogValueTotal / numReadings;
}
if(millis()-printTime>=printInterval)
{
  printTime=millis();
  averageVoltage=AnalogAverage*(float)5000/1024;
  gravityTds.setTemperature(temperature);
  gravityTds.update();
  tdsValue = gravityTds.getTdsValue();

//pH measurement
static unsigned long samplingTime = millis();
static unsigned long printTime = millis();
static float pHValue,voltage;
if(millis()-samplingTime > samplingInterval)
{

```

```

    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
}
float HiddenWeights[InputNodes][HiddenNodes] = {
{0.92, 20.83, 11.53, 1.01, 12.88, 0.60, 3.62, -10.08},
{-0.08, -9.81, -6.72, -0.48, -10.98, -0.61, -1.03, -2.56},
{-0.83, 5.62, -7.98, -1.32, -8.19, -2.07, -3.42, 11.86},
};
float OutputWeights[HiddenNodes][OutputNodes] = {
{-0.25, 0.62},
{-2.04, -15.04},
{8.76, 9.89},
{0.58, 0.76},
{10.46, 12.91},
{0.85, 1.07},
{1.15, 1.64},
{13.85, 1.24},
};

float HiddenBias[HiddenNodes] = {-2.55, -3.93, 6.06, -1.07, 6.57, -0.26,
-0.30, -6.98};
float OutputBias[OutputNodes] = {-2.79, 1.40};
float x = 7 - pHValue ;
float y = ( ntu /200)-13;
float z = ((tdsValue / 1000)-1)*2;
float Input[3] = { x, y, z };

for( i = 0 ; i < HiddenNodes ; i++) {
    Accum = HiddenWeights[InputNodes][i] ;
    for(j = 0 ; j < InputNodes ; j++) {
        Accum += ( Input[j] * HiddenWeights[j][i]) + HiddenBias[i] ;
    }
    Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
}

```



```

/*****
      Compute output layer activations and calculate errors
*****/

for( i = 0 ; i < OutputNodes ; i++ ) {
    Accum = OutputWeights[HiddenNodes][i] ;
    for(j = 0 ; j < HiddenNodes ; j++ ) {
        Accum += (Hidden[j] * OutputWeights[j][i]) + OutputBias[i] ;
    }
    Output[i] = 1.0/(1.0 + exp(-Accum)) ;
}
float a = Output[0];
float b = Output[1];
if( a < 0.5 && b < 0.55){
    lcd.setCursor(0,3);
    lcd.print("Segar");
}
else if ( a < 0.5 && b > 0.5){
    lcd.setCursor(0,3);
    lcd.print("Kurang Segar");
}
else if ( a > 0.5 && b > 0.5 ){
    lcd.setCursor(0,3);
    lcd.print("Basi");
}
}
}
else if(counter == 5){
    lcd.print(" Air Kelapa ");
    sensor();
}

if(millis()-AnalogSampleTime>=AnalogSampleInterval)
{
    AnalogSampleTime=millis();
    AnalogValueTotal = AnalogValueTotal - readings[index];
    readings[index] = analogRead(ECsensorPin);
    AnalogValueTotal = AnalogValueTotal + readings[index];
    index = index + 1;
}

```

```

    if (index >= numReadings)
    index = 0;
    AnalogAverage = AnalogValueTotal / numReadings;
  }
  if(millis()-printTime>=printInterval)
  {
    printTime=millis();
    averageVoltage=AnalogAverage*(float)5000/1024;
    gravityTds.setTemperature(temperature);
    gravityTds.update();
    tdsValue = gravityTds.getTdsValue();

//pH measurement
    static unsigned long samplingTime = millis();
    static unsigned long printTime = millis();
    static float pHValue,voltage;
    if(millis()-samplingTime > samplingInterval)
    {
      pHArray[pHArrayIndex++]=analogRead(SensorPin);
      if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
      voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
      pHValue = 3.5*voltage+Offset;
      samplingTime=millis();
    }
    float HiddenWeights[InputNodes][HiddenNodes] = {
    {8.93, -1.65, 0.16, -0.35, 0.70, 9.51, 1.54, 8.98},
    {0.58, -0.81, -0.78, -0.36, -0.09, 0.56, 0.63, 0.51},
    {-3.81, -7.88, -9.69, -2.29, -5.82, -5.20, 1.53, -4.90},
    };
    float OutputWeights[HiddenNodes][OutputNodes] = {
    {0.69, 7.60},
    {6.32, 0.38},
    {6.74, 1.87},
    {1.20, 0.95},
    {3.07, 1.48},
    {1.56, 8.63},
    {-1.16, 0.99},
    {1.16, 8.11},

```

```

};

float HiddenBias[HiddenNodes] = {0.30, -4.13, -3.64, -1.80, -2.37, 1.03,
-0.90, 0.84};
float OutputBias[OutputNodes] = {-1.66, -1.06};
float x = 7 - pHValue ;
float y = ( ntu /200 )-13;
float z = ((tdsValue / 1000)-1)*2;
float Input[3] = { x, y, z };
    for( i = 0 ; i < HiddenNodes ; i++ ) {
        Accum = HiddenWeights[InputNodes][i] ;
        for(j = 0 ; j < InputNodes ; j++ ) {
            Accum += ( Input[j] * HiddenWeights[j][i]) + HiddenBias[i] ;
        }
        Hidden[i] = 1.0/(1.0 + exp(-Accum)) ;
    }
    for( i = 0 ; i < OutputNodes ; i++ ) {
        Accum = OutputWeights[HiddenNodes][i] ;
        for(j = 0 ; j < HiddenNodes ; j++ ) {
            Accum += (Hidden[j] * OutputWeights[j][i]) + OutputBias[i] ;
        }
        Output[i] = 1.0/(1.0 + exp(-Accum)) ;
    }
float a = Output[0];
float b = Output[1];
if( a < 0.5 && b < 0.55){
    lcd.setCursor(0,3);
    lcd.print("Segar");
}
else if ( a < 0.5 && b > 0.5){
    lcd.setCursor(0,3);
    lcd.print("Kurang Segar");
}
else if ( a > 0.5 && b > 0.5 ){
    lcd.setCursor(0,3);
    lcd.print("Basi");
}
}

```

}
}

LAMPIRAN B

- Data Sheet Sensor Potensiometrik (SEN0161)



PH meter(SKU: SEN0161)



Analog pH Meter Kit SKU: SEN0161



Analog pH Meter Kit SKU: SEN0169

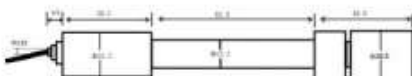
Contents

- 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 pH6~8, then SEN0161 is capable for that.

pH Electrode Characteristics

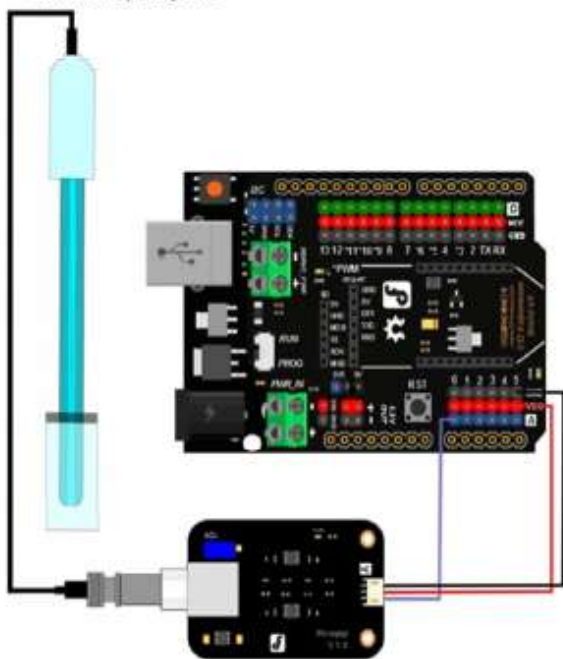
The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 °C):

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:

Before you insert the pH probe into one solution from another, or after you finish using the sensor, you must wash the pH electrode with pure water everytime (distilled water is the best)!

The closer **power supply** to +5.00V, the more accurate pH readings you could get. You have to immerse the pH probe into stationary solution instead of the running one to get relative stable pH readings.

How long should it be under the solution? It depends on the pH value, the closer to neutral solution (pH = 7.00), the longer it will take. As we tested in water pH = 6.0, the blue one costs 6 minutes, and in standard Acid/ Alkali (4.00/ 10.00) solutions, it only needs 10 seconds.

Method 1. Software Calibration

The software calibration is easier than the next part - Hardware Calibration through the Potentiometer. Because it writes the calibration values into Arduino's EEPROM, so you can calibrate once for all if you won't replace your Arduino. It uses mathematical method that to draw a line using two points, i.e. using the Acid standard solution, pH = 4.00 and alkaline pH = 10.00 or 9.18 to draw the linear relation between the voltage and the pH value.



For NOTE 3. Arduino sample sketch "EEPROM Clear"

NOTE:

During the calibration (from step 4 to step 7), power outage should be avoided, or you will have to start over from step 4.

Software Calibration has nothing to do with the potentiometer on the adapter. Especially after you finished the calibration, you should never adjust the potentiometer, or you should start over. Moreover, considering the mechanical vibration might interfere the potentiometer value, you could seal it by Hot Melt Adhesive.

If you want to try Hardware Calibration, you'd better reset the EEPROM setting by uploading the Arduino IDE sample sketch "EEPROM Clear" as shown as the right hand picture.



In Standard acid solution pH = 4.00



In Standard alkali solution pH = 10.00

FAQ

Q1. My PH sensor readings are not correct, what did I miss?

Or the module is defective?

A. 1. Check if the pH sensor circuit board is good? [Read on the Forum](#) or [on wiki](#) for the steps. During the transport, there might be crash causing the probe head cracked, please check if the probe is good or not.

2. If you don't use Arduino as the controller, then please check your ADC module that whether it converts the 5V analog input to 1024, if it is 4096(or other byte), please re-determine the equation in the code.

Q2. Big fluctuations in ph meter readings. When I make measurements in a glass, I have correct, stable reading. But when I put it inside the aquarium with the pumping system working, the measurement varies even more than a degree, and it's not stable, if I switch off the pump the given value doesn't oscilate anymore.

A. There should be NO working electrical device in the container. Any tiny leakage of electricity will cause the probe working error. Especially, many people bought the [EC meter](#) and put it into the same tank for the test, but then the pH meter cannot work well anymore. Please separate them into different containers, or turning off the EC meter when using the pH meter.

Q3. May I know the Maximum range different if we do not calibrate the pH meter.

A. The maximum range differs from probe, you have to calibrate it before use if the pH probe was kept long.

Q4. I would just like to ask if your pH sensor can be connect to any micro controller aside from arduino. Would it be compatible with a raspberry pi? Thank You!

A. Yes, it can be used on any device as long as it could give 5V power supply and accept 5V analog signal, but as the Rasp pi is only compatible with 3.3V sensor, so an expansion shield is suggested to use with (please make sure which kind of Pi you use)

For any questions and more cool ideas to share, please visit [DFRobot Forum](#)

-Halaman Ini Sengaja Dikosongkan-

LAMPIRAN C

- Data Sheet Sensor Konduktivitas Listrik (SEN0244)



Gravity: Analog TDS Sensor / Meter For Arduino

SKU: SEN0244

TDS (Total Dissolved Solids) indicates that how many milligrams of soluble solids dissolved in one liter of water. In general, the higher the TDS value, the more soluble solids dissolved in water, and the less clean the water is. Therefore, the TDS value can be used as one of the references for reflecting the cleanliness of water.

TDS pen is a widely used equipment to measure TDS value. The price is affordable, and it is easy to use, but it is not able to transmit data to the control system for online monitoring to do some water quality analysis. The professional instrument has high accuracy and can send data to the control system, but the price is expensive for the ordinary people. To this end, we have launched an analog TDS sensor kit which is compatible with Arduino, plug and play, easy to use. Matching with Arduino controller, you can build a TDS detector easily to measure the TDS value of liquid.

This product supports 3.3 ~ 5.5V wide voltage input, and 0 ~ 2.3V analog voltage output, which makes it compatible with 5V or 3.3V control system or board. The excitation source is AC signal, which can effectively prevent the probe from polarization and prolong the life of the probe, meanwhile, increase the stability of the output signal. The TDS probe is waterproof, it can be immersed in water for long time measurement.

This product can be used in water quality application, such as domestic water, hydroponics. With this product, you can easily DIY a TDS detector to reflect the cleanliness of water to protect your health.



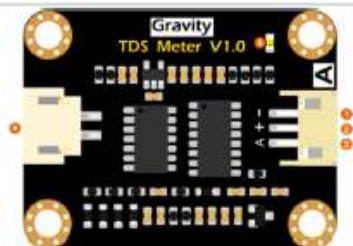
Attention:

- 1.The probe can not be used in water above 55 degrees centigrade.
- 2.The probe can not be left too close to the edge of the container, otherwise it will affect the reading.
- 3.The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof. Please be careful.

Specification

- **Signal Transmitter Board**
Input Voltage: 3.3 ~ 5.5V
Output Voltage: 0 ~ 2.3V
Working Current: 3 ~ 6mA
TDS Measurement Range: 0 ~ 1000ppm
TDS Measurement Accuracy: $\pm 10\%$ F.S. (25 °C)
Module Size: 42 * 32mm
Module Interface: PH2.0-3P
Electrode Interface: XH2.54-2P
- **TDS probe**
Number of Needle: 2
Total Length: 83cm
Connection Interface: XH2.54-2P
Colour: Black
Other: Waterproof Probe

Board Overview



Analog TDS Sensor / Meter For Arduino

Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.3 ~ 5.5V)
3	A	Analog Signal Output(0 ~ 2.3V)
4	TDS	TDS Probe Connector
5	LED	Power Indicator

Tutorial

This tutorial will show you how to measure the TDS value of the water. Please read this tutorial carefully, and pay attention to the steps and details.



The probe can not to be used in water above 55 degrees centigrade.
 The probe can not be too close to the edge of the container, otherwise it will affect the reading.
 The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof. Please pay attention to use.

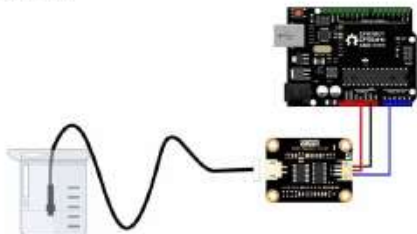
Requirements

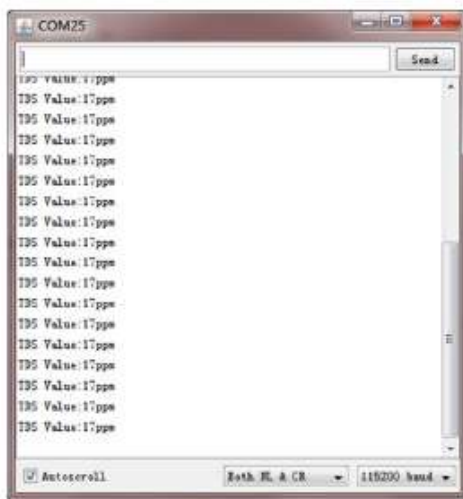
- **Hardware**
 DFRduino UNO R3 (or similar) x 1
 Analog TDS Sensor / Meter Module x 1
 TDS Probe x1
 Jumper Wires x3
 tested liquid x1
- **Software**

Arduino IDE (Version requirements: V1.0.x or V1.8.x), Click to Download Arduino IDE from Arduino®

<https://www.arduino.cc/en/Main/Software%7C>

Connection Diagram





FAQ

Q1. Does this sensor have a temperature sensor? How to make the temperature compensation?

A. This TDS probe has no temperature sensor, but the temperature compensation algorithm is reserved in the sample code. The temperature variable in the sample code will default to 25 °C without a temperature sensor. You can add a waterproof temperature sensor to read the temperature, then update the temperature variable, to make automatic temperature compensation.

For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](#).

https://www.dfrobot.com/wiki/index.php/Gravity:_Analog_TDS_Sensor/_Meter_For_Arduino_SKU:_SEN0244 12-8-17

LAMPIRAN D

- Data Sheet Sensor Optik (SEN0189)



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.

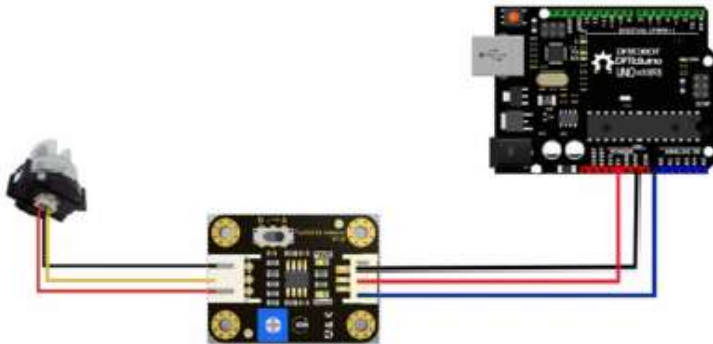
This sensor provides analog and digital signal output modes. The threshold is adjustable when in digital signal mode. You can select the mode according to your MCU.

 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 reading (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
```

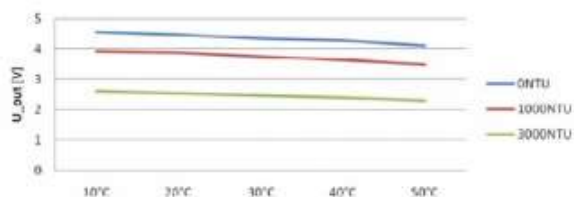
```

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
  }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

LAMPIRAN E

- Data Sheet PCF8584 (I2C Bus Controller)

INTEGRATED CIRCUITS

DATA SHEET

PCF8584 I²C-bus controller

Product specification
Supersedes data of 1997 Mar 19
File under Integrated Circuits, IC12

1997 Oct 21

Philips
Semiconductors



PHILIPS

I²C-bus controller

PCF8584

CONTENTS	7	SOFTWARE FLOWCHART EXAMPLES
1	7.1	Initialization
2	7.2	Implementation
3	8	I ² C-BUS TIMING DIAGRAMS
4	9	LIMITING VALUES
5	10	HANDLING
6	11	DC CHARACTERISTICS
6.1	12	I ² C-BUS TIMING SPECIFICATIONS
6.2	13	PARALLEL INTERFACE TIMING
6.3	14	APPLICATION INFORMATION
6.4	14.1	Application Notes
6.5	15	PACKAGE OUTLINES
6.6	16	SOLDERING
6.7	16.1	Introduction
6.8	16.2	DIP
6.8.1	16.2.1	Soldering by dipping or by wave
6.8.1.1	16.2.2	Repairing soldered joints
6.8.1.2	16.3	SO
6.8.1.3	16.3.1	Reflow soldering
6.8.1.4	16.3.2	Wave soldering
6.8.1.5	16.3.3	Repairing soldered joints
6.8.1.6	17	DEFINITIONS
6.8.2	18	LIFE SUPPORT APPLICATIONS
6.8.2.1	19	PURCHASE OF PHILIPS I ² C COMPONENTS
6.8.2.2		
6.8.2.3		
6.8.2.4		
6.8.2.5		
6.8.2.6		
6.8.2.7		
6.9		
6.10		
6.11		
6.11.1		
6.11.2		
6.12		
6.12.1		
6.12.2		
6.12.3		



I²C-bus controller

PCF8584

1 FEATURES

- Parallel-bus to I²C-bus protocol converter and interface
- Compatible with most parallel-bus microcontrollers/microprocessors including 8049, 8051, 6800, 68000 and Z80
- Both master and slave functions
- Automatic detection and adaption to bus interface type
- Programmable interrupt vector
- Multi-master capability
- I²C-bus monitor mode
- Long-distance mode (4-wire)
- Operating supply voltage 4.5 to 5.5 V
- Operating temperature range: -40 to +85 °C.

2 GENERAL DESCRIPTION

The PCF8584 is an integrated circuit designed in CMOS technology which serves as an interface between most standard parallel-bus microcontrollers/microprocessors and the serial I²C-bus. The PCF8584 provides both master and slave functions.

Communication with the I²C-bus is carried out on a byte-wise basis using interrupt or polled handshake. It controls all the I²C-bus specific sequences, protocol, arbitration and timing. The PCF8584 allows parallel-bus systems to communicate bidirectionally with the I²C-bus.

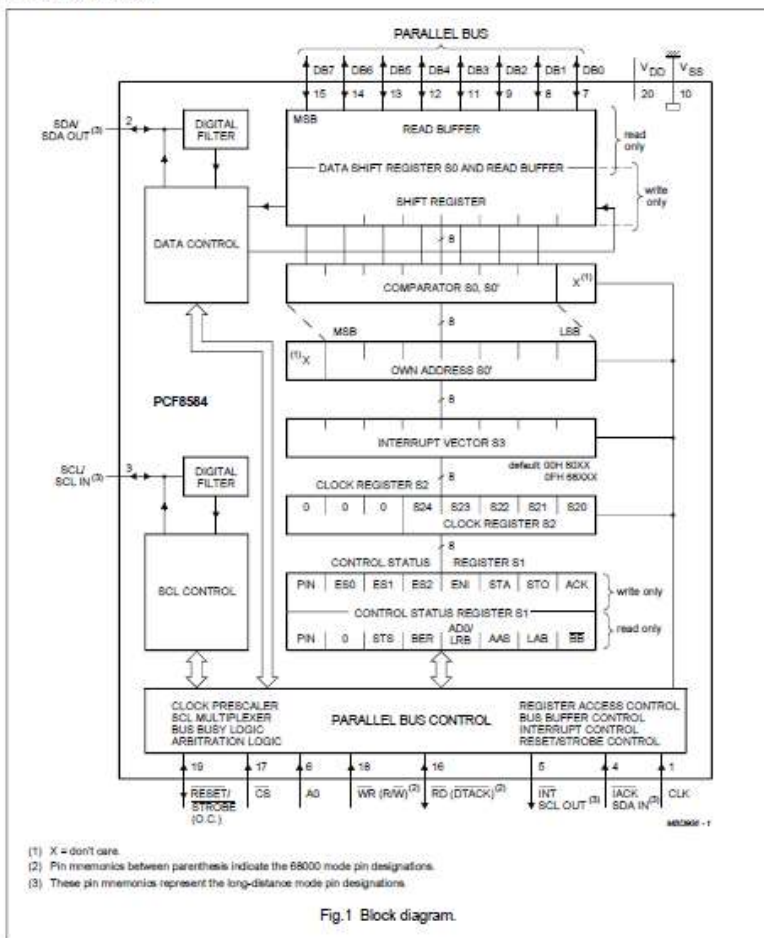
3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8584P	DIP20	plastic dual in-line package; 20 leads (300 mil)	SOT146-1
PCF8584T	SO20	plastic small outline package; 20 leads; body width 7.5 mm	SOT163-1

I²C-bus controller

PCF8584

4 BLOCK DIAGRAM



I²C-bus controller

PCF8584

5 PINNING

SYMBOL	PIN	I/O	DESCRIPTION
CLK	1	I	clock input from microcontroller clock generator (internal pull-up)
SDA or SDA OUT	2	I/O	I ² C-bus serial data input/output (open-drain). Serial data output in long-distance mode.
SCL or SCL IN	3	I/O	I ² C-serial clock input/output (open-drain). Serial clock input in long-distance mode.
TACK or SDA IN	4	I	Interrupt acknowledge input (internal pull-up); when this signal is asserted the interrupt vector in register S3 will be available at the bus Port if the ENI flag is set. Serial data input in long-distance mode.
TINT or SCL OUT	5	O	Interrupt output (open-drain); this signal is enabled by the ENI flag in register S1. It is asserted when the PIN flag is reset. (PIN is reset after 1 byte is transmitted or received over the I ² C-bus). Serial clock output in long-distance mode.
A0	6	I	Register select input (internal pull-up); this input selects between the control/status register and the other registers. Logic 1 selects register S1, logic 0 selects one of the other registers depending on bits loaded in ESO, ES1 and ES2 of register S1.
DB0	7	I/O	bidirectional 8-bit bus Port 0
DB1	8	I/O	bidirectional 8-bit bus Port 1
DB2	9	I/O	bidirectional 8-bit bus Port 2
V _{SS}	10	-	ground
DB3	11	I/O	bidirectional 8-bit bus Port 3
DB4	12	I/O	bidirectional 8-bit bus Port 4
DB5	13	I/O	bidirectional 8-bit bus Port 5
DB6	14	I/O	bidirectional 8-bit bus Port 6
DB7	15	I/O	bidirectional 8-bit bus Port 7
RD (DTACK)	16	I(O)	RD is the read control input for MAB8049, MAB8051 or Z80-types. DTACK is the data transfer control output for 68000-types (open-drain).
CS	17	I	chip select input (internal pull-up)
WR (R/W)	18	I	WR is the write control input for MAB8048, MAB8051, or Z80-types (internal pull-up). R/W control input for 68000-types.
RESET/ STROBE	19	I/O	Reset input (open-drain); this input forces the I ² C-bus controller into a predefined state; all flags are reset, except PIN, which is set. Also functions as strobe output.
V _{DD}	20	-	supply voltage

I²C-bus controller

PCF8584

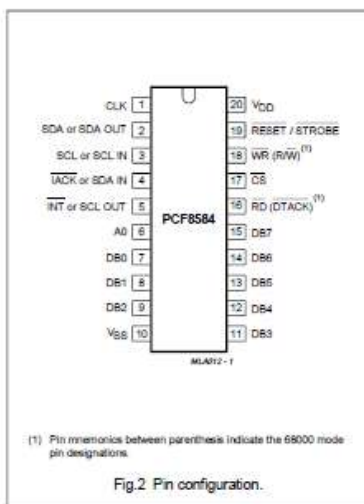


Table 1 Control signals utilized by the PCF8584 for microcontroller/microprocessor interfacing

TYPE	R/W	WR	R	DTACK	IACK
8048/8051	no	yes	yes	no	no
68000	yes	no	no	yes	yes
Z80	no	yes	yes	no	yes

The structure of the PCF8584 is similar to that of the I²C-bus interface section of the Philips' MABXXXX/PCF84(C)XX-series of microcontrollers, but with a modified control structure. The PCF8584 has five internal register locations. Three of these (own address register S0, clock register S2 and interrupt vector S3) are used for initialization of the PCF8584. Normally they are only written once directly after resetting of the PCF8584.

The remaining two registers function as double registers (data buffer/shift register S0, and control/status register S1) which are used during actual data transmission/reception. By using these double registers, which are separately write and read accessible, overhead for register access is reduced. Register S0 is a combination of a shift register and data buffer.

Register S0 performs all serial-to-parallel interfacing with the I²C-bus.

Register S1 contains I²C-bus status information required for bus access and/or monitoring.

6 FUNCTIONAL DESCRIPTION

6.1 General

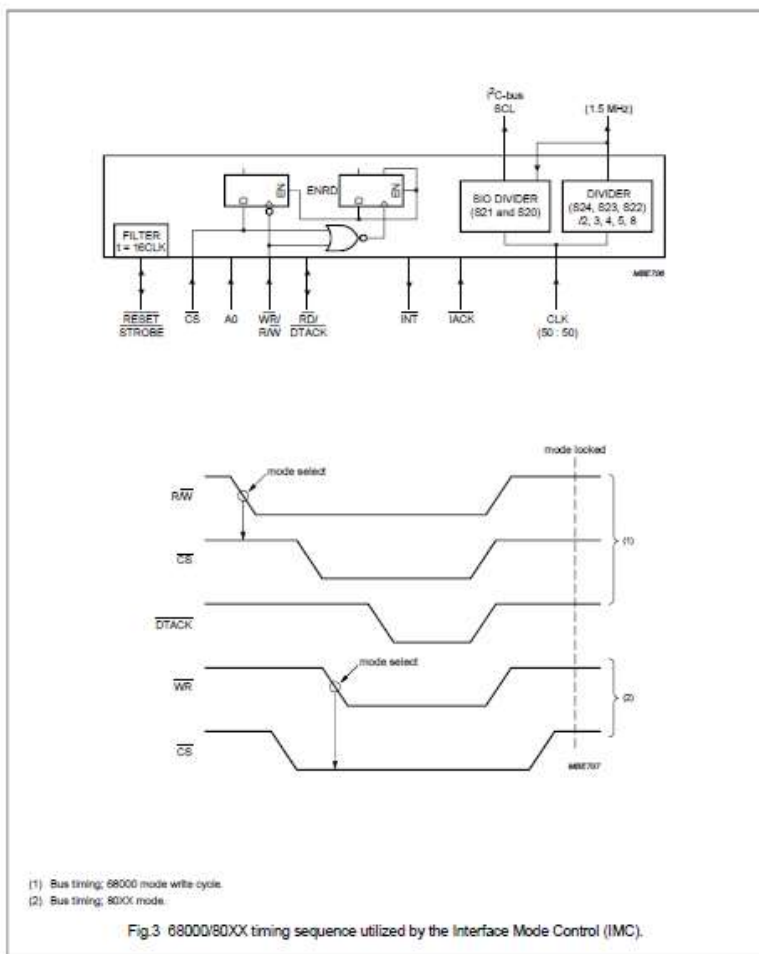
The PCF8584 acts as an interface device between standard high-speed parallel buses and the serial I²C-bus. On the I²C-bus, it can act either as master or slave. Bidirectional data transfer between the I²C-bus and the parallel-bus microcontroller is carried out on a byte-wise basis, using either an interrupt or polled handshake. Interface to either 80XX-type (e.g. 8048, 8051, Z80) or 68000-type buses is possible. Selection of bus type is automatically performed (see Section 6.2).

6.2 Interface Mode Control (IMC)

Selection of either an 80XX mode or 68000 mode interface is achieved by detection of the first WR-CS signal sequence. The concept takes advantage of the fact that the write control input is common for both types of interfaces. An 80XX-type interface is default. If a HIGH-to-LOW transition of WR (R/W) is detected while CS is HIGH, the 68000-type interface mode is selected and the DTACK output is enabled. Care must be taken that WR and CS are stable after reset.

I²C-bus controller

PCF8584

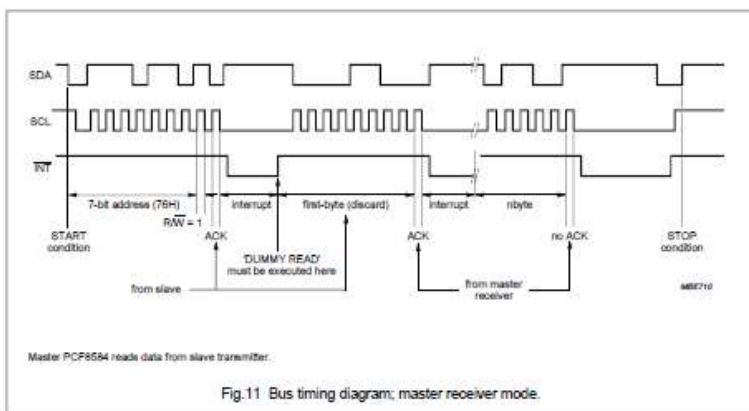
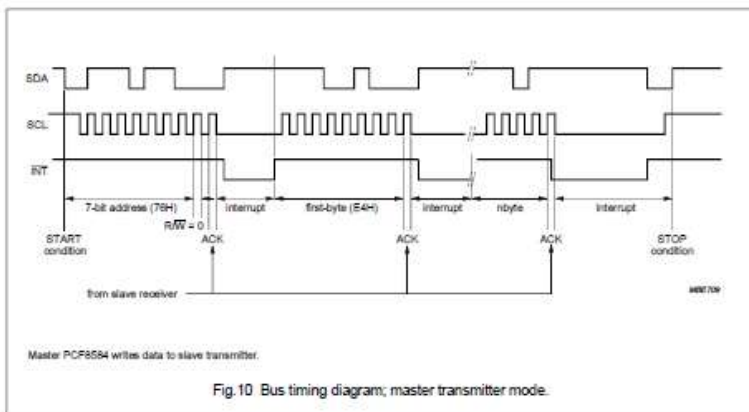


I²C-bus controller

PCF8584

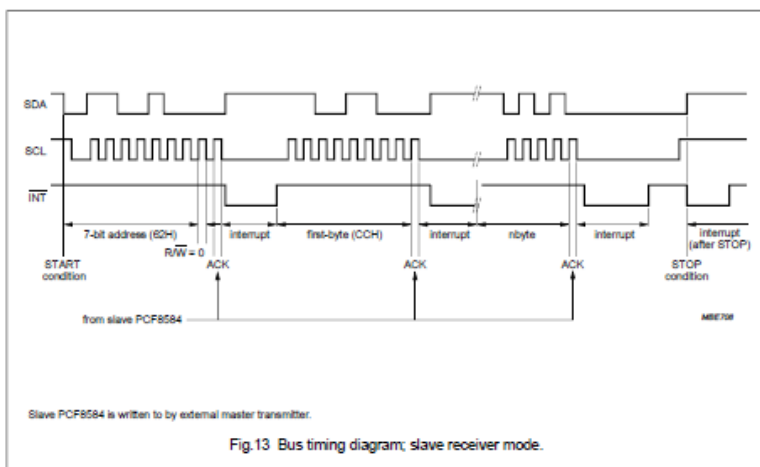
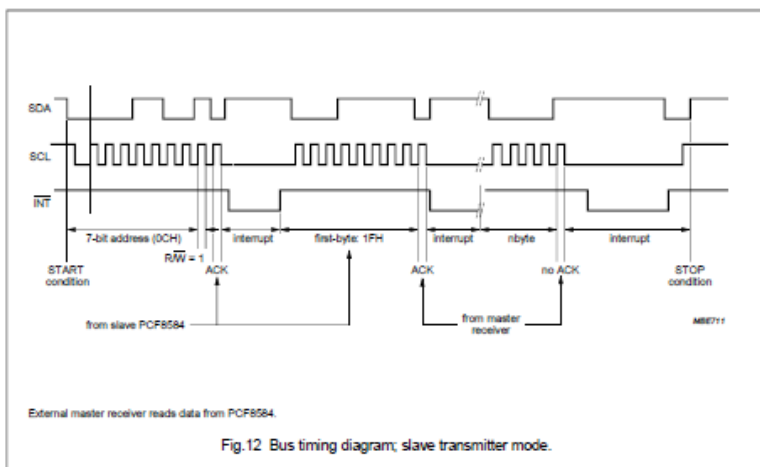
8 I²C-BUS TIMING DIAGRAMS

The diagrams (Figs 10 to 13) illustrate typical timing diagrams for the PCF8584 in master/slave functions. For detailed description of the I²C-bus protocol, please refer to *The I²C-bus and how to use it*; Philips document ordering number 9398 393 40011.



I²C-bus controller

PCF8584



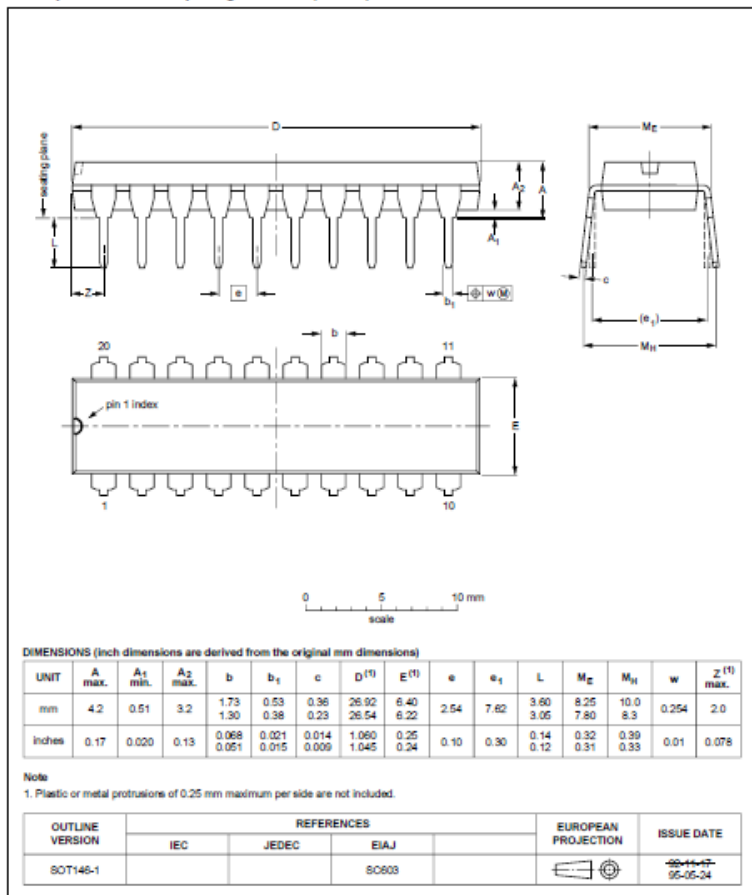
I²C-bus controller

PCF8584

15 PACKAGE OUTLINES

DIP20: plastic dual in-line package; 20 leads (300 mil)

SOT146-1



Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel: +61 2 9805 4455, Fax: +61 2 9805 4496

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213, Tel: +43 160 1010,
Fax: +43 160 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, 1 1211, Volodarski Str. 6,
220050 MINSK, Tel: +375 172 200 733, Fax: +375 172 200 773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoimport, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel: +359 2 689 211, Fax: +359 2 689 102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel: +1 800 234 7381

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel: +852 2319 7885, Fax: +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN 8,
Tel: +45 30 86 2636, Fax: +45 31 57 0044

Finland: Sinkkilaantie 3, FIN-02630 ESPOO,
Tel: +358 9 615600, Fax: +358 9 6158020

France: 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,
Tel: +33 1 40 99 6161, Fax: +33 1 40 99 6427

Germany: Hammettbrockstraße 60, D-20097 HAMBURG,
Tel: +49 40 23 63 60, Fax: +49 40 23 535 300

Greece: No. 15, 25th March Street, GR 17776 TAVROSATHENS,
Tel: +30 1 4804 330239, Fax: +30 1 4814 340

Hungary: see Austria

India: Philips INDIA Ltd. Band Box Building, 2nd floor,
254-D, Dr. Amree Besant Road, Worli, MUMBAI 400 025,
Tel: +91 22 463 8541, Fax: +91 22 463 0666

Indonesia: see Singapore

Ireland: Newstead, Clonsilla, DUBLIN 14,
Tel: +353 1 7640 000, Fax: +353 1 7640 200

Israel: RAPA Electronic, 7 Kfarat Bialerim St. PO Box 16053,
TEL AVIV 61180, Tel: +972 3 845 0444, Fax: +972 3 849 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel: +39 2 6762 2551, Fax: +39 2 6762 2557

Japan: Philips Bldg 13-37, Kojimae 2-chome, Minato-ku, TOKYO 108,
Tel: +81 3 3740 5130, Fax: +81 3 3740 5077

Korea: Philips House, 260-199 Saewoon-dong, Yongseon-ku, SEOUL,
Tel: +82 2 700 1410, Fax: +82 2 709 1415

Malaysia: No. 75 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel: +60 3 750 5214, Fax: +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel: +9-5 800 234 7381

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel: +31 40 27 82755, Fax: +31 40 27 88969

New Zealand: 2 Wagner Place, C.P.O. Box 1041, AUCKLAND,
Tel: +64 9 840 4160, Fax: +64 9 840 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel: +47 22 74 8000, Fax: +47 22 74 8341

Philippines: Philips Semiconductors Philippines Inc.,
105 Valero St. Salcedo Village, P.O. Box 2108 MCC, MAKATI,
Metro MANILA, Tel: +63 2 816 6360, Fax: +63 2 817 3474

Poland: Ul. Lukieka 10, PL 04-123 WARSZAWA,
Tel: +48 22 612 2831, Fax: +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 110048 MOSCOW,
Tel: +7 095 755 6918, Fax: +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1291,
Tel: +65 350 2539, Fax: +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S. A. PHILIPS Pty Ltd., 195-215 Main Road Matieland,
2092 JOHANNESBURG, P.O. Box 7430, Johannesburg 2000,
Tel: +27 11 470 5911, Fax: +27 11 470 5494

South America: Rua do Rio do 220, 5th floor, Suite 51,
04753-903 São Paulo, SÃO PAULO - SP, Brazil,
Tel: +55 11 821 2333, Fax: +55 11 829 1849

Spain: Balnear 22, 08007 BARCELONA,
Tel: +34 9 301 6312, Fax: +34 9 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel: +46 8 632 2000, Fax: +46 8 632 2745

Switzerland: Almdenstrasse 140, CH-8027 ZÜRICH,
Tel: +41 1 488 2686, Fax: +41 1 481 7730

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,
TAIPEI, Taiwan Tel: +886 2 2134 2665, Fax: +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
2092 Saranpuath-Bangna Road Fraikamong, BANGKOK 10260,
Tel: +66 2 745 4000, Fax: +66 2 398 0703

Turkey: Talatpaşa Cad. No. 5, 80640 GÖLTERPE/İSTANBUL,
Tel: +90 212 279 2770, Fax: +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Paltov Lumbra str., Building B, Floor 7,
252042 KIEV, Tel: +380 44 264 2776, Fax: +380 44 265 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX, ULS 9BX, Tel: +44 181 700 5000, Fax: +44 181 704 9421

United States: 611 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel: +1 800 234 7381

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trig N. Pasko 5V, 11000 BEOGRAD,
Tel: +381 11 625 344, Fax: +381 11 635 777

For all other countries apply to: Philips Semiconductors, Marketing & Sales Communications,
Building 9E-p, P.O. Box 219, 5600 MD EINDHOVEN, The Netherlands, Fax: +31 40 27 34825

Internet: <http://www.semiconductors.philips.com>

© Philips Electronics N.V. 1997

8CA55

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands

410707/0059/94e

Date of release: 1997 Oct 21

Document order number: 2397 790 02902

Let's make things better.

Philips
Semiconductors



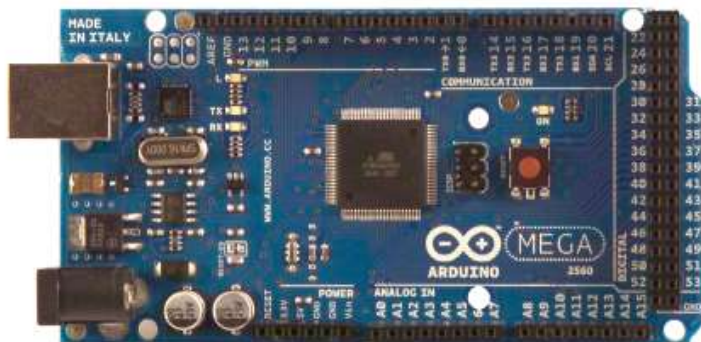
PHILIPS

-Halaman Ini Sengaja Dikosongkan-

LAMPIRAN F

- Data Sheet Arduino Mega 2560

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP 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 Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

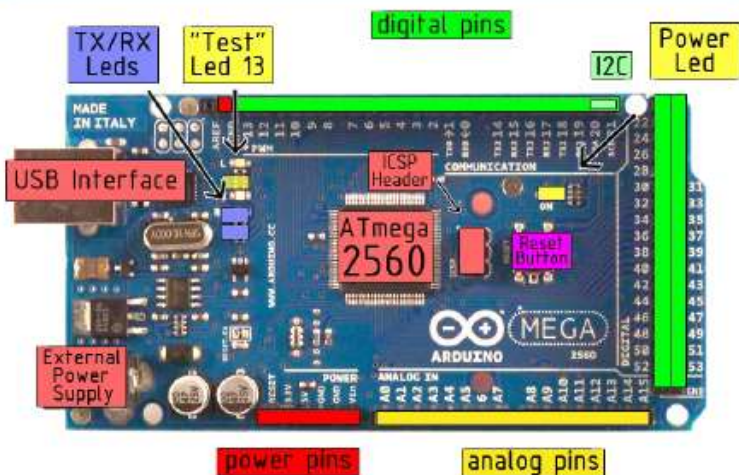


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 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 Mega2560 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.

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 ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega 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); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 5), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2). 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:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **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.
- **I²C:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 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 [analogReference\(\)](#) function.

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.



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// The loop() method runs over and over again,
// as long as the Arduino has power.

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000);               // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000);               // wait for a second
}
```

Done compiling
Press Compile button (to check for errors)

Upload

TX RX Flashing

Blinking Led!



radiospares

RADIONICS



BIODATA PENULIS



Penulis yang bernama lengkap Yusri Saputra atau biasa dipanggil Yusri lahir di Desa Barangmamase Kecamatan Sajoanging Kab. Wajo Provinsi Sulawesi Selatan. Sejak SMP sampai Perguruan Tinggi penulis menempuh pendidikan dengan merantau ke kota orang. Lahir dari pasangan yang sangat berbahagia dengan menanti anak laki-laki satu-satunya dari 3 bersaudara dalam keluarga dari bapak yang bernama H. Safri dan Ibu Hj. Nurbaya. Sekarang penulis menempuh pendidikan di Institut Teknologi Sepuluh Nopember Surabaya dengan Jurusan Teknik Elektro angkatan 2015. Selama menjalani perkuliahan, penulis aktif dalam berbagai organisasi dan kepanitaan. Penulis juga menjabat sebagai Koordinator Asisten pada Lab. Elektronika Mikro dan Sistem Tertanam B202 periode 2018-2019 dan menjadi salah satu peneliti di Lab. Elektronika Industri B402. Judul tugas akhir penulis adalah “Uji Kualitas Minuman Menggunakan Sensor Potensiometrik, Konduktivitas Listrik, Optik dan Metode Jaringan Syaraf Tiruan”. Penulis berharap kedepannya Tugas Akhir ini dapat dikembangkan dan berguna bagi banyak orang. Penulis dapat dihubungi melalui *e-mail*: yusrisaputra66@gmail.com.

-Halaman Ini Sengaja Dikosongkan-